Intel® Trust Domain Extensions (Intel® TDX) TD Migration Architecture Application Binary Interface (ABI) Reference Specification

369010-001US

June 2026

.

Notices and Disclaimers

Intel Corporation (“Intel”) provides these materials as-is, with no express or implied warranties.

All products, dates, and figures specified are preliminary, based on current expectations, and are subject to change without notice. Intel does not guarantee the availability of these interfaces in any future product. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

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

Intel technologies might require enabled hardware, software, or service activation. Some results have been estimated or simulated. Your costs and results might vary.

No product or component can be absolutely secure.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted that includes the subject matter disclosed herein.

No license (express, implied, by estoppel, or otherwise) to any intellectual-property rights is granted by this document.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice.

Copies of documents that have an order number and are referenced in this document or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting http://www.intel.com/design/literature.htm.

TD Migration Application Binary Interface (ABI) Reference

Table of Contents

TD Migration Application Binary Interface (ABI) Reference

1. About this Document

1.1. Scope of this Document

This document describes the Application Binary Interface (ABI) of the Intel® Trust Domain Extensions (Intel® TDX) module's TD Migration architecture.

This document is part of the **TDX Module Architecture Specification Set**, which includes the following documents:

Table 1.1: TDX Module Architecture Specification Set

Document Name	Reference	Description
TDX Module **Base Architecture Specification**	[TDX Module Base Spec]	Base TDX Module architecture overview and specification, covering key management, TD lifecycle management, memory management, virtualization, measurement and attestation, service TDs, debug aspects etc.
TDX Module **TD Migration Architecture Specification**	[TD Migration Spec]	Architecture overview and specification for TD migration
TDX Module **TD Partitioning Architecture Specification**	[TD Partitioning Spec]	Architecture overview and specification for TD Partitioning
TDX Module Interrupt Virtualization Architecture Specification	[Interrupt Virtualization Spec]	Architecture overview and specification for interrupt virtualization
TDX Module **TDX Connect Specification**	[TDX Connect Spec]	Architecture overview and specification for TDX Connect
TDX Module **ABI Reference Specification**	[TDX Module ABI Spec]	Detailed TDX Module Application Binary Interface (ABI) reference specification, covering the TDX Module architecture (except TD Migration and TDX Connect)
→ **TDX Module** **TD Migration ABI Reference Specification**	[TD Migration ABI Spec]	Detailed TDX Module Application Binary Interface (ABI) reference specification, covering the TD Migration architecture
TDX Module **TDX Connect ABI Reference Specification**	[TDX Connect ABI Spec]	Detailed TDX Module Application Binary Interface (ABI) reference specification, covering the TDX connect architecture
TDX Module ABI Reference Tables	[TDX Module ABI Tables]	A set of files detailing TDX Module Application Binary Interface (ABI)
TDX Module ABI Incompatibilities	[TDX Module ABI Incompatibilities]	Description of the incompatibilities between TDX 1.0 and TDX 1.4/1.5 that may impact the host VMM and/or guest TDs

This document is a work in progress and is subject to change based on customer feedback and internal analysis. This document does not imply any product commitment from Intel to anything in terms of features and/or behaviors.

Note: The contents of this document are accurate to the best of Intel's knowledge as of the date of publication, though Intel does not represent that such information will remain as described indefinitely in light of future research and design implementations. Intel does not commit to updating this document in real time when such changes occur.

1.2. Glossary

See the [TDX Module Base Spec].

1.3. Notation

See the [TDX Module Base Spec].

1.4. References

See the [TDX Module Base Spec].

TD Migration Application Binary Interface (ABI) Reference

2. Data Types

This section describes data types that are designed to be used by the Intel TDX Module.

2.1. Service TD Types

2.1.1. SERVTD_BINDING_TABLE: Service TD Binding Table

SERVTD_BINDING_TABLE is a table of service TD binding information, held in the TDCS. For details, see the [TDX Module Base Spec].

Table 2.1: Service TD Binding Entry Definition

Field	Type	Offset (Bytes)	Size (Bytes)	Description
STATE	SERVTD_BINDING_STATE	0	1	See below and [TDX Module Base Spec]
Reserved		1	1	Must be 0
TYPE	SERVTD_TYPE	2	2	See below and [TDX Module Base Spec]
Reserved		4	4	Must be 0
ATTR	SERVTD_ATTR	8	8	See below and [TDX Module Base Spec]
UUID	256-bit blob	16	32	See [TDX Module Base Spec]
INFO_HASH	SHA384_HASH	48	48	See [TDX Module Base Spec]
Reserved		96	32	Must be 0

TD-Preserving Update TDX Module Handoff Compatibility

SERVTD_BINDING_TABLE is preserved in memory across TD-preserving updates. The table below specifies the MODULE_HV versions for which the above MIGSC definition is appliable.

Table 2.2: SERVTD_BINDING_TABLE Compatibility with TD Preserving Updates

Module Handoff Version	Value
Minimum MODULE_HV	0
Maximum MODULE_HV	0

2.1.2. SERVTD_BINDING_STATE: Service TD Binding State

SERVTD_BINDING_STATE indicates the state of the service TD binding slot. For details, see the [TDX Module Base Spec].

Table 2.3: SERVTD_BINDING_STATE Values

Value	Name
0	NOT_BOUND
1	PRE_BOUND
2	BOUND
3	REBIND_BOUND

2.1.3. SERVTD_TYPE: Service TD Binding Type

SERVTD_TYPE is a 16-bit field which specifies the binding type of a service TD. For details, see the [TDX Module Base Spec].

Table 2.4: SERVTD_TYPE Definition

Value	Meaning	Multiple Bindings	Metadata Access
0	Migration TD	No	Migration session key
Other	Reserved	N/A	N/A

2.1.4. SERVTD_ATTR: Service TD Binging Attributes

SERVTD_ATTR is a 64-bit field which specifies the binding attributes of a service TD. For details, see the [TDX Module Base Spec].

Table 2.5: SERVTD_ATTR Definition

Bit(s)	Name	Description
31:0	RESERVED	Must be 0
32	IGNORE_ATTRIBUTES	If set to 1, a value of 0 is used instead of the service TD's ATTRIBUTES field when calculating SERVTD_INFO_HASH
33	IGNORE_XFAM	If set to 1, a value of 0 is used instead of the service TD's XFAM field when calculating SERVTD_INFO_HASH
34	IGNORE_MRTD	If set to 1, a value of 0 is used instead of the service TD's MRTD field when calculating SERVTD_INFO_HASH
35	IGNORE_MRCONFIGID	If set to 1, a value of 0 is used instead of the service TD's MRCONFIGID field when calculating SERVTD_INFO_HASH
36	IGNORE_MROWNER	If set to 1, a value of 0 is used instead of the service TD's MROWNER field when calculating SERVTD_INFO_HASH
37	IGNORE_MROWNERCONFIG	If set to 1, a value of 0 is used instead of the service TD's MROWNERCONFIG field when calculating SERVTD_INFO_HASH
38	IGNORE_RTMR0	If set to 1, a value of 0 is used instead of the service TD's RTMR0 field when calculating SERVTD_INFO_HASH
39	IGNORE_RTMR1	If set to 1, a value of 0 is used instead of the service TD's RTMR1 field when calculating SERVTD_INFO_HASH
40	IGNORE_RTMR2	If set to 1, a value of 0 is used instead of the service TD's RTMR2 field when calculating SERVTD_INFO_HASH
41	IGNORE_RTMR3	If set to 1, a value of 0 is used instead of the service TD's RTMR3 field when calculating SERVTD_INFO_HASH
42	IGNORE_SERVTD_HASH	If set to 1, a value of 0 is used instead of the service TD's SERVTD_HASH field when calculating SERVTD_INFO_HASH
43	IGNORE_MRSIGROOT	If set to 1, a value of 0 is used instead of the service TD's MRSIGROOT field when calculating SERVTD_INFO_HASH **Enumeration:** Support of this bit is enumerated by TDX_FEATURES0.TD_SIGNING_AND_SVN (bit 22). If not supported, it must be 0.
44	IGNORE_MRSIGNER	If set to 1, a value of 0 is used instead of the service TD's MRSIGNER field when calculating SERVTD_INFO_HASH **Enumeration:** Support of this bit is enumerated by TDX_FEATURES0.TD_SIGNING_AND_SVN (bit 22). If not supported, it must be 0.

Bit(s)	Name	Description
45	IGNORE_PRODID	If set to 1, a value of 0 is used instead of the service TD's PRODID field when calculating SERVTD_INFO_HASH **Enumeration:** Support of this bit is enumerated by TDX_FEATURES0.TD_SIGNING_AND_SVN (bit 22). If not supported, it must be 0.
46	IGNORE_ISVSVN	If set to 1, a value of 0 is used instead of the service TD's ISVSVN field when calculating SERVTD_INFO_HASH **Enumeration:** Support of this bit is enumerated by TDX_FEATURES0.TD_SIGNING_AND_SVN (bit 22). If not supported, it must be 0.
47	IGNORE_MRCONFIGSVN	If set to 1, a value of 0 is used instead of the service TD's MRCONFIGSVN field when calculating SERVTD_INFO_HASH **Enumeration:** Support of this bit is enumerated by TDX_FEATURES0.TD_SIGNING_AND_SVN (bit 22).
48	IGNORE_MROWNERCONFIGSVN	If set to 1, a value of 0 is used instead of the service TD's MROWNERCONFIGSVN field when calculating SERVTD_INFO_HASH **Enumeration:** Support of this bit is enumerated by TDX_FEATURES0.TD_SIGNING_AND_SVN (bit 22).
Other	RESERVED	Must be 0

2.1.5. SERVTD_EXT_STRUCT: Extended Service TD Info

SERVTD_EXT_STRUCT provides additional attestation context about the TD's TCB.

Table 2.6: SERVTD_EXT_STRUCT Definition

Name	Offset	Size	Type	Description
INIT_INFO_HASH	0	48	SHA384	For Service TD 0 (Migration TD), if MIG_SETUP_TD_POLICY_HASH has been configured[1], INIT_INFO_HASH contains the value of MIG_SETUP_TD_POLICY_HASH at TDH.MR.FINALIZE time. Else, INIT_INFO_HASH contains the value of the Service TD binding entry's INFO_HASH at TDH.MR.FINALIZE time.
INIT_ATTR	48	8	MASK	For Service TD 0 (Migration TD), if MIG_SETUP_TD_POLICY_HASH has been configured[1], INIT_ATTR's value is 0. Else, INIT_ATTR contains the value of the Service TD binding entry's ATTR at TDH.MR.FINALIZE time
RESERVED	56	8		Set to 0
INIT_CPUSVN	64	16		CPUSVN at TDH.MNG.INIT time
INIT_TEE_TCB_SVN	80	16		TEE_TCB_SVN at TDH.MNG.INIT time
INIT_TEE_MODEL	96	12	TEE_MODEL_STRUCT	CPU and Platform Model information, as captured at TDH.SYS.INIT time. See 2.1.6 below.
RESERVED	108	4		Set to 0

[1] MIG_SETUP_TD_POLICY_HASH can only be written if the TDX Module supports the Migration Setup Service, as enumerated by TDX_FEATURES0.MIG_SETUP (bit 55), and it was enabled by TDH.SYS.CONFIG or TDH.SYS.UPDATE.

Name	Offset	Size	Type	Description
CUR_INFO_HASH	112	48	SHA384	For Service TD 0 (Migration TD), if MIG_SETUP_TD_POLICY_HASH has been configured[1], INIT_INFO_HASH contains the value of MIG_SETUP_TD_POLICY_HASH at TDH.MR.FINALIZE time. Else, CUR_INFO_HASH contains the value or the current Service TD binding entry's INFO_HASH
CUR_ATTR	160	8	MASK	For Service TD 0 (Migration TD), if MIG_SETUP_TD_POLICY_HASH has been configured[1], CUR_ATTR's value is 0. Else, CUR_ATTR contains the value of the current Service TD binding entry's ATTR
RESERVED	168	8		Set to 0
RESERVED	176	48	SHA384	Reserved for Cascading hash of Audit Log. Set to 0
RESERVED	224	48	SHA384	Set to 0

2.1.6. TEE_MODEL_STRUCT

TEE_MODEL_STRUCT encodes platform-specific identifiers, including the Family/Model (FM) and Platform ID, into a single value. This value is captured during TDH.SYS.INIT and is used to:

- Bind the TD's initial state to a specific platform configuration.
- Differentiate between SVNs of different platforms across operations such as migration, rebind, or attestation.

Table 2.7: TEE_MODEL_STRUCT Definition

Name	Offset (Bytes)	Size (Bytes)	Type	Description
CUSTOM	0	2	Unsigned Integer	Custom data, set to 0.
PLATFORM_ID	2	2	Unsigned Integer	Platform ID as defined in the IA32_PLATFORM_ID MSR bits 52:50, other bits are 0.
FM	4	4	Unsigned Integer	Family and Model, formatted as in CPUID(1).EAX, with the Stepping value set to 0.
RESERVED	8	4	N/A	Reserved, set to 0x80000000.

2.1.7. SERVTD_EXT_HASH

The SERVTD_EXT_HASH is the SHA384 hash of SERVTD_EXT_STRUCT.

When computing the hash for purpose of TDCS.SERVTD_ACCEPT_SERVTD_EXT_HASH, the CUR_INFO_HASH and CUR_ATTR fields are zeroed before computing the hash.

2.2. Migration Types

Enumeration: The following definitions are applicable for TDX Module which support TD migration, as enumerated by TDX_FEATURES0.TD_MIGRATION (bit 0) or S4, as enumerated by TDX_FEATURES0.S4 (bit 13).

2.2.1. MBMD: Migration Bundle Metadata

MBMD is composed of a common header and variable type-specific information.

2.2.1.1. Generic MBMD Structure

The maximum overall size of MBMD is 128 bytes.

Table 2.8: Generic MBMD Structure Definition

Field	Offset (Bytes)	Size (Bytes)	Description	Included In MAC	Included in IV
SIZE	0	2	Overall size of the MBMD structure, in bytes	Yes	No
MIG_VERSION	2	2	Migration protocol version Changes in MBMB format, other migration bundle components format or migration protocol sequence require updating the protocol version. The migration protocol version is set by the MigTD before migration session starts.	Yes	No
MIGS_INDEX	4	2	Index of the migration stream used for migrating this migration bundle	As 0	Yes
MB_TYPE	6	1	The type of information being migrated: 0: TD-scope immutable non-memory state 1: TD-scope mutable non-memory state 2: VCPU-scope mutable non-memory state 3–15: Reserved 16: TD private memory 17–31: Reserved 32: Epoch token 33: Abort token Other: Reserved	Yes	No
RESERVED	7	1	Reserved, must be 0	Yes	No
MB_COUNTER	8	4	Per-stream migration bundle counter Starts from 0 on each migration epoch start, incremented by 1 on each MBMD export to the associated stream.	Yes	No
MIG_EPOCH	12	4	Migration epoch Starts from 0 on migration session start, incremented by 1 on each epoch token. A value of 0xFFFFFFFF indicates out-of-order phase.	Yes	No
IV_COUNTER	16	8	Monotonously incrementing counter, used as a component in the AES-GCM IV	As 0	Yes
Type-Specific Information	24	Variable	Variable-sized additional information for each specific type of MBMD	Yes	No
MAC	24+V	16	AES-256-GCM MAC over other MBMD fields and any associated migration data (all the migration pages)	No	No

2.2.1.2. TD-Scope Immutable Non-Memory State MBMD Fields

Table 2.9: TD-Scope Immutable Non-Memory State MBMD Fields Definition

Field	Offset (Bytes)	Size (Bytes)	Description	Included In MAC	Included in IV
NUM_F_MIGS	24	2	Maximum number of forward migration streams that will be used	Yes	No
RESERVED	26	2	Reserved, must be 0	Yes	No
NUM_SYS_MD_PAGES	28	1	Number of pages in the page list used for migrating TDX Module metadata	Yes	No
RESERVED	29	3	Reserved, must be 0	Yes	No

2.2.1.3. TD-Scope Mutable Non-Memory State MBMD Fields

Table 2.10: TD-Scope Mutable Non-Memory State MBMD Fields Definition

Field	Offset (Bytes)	Size (Bytes)	Description	Included In MAC	Included in IV
RESERVED	24	8	Reserved, must be 0	Yes	No

2.2.1.4. VCPU-Scope Mutable Non-Memory State MBMD Fields

Table 2.11: VCPU-Scope Mutable Non-Memory State MBMD Fields Definition

Field	Offset (Bytes)	Size (Bytes)	Description	Included In MAC	Included in IV
VP_INDEX	24	2	Virtual CPU index	Yes	No
RESERVED	26	6	Reserved, must be 0	Yes	No

2.2.1.5. TD Private Memory MBMD Fields

Table 2.12: TD Private Memory MBMD Type-Specific Fields Definition

Field	Offset (Bytes)	Size (Bytes)	Description	Included In MAC	Included in IV
NUM_GPAS	24	2	Number of entries in the GPA list	Yes	No
GPA_LIST_ATTRIBUTES	26	1	Attributes of the GPA list, see Table 2.13 below	Yes	No
RESERVED	27	5	Reserved, must be 0	Yes	No

Table 2.13: GPA_LIST_ATTRIBUTES

Bits	Name	Description		
2:0	**FORMAT**	GPA list format		
		Value	**Name**	**Description**
		0	GPA_ONLY	A GPA list page is provided
		1	GPA_AND_ATTR	A GPA list page and a page attributes list page are provided

Bits	Name	Description		
		Other	RESERVED	Reserved
7:3	**RESERVED**	Reserved: must be 0		

2.2.1.6. Epoch Token MBMD Fields

Table 2.14: Epoch Token MBMD Fields Definition

Field	Offset (Bytes)	Size (Bytes)	Description	Included In MAC	Included in IV
TOTAL_MB	24	8	The total number of migration bundles, including the current one, which have been exported since the beginning of the migration session	Yes	No

2.2.1.7. Abort Token MBMD Fields

Table 2.15: Abort Token MBMD Fields Definition

Field	Offset (Bytes)	Size (Bytes)	Description	Included In MAC	Included in IV
RESERVED	24	8	Reserved, must be 0	Yes	No

2.2.1.8. TD Migration Protocol Version Compatibility

The table below specifies the TD migration protocol versions for which the above MBMD definition is appliable.

Table 2.16: MBMD Compatibility with TD Migration Versions

TD Migration Version	Minimum	Maximum
Export version	0	0
Import version	0	0

2.2.2. GPA List

2.2.2.1. Overview

A GPA list specifies a list of 4KB-aligned GPAs, each with associated attributes, required operation and status. A single GPA list may have up to 512 entries, is contained in a single 4KB page and must be aligned on 4KB. The GPA list may contain null entries, as indicated by the OPERATION field's value set to 0 (NOP).

Interface functions that process up to 512 GPAs, such as TDH.EXPORT.MEM, use a single GPA list page. The GPA list is specified by GPA_LIST_INFO, which contains the HPA of the GPA list page and the index of the first entry and last entry to be processed.

Interface functions that process up to 512^2 GPAs, such as TDH.MEM.SCAN, use multiple GPA list pages. The list of pages is specified by a page of GPA_LIST_INFO entries. The GPA list info page is specified by LIST_OF_LISTS_INFO, which contains the HPA of the GPA list info page and the index of the first entry and last entry to be processed.

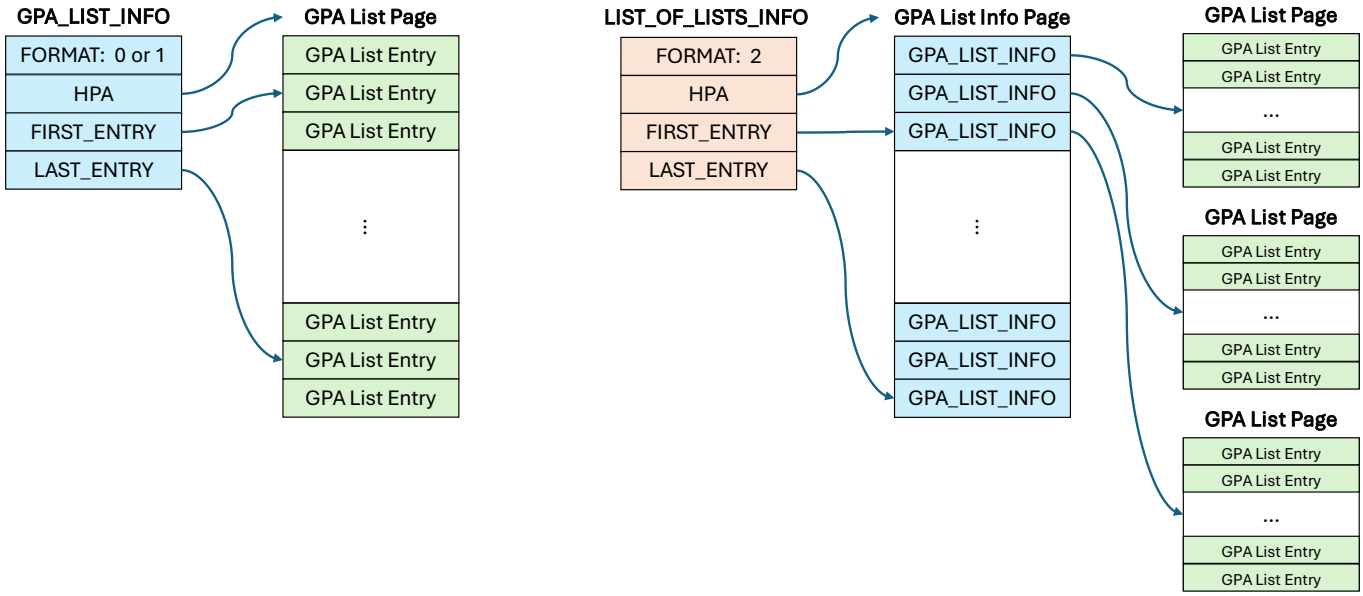

Figure 2.1: GPA List and List of Lists

2.2.2.2. GPA_LIST_INFO: HPA, First and Last Entries of a GPA List

GPA_LIST_INFO is a 64b structure used as a GPR input and output operand of multiple migration interface functions, e.g., TDH.EXPORT.MEM. It provides the HPA of the GPA list page in shared memory, and the index of the first entry and last entries to be processed. When used with some interface functions, it may provide list-of-lists information.

Table 2.17: GPA_LIST_INFO

Bits	Name	Description		
2:0	**FORMAT**	GPA list format		
		Value	**Name**	**Description**
		0	GPA_ONLY	A GPA list page is provided
		1	GPA_AND_L2_ATTR	GPA list and L2 page attributes list pages are provided. This format is only used by TDH.EXPORT.MEM and TDH.IMPORT.MEM. It is mandatory for migrating partitioned TDs (which contain one or more L2 VMs). TDX Module support of this feature is enumerated by TDX_FEATURES0.PARTITIONED_TD_MIGRATION, readable by TDH.SYS.RD*.
		2	LIST_OF_LISTS	A list-of-lists page is provided
		Other	RESERVED	Reserved
11:3	**FIRST_ENTRY**	Index of the first entry of the list to be processed		
51:12	**HPA**	Bits 51:12 of the host physical address (including HKID) of the GPA list page, which must be a shared HPA		
54:52	**RESERVED**	Reserved: must be 0		
63:55	**LAST_ENTRY**	Index of the last entry in the GPA list **Note:** LAST_ENTRY may be used as the index of the last entry to process in the current invocation. There may be additional entries in the GPA list. See the definition of TDH.IMPORT.MEM for details.		

2.2.2.3. GPA List Entry

Table 2.18 below shows the format of a GPA list entry as used. The GPA list entry format is designed so that the output of TDH.EXPORT.BLOCKW, TDH.MEM.SCAN.COMP and TDH.MEM.SCAN.RANGE can be used directly with TDH.EXPORT.MEM, and the output of TDH.EXPORT.MEM can be used directly with TDH.IMPORT.MEM.

Table 2.18: GPA List Entry Definition

Bit(s)	Size	Name	Description	TDH.MEM.SCAN	TDH.EXPORT.BLOCKW		TDH.EXPORT.MEM		TDH.IMPORT.MEM		TDH.EXPORT.RESTORE	
				Out	In	Out	In	Out	In	Out	In	Out
1:0	2	LEVEL	Mapping level (size)	0 (4KB), 1 (2MB) or 2 (1GB)	Must be 0 (4KB)	Unmod.	Must be 0 (4KB)	Unmod.	Must be 0 (4KB)	Unmod.	Must be 0 (4KB)	Unmod.
2	1	PENDING	See below	Yes	Ignored	Unmod.	Ignored	Yes	Yes	Unmod.	Ignored	Unmod.
4:3	2	STATE	See below	Yes	Must be 0	Unmod.	Ignored	0	Must be 0	Unmod.	Must be 0	Unmod.
6:5	2	RESERVED	Reserved	0	Must be 0	Unmod.	Must be 0	Unmod.	Must be 0	Unmod.	Must be 0	Unmod.
9:7	3	L2_MAP	See below	0	Ignored	Unmod.	Ignored	Yes	Yes	Unmod.	Ignore	Unmod.
11:10	2	MIG_TYPE	See below	0	Yes	Unmod.	Yes	Unmod.	Yes	Unmod.	Yes	Unmod.
51:12	40	GPA	Guest Physical Address bits 51:12	Yes	Yes	Unmod.	Yes	Unmod.	Yes	Unmod.	Yes	Unmod.
53:52	2	OPERATION	See below	Yes[2]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
55:54	2	RESERVED	Reserved	0	Must be 0	Unmod.	Must be 0	Unmod.	Must be 0	Unmod.	Must be 0	Unmod.
60:56	5	STATUS	See below	Yes[3]	Ignored	Yes	Ignored	Yes	Ignored	Yes	Ignored	Yes
63:61	3	RESERVED	Reserved	0	Must be 0	Unmod.	Must be 0	Unmod.	Must be 0	Unmod.	Must be 0	Unmod.

2.2.2.4. GPA List Entry Details

2.2.2.4.1. LEVEL

LEVEL specifies the GPA mapping size for this entry, as shown in the table below.

Table 2.19: LEVEL Definition

Level	GPA Mapping Size
0	4KB
1	2MB
2	1GB
3	Reserved

For TDH.EXPORT.BLOCKW, TDH.EXPORT.MEM, TDH.EXPORT.RESTORE and TDH.IMPORT.MEM only level 0 (4KB) is supported.

[2] OPERATION value for TDH.MEM.SCAN.RANGE(DSCAN) and TDH.MEM.SCAN.COMP(DCHECK) is always MIGRATE.

[3] STATUS value for TDH.MEM.SCAN.RANGE(DSCAN) and TDH.MEM.SCAN.COMP(DCHECK) is always SUCCESS.

2.2.2.4.2. PENDING

Indicates that the page is Pending. Note that the actual SEPT entry state may be one of multiple PENDING* states.

2.2.2.4.3. STATE

Provides a hint about the page state.

Table 2.20: STATE Values Definition for TDH.MEM.SCAN*

Value	Output	
	Name	Description
0	NOT_EXPORTED	Page is a candidate for initial export.
1	EXPORTED_MODIFIED	Page is a candidate for re-export due to updated content or attributes.
2	EXPORTED_BLOCKED	Page is a candidate for re-export due to being blocked. Blocking is an interim state followed by some other memory management operation; the host VMM may decide to defer the re-export.
3	EXPORTED_REMOVED	Page is a candidate for re-export due to being removed.

2.2.2.4.4. L2_MAP

A bitmap which indicates whether the page is mapped in one or more L2 VM. This field is provided as part of the GPA list entry to enable the host VMM to prepare L2 SEPT pages before invoking TDH.IMPORT.MEM.

2.2.2.4.5. OPERATION

The following tables describe the meaning of OPERATION, as used for each applicable interface function. Note that the OPERATION definitions for TDH.EXPORT.BLOCKW, TDH.EXPORT.MEM and TDH.IMPORT.MEM are designed to be compatible, so that the same GPA list can be used for all of them.

2.2.2.4.5.1. OPERATION Values for TDH.EXPORT.BLOCKW

Table 2.21: Input OPERATION Values Definition for TDH.EXPORT.BLOCKW

Input Value	Name	Description
0	NOP	No operation
1	BLOCKW	Block for writing
2	NOP	No operation
3	BLOCKW	Block for writing

Table 2.22: Output OPERATION Values Definition for TDH.EXPORT.BLOCKW

Output Value	Name	Description
0	NOP	Not blocked for writing
1	BLOCKW	Blocked for writing
2	NOP	Not blocked for writing
3	BLOCKW	Blocked for writing

2.2.2.4.5.2. OPERATION Values for TDH.MEM.SCAN*

Table 2.23: Output OPERATION Values Definition for TDH.MEM.SCAN*

Output Value	Name	Description
0	RESERVED	Reserved
1	EXPORT	Export
2	RESERVED	Reserved
3	RESERVED	Reserved

2.2.2.4.5.3. OPERATION Values for TDH.EXPORT.MEM (For Write Blocking Based Export)

Note that on input, the MIGRATE operation is represented by two possible values. This is done for direct compatibility with the TDH.EXPORT.BLOCKW output.

Table 2.24: Input OPERATION Values Definition for TDH.EXPORT.MEM (Write Blocking Export)

Input Value	Name	Description
0	NOP	No operation
1	MIGRATE	Export (may result in MIGRATE, REMIGRATE or NOP, as determined by the TDX Module)
2	CANCEL	Cancel previous export (may result in CANCEL or NOP, as determined by the TDX Module)
3	MIGRATE	Export (may result in MIGRATE, REMIGRATE or NOP, as determined by the TDX Module)

Table 2.25: Output OPERATION Values Definition for TDH.EXPORT.MEM (Write Blocking Export)

Output Value	Name	Description
0	NOP	Not exported
1	MIGRATE	Initial export during this migration session or following a CANCEL
2	CANCEL	Cancellation of a previous export Not applicable for S4 hibernation.
3	REMIGRATE	Re-export of updated content or attributes

2.2.2.4.5.4. OPERATION Values for TDH.EXPORT.MEM (For Non-Blocking Export)

Table 2.26: Input OPERATION Values Definition for TDH.EXPORT.MEM (Non-Blocking Export)

Input Value	Name	Description
0	NOP	No operation
1	EXPORT	Export request (may result in MIGRATE, REMIGRATE, CANCEL or NOP, as determined by the TDX Module)
2	NOP	No operation
3	NOP	No operation

Table 2.27: Output OPERATION Values Definition for TDH.EXPORT.MEM (Non-Blocking Export)

Output Value	Name	Description
0	NOP	Not exported
1	MIGRATE	Initial export during this migration session or following a CANCEL
2	CANCEL	Cancellation of a previous export Not applicable for S4 hibernation.
3	REMIGRATE	Re-export of updated content or attributes

2.2.2.4.5.5. OPERATION Values for TDH.IMPORT.MEM

Table 2.28: Input OPERATION Values Definition for TDH.IMPORT.MEM

Output Value	Name	Description
0	NOP	No operation
1	MIGRATE	Initial import during this migration session or following a CANCEL
2	CANCEL	Cancel previous import
3	REMIGRATE	Re-import of updated page content or attributes

Table 2.29: Output OPERATION Values Definition for TDH.IMPORT.MEM

Input Value	Name	Description
0	NOP	Not imported
1	MIGRATE	Imported
2	CANCEL	Removed previous import Not applicable for S4 resumption.
3	REMIGRATE	Imported Not applicable for S4 resumption.

2.2.2.4.5.6. OPERATION Values for TDH.EXPORT.RESTORE

Table 2.30: Input OPERATION Values Definition for TDH.EXPORT.RESTORE

Input Value	Name	Description
0	NOP	No operation
1	RESTORE	Restore SEPT entry to non-migration state
2	NOP	Reserved
3	RESTORE	Restore SEPT entry to non-migration state

Table 2.31: Output OPERATION Values Definition for TDH.EXPORT.RESTORE

Output Value	Name	Description
0	NOP	Not restored

Output Value	Name	Description
1	RESTORE	Restored
2	NOP	Not restored
3	RESTORE	Restored

2.2.2.4.6. MIG_TYPE

Table 2.32: MIG_TYPE Values Definition

Value	Name	Description
0	PAGE_4K	4KB private memory page
Other	RESERVED	Reserved for future types

2.2.2.4.7. STATUS

Table 2.33: STATUS Values Definition

Value	Name	Description
0	SUCCESS	GPA list entry was processed successfully
1	SKIPPED	The GPA list entry was skipped because NOP was requested; this is not an error
2	SEPT_WALK_FAILED	Secure EPT walk failed for the requested GPA
3	SEPT_ENTRY_BUSY_HOST_PRIORITY	Secure EPT entry was busy. The host VMM should retry the operation until successful.
4	SEPT_ENTRY_STATE_INCORRECT	Secure EPT entry state was incorrect for the requested operation and the TD's OP_STATE
5	TLB_TRACKING_NOT_DONE	TLB tracking was not done for the requested GPA
6	OP_STATE_INCORRECT	The TD's OP_STATE was incorrect for the requested operation and Secure EPT entry state
7	MIGRATED_IN_CURRENT_EPOCH	Requested GPA has already been migrated during the current migration epoch
8	MIG_BUFFER_NOT_AVAILABLE	Required migration buffer was not provided
9	NEW_PAGE_NOT_AVAILABLE	Required new TD page was not provided
10	INVALID_PAGE_MAC	Page MAC was invalid
11	DISALLOWED_IMPORT_OVER_REMOVED	Page import over a removed page is not allowed
12	TD_PAGE_BUSY_HOST_PRIORITY	TD page was busy. The host VMM should retry the operation until successful.
13	L2_SEPT_WALK_FAILED	L2 Secure EPT walk failed for the requested GPA
14	ATTR_LIST_ENTRY_INVALID	The L2 attributes list entry is invalid
15	GPA_LIST_ENTRY_INVALID	The GPA list entry is invalid
16	INVALID_MIGRATION_BUFFER_HPA	The provided migration buffer HPA is not valid
17	PAGE_DIRTY	Page not exported because the SEPT entry's Dirty bit was set
18	REOWN_DISALLOWED	TDH.IMPORT.MEM operation attempted to change physical page ownership between TD and host VMM, but NO_REOWN input flag was set

Value	Name	Description
19	IOTLB_TRACKING_NOT_DONE	IOMMU IOTLB tracking was not done for the requested GPA
20	SCAN_EPOCH_NOT_RECORDED	TD Epoch value at the time of memory scan is not recorded. This may happen either if no memory scan was done for this page, or if some other memory management operation (e.g., block) was done after the page had been scanned.
Other	Reserved	Reserved

2.2.2.5. TD Migration Protocol Version Compatibility

The table below specifies the TD migration protocol versions for which the above GPA List definition is appliable.

Table 2.34: GPA List Compatibility with TD Migration Versions

TD Migration Version	Minimum	Maximum
Export version	0	0
Import version	0	0

2.2.3. Memory Migration Buffers List

A memory migration buffer list specifies a list of HPAs of 4KB pages in shared memory, to be used as output by TDH.EXPORT.MEM and as input by TDH.IMPORT.MEM. The list may have up to 512 64-bit entries, each containing a 4KB-aligned HPA (including HKID bits) of a page in shared memory. The list is contained in a single 4KB page and must be aligned on 4KB. The page list may contain null entries, indicated by the INVALID bit.

2.2.3.1. Migration Buffers List Entry

Table 2.35: Migration Buffers List Entry

Bits	Name	Description
11:0	**RESERVED**	Reserved: must be 0 (unless bit 63 indicates an invalid entry)
51:12	**HPA**	Bits 51:12 of the host physical address (including HKID) of the migration buffer page, which must be a shared HPA
62:52	**RESERVED**	Reserved: must be 0 (unless bit 63 indicates an invalid entry)
63	**INVALID**	A value of 1 indicates that this entry is invalid

2.2.4. Page Attributes List

A page attributes list specifies a list of page aliases, to be used as output by TDH.EXPORT.MEM and as input by TDH.IMPORT.MEM. The list must contain an entry for each respective GPA list entry used with the same interface functions. The list may have up to 512 64-bit entries, in page L2 attributes format as defined in the [ABI Spec]. The list is contained in a single 4KB page and must be aligned on 4KB. A page attributes list is mandatory for migrating partitioned TDs (which contain one or more L2 VMs).

Enumeration: TDX Module support of page attributes list is enumerated by TDX_FEATURES0.PARTITIONED_TD_MIGRATION, readable by TDH.SYS.RD*.

2.2.5. Memory Migration Page MAC List

A page MAC list specifies a list of MACs over 4KB migrated pages, their GPA list entries and (if applicable) page L2 attributes list entries, to be used as output by TDH.EXPORT.MEM and as input by TDH.IMPORT.MEM. The list must contain an entry for each respective GPA list entry used with the same interface functions. The list may have up to 256

128-bit entries, each containing a single AES-GMAC-256 of a migrated page. The list is contained in a single 4KB page and must be aligned on 4KB.

2.2.6. Non-Memory State Migration Buffers List

A non-memory state migration buffer list specifies a list of HPAs of 4KB pages in shared memory, to be used as output by TDH.EXPORT.STATE.* and as input by TDH.IMPORT.STATE.*. The list may have up to 512 64-bit entries, each containing a 4KB-aligned HPA (including HKID bits) of a page in shared memory. The list is contained in a single 4KB page and must be aligned on 4KB.

TD Migration Application Binary Interface (ABI) Reference

3. Interface Functions

3.1. Host-Side (SEAMCALL) Interface Functions

3.1.1. TDH.EXPORT.ABORT Leaf

TDH.EXPORT.ABORT aborts an export session and allows the source TD to resume normal operation, depending on export state and an abort token received from the destination platform.

3.1.1.1. Input Operands

Table 3.1: TDH.EXPORT.ABORT Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 64
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	HPA of the source TD's TDR page (HKID bits must be 0)		
R8	If an abort token is available, R8 provides the HPA and size of memory of an MBMD structure in memory, as described below. Otherwise, R8's value must be 0.		
	Bits	**Name**	**Description**
	51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
	63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.
R10	Migration stream index:		
	Bits	**Name**	**Description**
	15:0	MIGS_INDEX	Migration stream index – must be 0
	63:16	RESERVED	Reserved: must be 0

3.1.1.2. Output Operands

Table 3.2: TDH.EXPORT.ABORT Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
Other	Unmodified

3.1.1.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.1.3.1. Overview

TDH.EXPORT.ABORT aborts an export session. If successful, i.e., the target TD does not run, the source TD becomes runnable. If called during the out-of-order phase, an abort token received from the destination platform is required.

If the TDX module is configured for non-blocking export, TDH.EXPORT.ABORT resets the internal state held by the TDX module for comprehensive memory scans of the specified TD's GPA address space.

3.1.1.3.2. Enumeration

Availability of TDH.EXPORT.ABORT is enumerated by TDX_FEATURES0.TD_MIGRATION (bit 0), readable by TDH.SYS.RD*. If not supported, calling TDH.EXPORT.ABORT returns a TDX_OPERAND_INVALID(RAX) status.

3.1.1.3.3. Preconditions

An export session must be in progress.

3.1.1.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.3: TDH.EXPORT.ABORT Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	R8	HPA	MBMD buffer	MBMD	R	Shared	128B	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	Comprehensive scan state	N/A	RW	Opaque	N/A	Exclusive	N/A	N/A

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	N/A	All comprehensive scan contexts	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A

3.1.1.5. Completion Status Codes

Table 3.4: TDH.EXPORT.ABORT Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_INCORRECT_MBMD_MAC	
TDX_INVALID_MBMD	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SUCCESS	Operation is successful
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

5

3.1.2. TDH.EXPORT.BLOCKW Leaf

Block a list of TD private 4KB pages for writing and for attributes modification.

3.1.2.1. Input Operands

Table 3.5: TDH.EXPORT.BLOCKW Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 65
		23:16	Version Number	Selects the SEAMCALL interface function version Version may be 0 or 1. See enumeration details below.
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	GPA_LIST_INFO	GPA_LIST_INFO: HPA of a GPA list page in shared memory, and first and last entries to process, as defined in 2.2.2 FORMAT must be GPA_ONLY.		
RDX	TDR	HPA of the source TD's TDR page (HKID bits must be 0)		

3.1.2.2. Output Operands

Table 3.6: TDH.EXPORT.BLOCKW Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RCX	GPA_LIST_INFO	Same as the input value, except that FIRST_ENTRY is updated to the index of the next entry to be processed. If all entries have been processed, FIRST_ENTRY is updated to (LAST_ENTRY + 1) Modulo 512.
R8	LIST_ERR_COUNT	If TDH.EXPORT.BLOCKW was called with version 1 or higher, R8 returns the number of GPA list entries where an error was encountered. Else, R8 is unmodified.
Other		Unmodified

3.1.2.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.2.3.1. Overview

For each 4KB page in the GPA list, if a blocking operation has been requested, TDH.EXPORT.BLOCKW finds the Secure EPT entry for the provided page. If the entry state is correct (MAPPED, PENDING, EXPORTED_DIRTY or PENDING_EXPORTED_DIRTY), TDH.EXPORT.BLOCKW blocks it for writing, by saving and clearing the W bit and setting the Secure EPT entry state (to BLOCKEDW, PENDING_BLOCKEDW, EXPORTED_DIRTY_BLOCKEDW or PENDING_EXPORTED_DIRTY_BLOCKEDW respectively). It records the current TD's TLB epoch in the TD's global BW_EPOCH and marks the GPA list entry as ready for export. If the TD is partitioned, TDH.EXPORT.BLOCKW also blocks any L2 SEPT entries mapping the 4KB page.

3.1.2.3.2. Enumeration

Availability of TDH.EXPORT.BLOCKW is enumerated by TDX_FEATURES0.TD_MIGRATION (bit 0), readable by TDH.SYS.RD*. If not supported, calling TDH.EXPORT.BLOCKW returns a TDX_OPERAND_INVALID(RAX) status.

Support of TDH.EXPORT.BLOCKW version 1 or higher is enumerated by TDX_FEATURES0.LIST_ERROR_COUNT (bit 54).

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.2.3.3. Preconditions

- TDX module is configured for write-blocking based export (see below).
- An export session must be in progress and the TD must not have been paused yet by TDH.EXPORT.PAUSE.

3.1.2.3.4. Export Mode

TDH.EXPORT.BLOCKW is only available if the TDX module is configured for write-blocking based export (the default). If not available, calling TDH.EXPORT.BLOCKW returns a TDX_OPERAND_INVALID(RAX) status.

3.1.2.3.5. GPA List Entry Error Handling

If a page can't be blocked for writing, TDH.EXPORT.BLOCKW marks its GPA list entry as unsuccessful. List processing is not aborted; it continues to the next entry, if applicable.

Note: Skipping a GPA list entry whose OPERATION is NOP is not considered an error. The STATUS field is written as SKIPPED.

If TDH.EXPORT.BLOCKW was called with version 1 or higher, R8 returns the number of GPA list entries where an error was encountered. Else, R8 is unmodified.

The following table shows the possible GPA list entry STATUS values written by TDH.EXPORT.BLOCKW. Refer to 2.2.2.4.7 for STATUS definition.

Table 3.7: GPA List Entry STATUS Values Returned by TDH.EXPORT.BLOCKW

Value	Name	Description
0	SUCCESS	The GPA list entry was processed successfully.
1	SKIPPED	The GPA list entry was skipped because NOP was requested; this is not an error.
2	SEPT_WALK_FAILED	The GPA list entry was skipped due to an SEPT walk failure, e.g., some non-leaf SEPT entry at an upper level was blocked.
3	SEPT_ENTRY_BUSY_HOST_PRIORITY	The GPA list entry was skipped because of a conflict with a concurrent operation.
4	SEPT_ENTRY_STATE_INCORRECT	The GPA list entry was skipped due to an incorrect SEPT entry state. Only entries in the MAPPED, PENDING, EXPORTED_DIRTY and PENDING_EXPORTED_DIRTY states can be blocked for writing.
15	GPA_LIST_ENTRY_INVALID	The GPA list entry is invalid. TDH.EXPORT.BLOCKW terminated.

3.1.2.3.6. Interruptibility

TDH.EXPORT.BLOCKW is interruptible. If a pending interrupt is detected during operation, TDH.EXPORT.BLOCKW returns with a TDX_INTERRUPTED_RESUMABLE status in RAX. RCX is updated with the next list entry index to process, so the host VMM may re-invoke TDH.EXPORT.BLOCKW immediately after handling the interrupt.

3.1.2.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.8: TDH.EXPORT.BLOCKW Memory Operands Information

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	GPA List page	GPA_LIST	RW	Shared	4KB	None	None	None
Explicit	RDX	HPA	TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	N/A	GPA	TD private pages (via GPA list)	Block	None	Private	4KB	None	None	None
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	None	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT tree	N/A	RW	Private	N/A	Shared	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT entry	SEPT Entry	RW	Private	N/A	Exclusive(h)	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT trees	N/A	RW	Private	N/A	Shared(i)	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT entries	SEPT Entry	RW	Private	N/A	Exclusive(i)	N/A	N/A

3.1.2.5. Completion Status Codes

Table 3.9: TDH.EXPORT.BLOCKW Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_EPT_ENTRY_STATE_INCORRECT	
TDX_EPT_WALK_FAILED	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.

Completion Status Code	Description
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SUCCESS	The operation is successful. **Note:** Processing of some GPA list entries may have encountered errors, but this did not cause an abort of the overall operation. The number of such errors is reported in R8.
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.3. TDH.EXPORT.MEM Leaf

TDH.EXPORT.MEM exports a list of TD private pages contents and/or cancellation requests and prepares a migration bundle in shared memory.

3.1.3.1. Input Operands

5 **Table 3.10: TDH.EXPORT.MEM Input Operands Definition**

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 68
		23:16	Version Number	Selects the SEAMCALL interface function version Version may be 0 or 1.
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
		63:24	Reserved	Must be 0
RCX	GPA_LIST_INFO	HPA of a GPA list page in shared memory, and first and last entries to process, as defined in 2.2.2 On a new invocation, FIRST_ENTRY must be 0. On a resumed invocation, FIRST_ENTRY must be the index of the next GPA list entry to export.		
RDX	TDR	HPA of the source TD's TDR page (HKID bits must be 0)		
R8	MBMD	HPA and size of memory of a memory buffer to use for MBMD:		
		Bits	**Name**	**Description**
		51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
		63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.
R9	MIG_BUFF_LIST	HPA (including HKID bits) of a migration buffer list in shared memory, corresponding to the GPA list pointed by RCX – see 2.2.3.		
R10	MIG_STREAM	Migration stream and resume flag:		
		Bits	**Name**	**Description**

Operand	Name	Description		
		15:0	MIGS_INDEX	Migration stream index If N migration streams have been created by TDH.MIG.STREAM.CREATE, then MIGS_INDEX must be lower than N-1.
		62:16	RESERVED	Reserved: must be 0
		63	RESUME	0: This is a new invocation 1: This is resumption of a previously interrupted operation
R11	MAC_LIST_0	HPA (including HKID bits) of a MAC list in shared memory, corresponding to the first 256 entries of the GPA list pointed by RCX – see 2.2.3. If GPA_LIST_INFO.FIRST_ENTRY >= 256, then MAC_LIST_0 is ignored.		
R12	MAC_LIST_1	HPA (including HKID bits) of a MAC list in shared memory, corresponding to the last 256 entries of the GPA list pointed by RCX – see 2.2.3. If GPA_LIST_INFO.LAST_ENTRY < 256, then MAC_LIST_1 is ignored.		
R14	ATTRIB_LIST	If GPA_LIST_INFO.FORMAT is GPA_AND_L2_ATTR, then R14 contains the HPA (including HKID bits) of a page attributes list in shared memory – see 2.2.4. Else, R14 is ignored. An ATTRIB_LIST is mandatory for exporting partitioned TDs (which contain one or more L2 VMs). **Enumeration:** Support of this feature is enumerated by TDX_FEATURES0.PARTITIONED_TD_MIGRATION, readable by TDH.SYS.RD*.		

3.1.3.2. Output Operands

Table 3.11: TDH.EXPORT.MEM Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RCX	GPA_LIST_INFO	Same as the input value, except that FIRST_ENTRY is updated to the index of the next entry to be processed. If all entries have been processed, FIRST_ENTRY is updated to (LAST_ENTRY + 1) Modulo 512.
RDX	NUM_EXPORTED	If TDH.EXPORT.MEM is successful, RDX returns the number of exported 4KB migration buffers, including: • The GPA list page • Attributes list page (if GPA_LIST_INFO.FORMAT is GPA_AND_L2_ATTR) • One or two MAC pages (depending on GPA_LIST_INFO.FIRST_ENTRY and GPA_LIST_INFO.LAST_ENTRY • Up to 512 encrypted memory pages. Only the pages where actual data is exported are counted. See 3.1.3.3.6 below for details. If TDH.EXPORT.MEM is not successful, RDX is unmodified.

Operand	Name	Description
R8	LIST_ERR_COUNT	If TDH.EXPORT.MEM was called with version 1 or higher, R8 returns the number of GPA list entries where an error was encountered. Else, R8 is unmodified.
AVX, AVX2 and AVX512 state		May be reset to the architectural RESET state
Other		Unmodified

3.1.3.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.3.3.1. Overview

TDH.EXPORT.MEM exports a list of up to 512 TD private 4KB pages as a migration bundle, which includes an MBMD, set of 4KB pages encrypted with the migration session key, a 4KB page containing the GPA list, an optional 4KB page containing attributes list, and two 4KB pages containing page MACs.

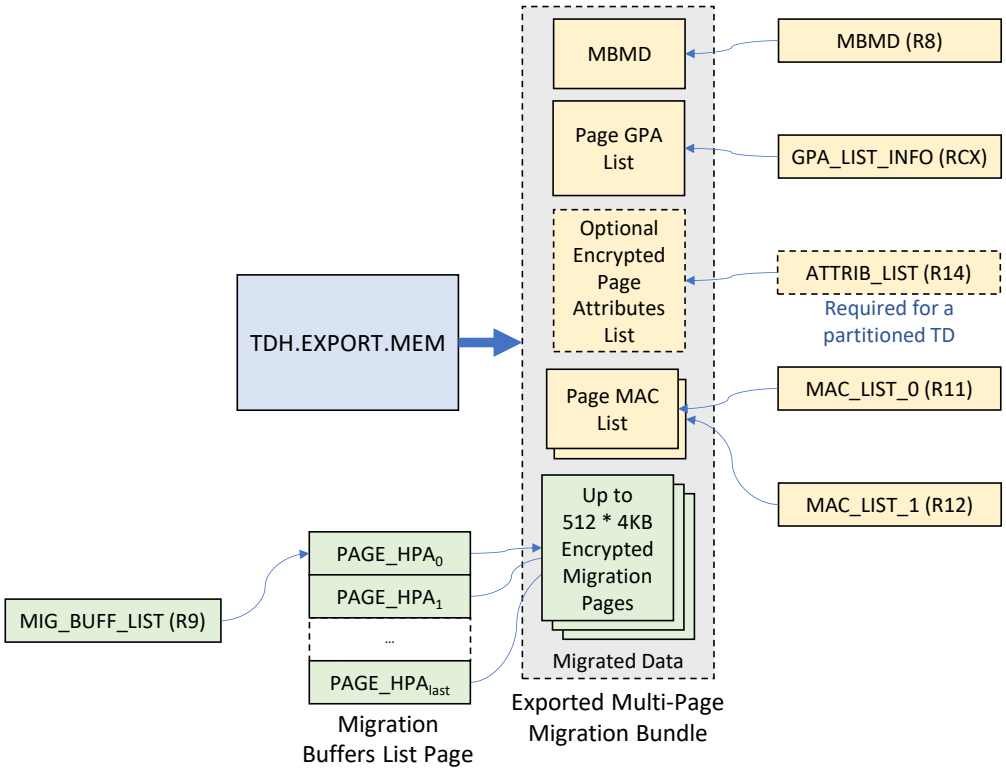

Figure 3.1: TDH.EXPORT.MEM Inputs and Outputs

Entries in the Page GPA List (pointed by RCX), Page Attributes List (pointed by R14), Page MAC List (pointed by R11 and R12) and Migration Buffers List (pointed by R9) are sorted in the same order. I.e., entry N in each of those lists applies to the same exported page.

3.1.3.3.2. Enumeration

Availability of TDH.EXPORT.MEM is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.EXPORT.MEM returns a TDX_OPERAND_INVALID(RAX) status.

TDX_FEATURES0.PARTITIONED_TD_MIGRATION (bit 21) enumerates TDX Module support of migrating partitioned TDs (which contain one or more L2 VMs).

Interruption due to yielding to concurrent functions that try to acquire an exclusive lock on the SEPT trees (see below) is supported if TDX_FEATURES0.NON_BLOCKING_EXPORT (bit 41) is set to 1.

Support of TDH.EXPORT.MEM version 1 or higher is enumerated by TDX_FEATURES0.LIST_ERROR_COUNT (bit 54).

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.3.3.3. Preconditions

- An export session must be in progress.
- The specified migration stream must have been created by TDH.MIG.STREAM.CREATE.

3.1.3.3.4. GPA List

A GPA list is provided as an input and is updated by TDH.EXPORT.MEM. The GPA list format is described in 2.2.2. Only 4KB pages are supported.

For write-blocking based export, the requested operation for each page in the list may be one of the following:

MIGRATE: Export the page. TDH.EXPORT.MEM decides, based on the SEPT entry state, whether the operation is an export or a re-export of a previously exported page. Note that the MIGRATE operation request is represented by two possible values.

CANCEL: Cancel a previous page export.

NOP: No operation.

For non-blocking export, the requested operation for each page in the list may be one of the following:

EXPORT: Export the page. TDH.EXPORT.MEM decides, based on the SEPT entry state, whether the operation is export, re-export of a previously exported page, or cancellation of a previous export.

NOP: No operation. Note that the NOP operation request is represented by three possible values.

TDH.EXPORT.MEM updates the GPA list with the actual OPERATION code (MIGRATE, REMIGRATE, CANCEL or NOP) and STATUS; the host VMM is expected to send the GPA list as part of the migration bundle, to be imported on the destination platform by TDH.IMPORT.MEM.

3.1.3.3.5. S4 Hibernation

If TDH.EXPORT.MEM is called as part of an S4 hibernation, it only supports write-blocking based export (though no write blocking is actually required) and the out-of-order export phase. As a result, the GPA list may not contain a CANCEL operation. In addition, blocking and TLB tracking is not required.

3.1.3.3.6. Migration Buffers List

A list of 4KB page buffers is provided as an input and is updated by TDH.EXPORT.MEM. In case no data is exported (PENDING page, page cancellation or some state error) TDH.EXPORT.MEM marks the applicable list entry as invalid by setting the INVALID bit (63). Such cases are not included in the NUM_EXPORTED output operand.

3.1.3.3.7. Write-Blocking and TLB Tracking (Write-Blocking based Export)

The following applies if the TDX module is configured for write-blocking based export:

If the TD may be running, the exported pages must be blocked for writing by TDH.EXPORT.BLOCKW and TLB tracked (TDH.MEM.TRACK followed by IPI to all the LPs running the TD VCPUs) to be exported. Unlike memory management operations such as TDH.MEM.PAGE.REMOVE, the TLB tracking is not page-specific; it should be done after the last TDH.EXPORT.BLOCKW of any page has been called for the current export round.

Else (e.g., the TD has been paused for export), no blocking and tracking is required.

3.1.3.3.8. Scanning and TLB Tracking (Non-Blocking Export)

The following applies if the TDX module is configured for non-blocking export:

If the TD may be running, the exported pages must have been scanned by TDH.MEM.SCAN(DSCAN) and TLB tracked to be exported. Unlike write-blocking based export, TLB tracking is page-specific, allowing TDH.MEM.SCAN(DSCAN) to run concurrently with TDH.EXPORT.MEM. For more information on TLB tracking, see the [TD Migration Spec].

Else (e.g., the TD has been paused for export), no tracking is required.

3.1.3.3.9. GPA List Entry Error Handling

If a page can't be exported for a reason that is specific to that page, TDH.EXPORT.MEM marks its GPA list entry as unsuccessful, but does not abort. It continues to the next entry, if applicable.

When a page is not exported, its GPA list entry is updated as follows:

- The OPERATION field is written as NOP, indicating that the page has not been exported.
- The STATUS field is written with the status code, as shown below.

Note: Skipping a GPA list entry whose OPERATION is NOP is not considered an error. The STATUS field is written as SKIPPED.

If TDH.EXPORT.MEM was called with version 1 or higher, R8 returns the number of GPA list entries where an error was encountered since it was last invoked, either as a new invocation or as a resumption of a previously interrupted operation. Else, R8 is unmodified.

The following table shows the possible GPA list entry STATUS values written by TDH.EXPORT.MEM. Refer to 2.2.2.4.7 for STATUS definition.

Table 3.12: GPA List Entry STATUS Values Returned by TDH.EXPORT.MEM

Value	Name	Description
0	SUCCESS	The GPA list entry was processed successfully.
1	SKIPPED	The GPA list entry was skipped because NOP was requested; this is not an error.
2	SEPT_WALK_FAILED	The GPA list entry was skipped due to an SEPT walk failure, e.g., some non-leaf SEPT entry at an upper level was blocked.
3	SEPT_ENTRY_BUSY_HOST_PRIORITY	The GPA list entry was skipped because of a conflict with a concurrent operation.
4	SEPT_ENTRY_STATE_INCORRECT	The GPA list entry was skipped due to an incorrect SEPT entry state. This may happen due to the following reasons: • For write-blocking export, MIGRATE or CANCEL operations are only allowed for certain page states. • A page cannot be exported if it is blocked (BLOCKED or PENDING_BLOCKED). Blocking is typically a temporary state, as preparation for some memory management operation. The host VMM should typically try to export the page later, after the memory management operation is done. • A page cannot be exported if it is not a TD private page (FREE or REMOVED). • MMIO pages cannot be exported.
5	TLB_TRACKING_NOT_DONE	The GPA list entry was skipped because TLB tracking was not done for the requested GPA.

Value	Name	Description
6	OP_STATE_INCORRECT	The GPA list entry was skipped because the TD's OP_STATE was incorrect for the requested operation and Secure EPT entry state. This may happen due to the following reasons: • Export is only allowed when a migration session is active. • For write-blocking export, CANCEL operation is only allowed during the in-order export phase.
8	MIG_BUFFER_NOT_AVAILABLE	The GPA list entry was skipped because the required migration buffer was not provided.
15	GPA_LIST_ENTRY_INVALID	The GPA list entry was skipped because it was invalid.
16	INVALID_MIGRATION_BUFFER_HPA	The GPA list entry was skipped because the provided migration buffer HPA is not valid.
17	PAGE_DIRTY	The GPA list entry was skipped and the page not exported because the SEPT entry's Dirty bit was set.
20	SCAN_EPOCH_NOT_RECORDED	The GPA list entry was skipped because TD Epoch value at the time of memory scan is not recorded. This may happen due to the following reasons: • No memory scan was done for this page. • Some other memory management operation (e.g., block) was done after the page had been scanned.

3.1.3.3.10. Concurrency

TDH.EXPORT.MEM may be called concurrently on multiple LPs (each specifying a separate migration stream) and may run concurrently with other functions (e.g., TDH.MEM.SCAN.*).

SEPT Tree Concurrency

TDH.EXPORT.MEM acquires a shared lock on the SEPT trees, to prevent blocking or changes to the tree structure (e.g., by TDH.MEM.RANGE.BLOCK or TDH.MEM.SEPT.REMOVE) while it runs. If supported, TDH.EXPORT.MEM may prevent starvation of concurrent memory management functions by detecting that they failed to acquire an exclusive lock on the SEPT trees. In this case, TDH.EXPORT.MEM yields and returns with a TDX_INTERRUPTED_BUSY status in RAX. The host VMM is expected to resume TDH.EXPORT.MEM. See the discussion on interruptibility below.

SEPT Entry Concurrency

Failure to acquire a lock on an SEPT entry is handled as a page-specific error, as described above. TDH.EXPORT.MEM skips the busy page.

3.1.3.3.11. Interruption and Resumption

TDH.EXPORT.MEM is interruptible. An interruption occurs in the following cases:

- If a pending interrupt is detected during operation, TDH.EXPORT.MEM returns with a TDX_INTERRUPTED_RESUMABLE status in RAX.
- If supported and if TDH.EXPORT.MEM detects that a concurrent function (such as TDH.MEM.PAGE.PROMOTE) has failed to acquire an exclusive lock on the SEPT trees, it may yield and return with a TDX_INTERRUPTED_BUSY status in RAX.

RCX is updated with the next list entry index to process.

The host VMM should re-invoke TDH.EXPORT.MEM after handling the interruption reason, keeping the same inputs (and updated value in RCX) except setting R10.RESUME to 1. If the host VMM cannot resume TDH.EXPORT.MEM for some reason, it should abort the export session (TDH.EXPORT.ABORT). Failing to resume TDH.EXPORT.MEM and to export the generated migration bundle when it completes successfully will result in the migration protocol going out of sync; this will be detected by the destination side, resulting in an import failure.

The host VMM should not transmit the generated migration bundle to the destination side until TDH.EXPORT.MEM is completed successfully. Doing so will result in the migration protocol going out of sync; this will be detected by the destination side, resulting in an import failure.

3.1.3.3.12. Interrupt Latency

TDH.EXPORT.MEM may exceed the normal interrupt response latency limit if the core frequency is minimal. However, since TD Migration is a CPU-intensive operation, this is condition is not expected to happen in real-life scenarios. See the [Base Spec] section titled “Latency of the Intel TDX Interface Functions”.

3.1.3.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.13: TDH.EXPORT.MEM Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	GPA List page	GPA_LIST	RW	Shared	4KB	None	None	None
Explicit	RDX	HPA	TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	R8	HPA	Memory to use for MBMD	MBMD	RW	Shared	128B	None	None	None
Explicit	R9	HPA	Migration buffer list	PAGE_LIST	RW	Shared	4KB	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Explicit	R11	HPA	MAC list page 1	MAC list	RW	Shared	4KB	None	None	None
Explicit	R12	HPA	MAC list page 2	MAC list	RW	Shared	4KB	None	None	None
Explicit	R14	HPA	attributes list page	page attributes	R	Shared	4KB	None	None	None
Explicit	N/A	GPA	TD private pages (via GPA list)	Blob	R	Private	4KB	None	None	None
Explicit	N/A	HPA	Migration buffer pages (via page list)	Blob	RW	Shared	4KB	None	None	None
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT tree	N/A	RW	Private	N/A	Shared	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT entry	SEPT Entry	RW	Private	N/A	Exclusive(h)	N/A	N/A

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	GPA	L2 Secure EPT trees	N/A	RW	Private	N/A	Shared(i)	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT entries	SEPT Entry	RW	Private	N/A	Exclusive(i)	N/A	N/A

3.1.3.5. Completion Status Codes

Table 3.14: TDH.EXPORT.MEM Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_INCOMPATIBLE_EXPORT_MODE_TD_NON_ACCESSIBLE	
TDX_INTERRUPTED_BUSY	
TDX_INTERRUPTED_RESUMABLE	
TDX_INVALID_RESUMPTION	
TDX_MIGRATION_STREAM_STATE_INCORRECT	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SUCCESS	The operation is successful. **Note:** Processing of some GPA list entries may have encountered errors, but this did not cause an abort of the overall operation. The number of such errors is reported in R8.
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.4. TDH.EXPORT.PAUSE Leaf

TDH.EXPORT.PAUSE starts the TDX-enforced blackout period on the source platform, where the source TD is paused.

3.1.4.1. Input Operands

Table 3.15: TDH.EXPORT.PAUSE Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 70
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	HPA of Source TD TDR page (HKID bits must be 0)		

3.1.4.2. Output Operands

Table 3.16: TDH.EXPORT.PAUSE Output Operands Definitions

Operand	Description
RAX	SEAMCALL instruction return code
Other	Unmodified

3.1.4.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.4.3.1. Overview

TDH.EXPORT.PAUSE pauses the TD and starts the Live Migration Blackout period on the source platform.

- All TD VCPUs have stopped executing.
- No TD-specific SEAMCALL is running.

3.1.4.3.2. Enumeration

Availability of TDH.EXPORT.PAUSE is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.EXPORT.PAUSE returns a TDX_OPERAND_INVALID(RAX) status.

3.1.4.3.3. Preconditions

TDH.EXPORT.PAUSE checks the following preconditions:

- An export session must be in progress, and the guest TD must not have been paused by TDH.EXPORT.PAUSE, i.e., its OP_STATE is LIVE_EXPORT.
- If the TDX module is configured for non-blocking export, no TD private page may be blocked.
- If the TD is configured for TDX Connect, no TDIs may be attached.

3.1.4.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.17: TDH.EXPORT.PAUSE Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	R	Opaque	4KB	Exclusive	Shared	Shared
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	TDCS Epoch Tracking Fields	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A

5

3.1.4.5. Completion Status Codes

Table 3.18: TDH.EXPORT.PAUSE Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_BLOCKED_MEMORY_EXIST	There are memory pages that have been blocked by TDH.MEM.RANGE.BLOCK. This is not permitted if the TDX module is configured for non-blocking export.
TDX_IOMMU_IOTLB_TRACKING_NOT_DONE	Applies only if TDX Connect is enabled for this TD
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SUCCESS	Operation is successful
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_HAS_ATTACHED_DEVICES	Applies only if TDX Connect is enabled for this TD
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.5. TDH.EXPORT.RESTORE Leaf

TDH.EXPORT.RESTORE restores a list of TD private 4KB pages' Secure EPT entry states after an export abort.

3.1.5.1. Input Operands

Table 3.19: TDH.EXPORT.RESTORE Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 66
		23:16	Version Number	Selects the SEAMCALL interface function version Version may be 0 or 1. See enumeration details below.
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	GPA_LIST_INFO	GPA_LIST_INFO: HPA of a GPA list page in shared memory, and first and last entries to process, as defined in 2.2.2 FORMAT must be GPA_ONLY.		
RDX	TDR	HPA of the source TD's TDR page (HKID bits must be 0)		

5

3.1.5.2. Output Operands

Table 3.20: TDH.EXPORT.RESTORE Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RCX	GPA_LIST_INFO	Same as the input value, except that FIRST_ENTRY is updated to the index of the next entry to be processed. If all entries have been processed, FIRST_ENTRY is updated to (LAST_ENTRY + 1) Modulo 512.
R8	LIST_ERR_COUNT	If TDH.EXPORT.RESTORE was called with version 1 or higher, R8 returns the number of GPA list entries where an error was encountered. Else, R8 is unmodified.
Other		Unmodified

3.1.5.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.5.3.1. Overview

TDH.EXPORT.RESTORE restores a list of TD private 4KB pages' Secure EPT entry states after an aborted export session. It reverts each L1 Secure EPT entry and any applicable L2 Secure EPT entries to their original non-exported state.

3.1.5.3.2. Enumeration

Availability of TDH.EXPORT.RESTORE is enumerated by TDX_FEATURES0.TD_MIGRATION (bit 0), readable by TDH.SYS.RD*. If not supported, calling TDH.EXPORT.RESTORE returns a TDX_OPERAND_INVALID(RAX) status.

Support of TDH.EXPORT.RESTORE version 1 or higher is enumerated by TDX_FEATURES0.LIST_ERROR_COUNT (bit 54).

3.1.5.3.3. Preconditions

The guest TD has been initialized or imported, and no migration session is in progress, i.e., the TD's OP_STATE is RUNNABLE.

3.1.5.3.4. GPA List Entry Error Handling

If a page's Secure EPT entry can't be restored, TDH.EXPORT.RESTORE marks its GPA list entry as unsuccessful. List process is not aborted; it continues to the next entry, if applicable.

Note: Skipping a GPA list entry whose OPERATION is NOP is not considered an error. The STATUS field is written as SKIPPED.

If TDH.EXPORT.RESTORE was called with version 1 or higher, RDX returns the number of GPA list entries where an error was encountered. Else, RDX is unmodified.

The following table shows the possible GPA list entry STATUS values written by TDH.EXPORT.RESTORE. Refer to 2.2.2.4.7 for STATUS definition.

Table 3.21: GPA List Entry STATUS Values Returned by TDH.EXPORT.RESTORE

Value	Name	Description
0	SUCCESS	The GPA list entry was processed successfully.
1	SKIPPED	The GPA list entry was skipped because NOP was requested; this is not an error.
2	SEPT_WALK_FAILED	The GPA list entry was skipped due to an SEPT walk failure, e.g., some non-leaf SEPT entry at an upper level was blocked.
3	SEPT_ENTRY_BUSY_HOST_PRIORITY	The GPA list entry was skipped because of a conflict with a concurrent operation.
4	SEPT_ENTRY_STATE_INCORRECT	The GPA list entry was skipped due to an incorrect SEPT entry state. If the SEPT entry state is BLOCKEDW or PENDING_BLOCKEDW, i.e., the page has been blocked by TDH.EXPORT.BLOCKW but has not yet been exported, the host VMM should call TDH.EXPORT.UNBLOCKW to restore the SEPT entry. This is only applicable if the TDX Module is configured for write-blocking export mode. Else, there is no need to restore the SEPT entry.
15	GPA_LIST_ENTRY_INVALID	The GPA list entry is invalid. TDH.EXPORT.RESTORE terminated.

3.1.5.3.5. Interruptibility

TDH.EXPORT.RESTORE is interruptible. If a pending interrupt is detected during operation, TDH.EXPORT.RESTORE returns with a TDX_INTERRUPTED_RSUMABLE status in RAX. RCX is updated with the next list entry index to process, so the host VMM may re-invoke TDH.EXPORT.RESTORE immediately after handling the interrupt.

3.1.5.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.22: TDH.EXPORT.RESTORE Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	GPA List page	GPA_LIST	RW	Shared	4KB	None	None	None
Explicit	RDX	HPA	TDR page	TDR	RW	Opaque	4KB	Shared	Shared	Shared
Explicit	N/A	GPA	TD private pages (via GPA list)	Block	None	Private	4KB	None	None	None
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	None	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT tree	N/A	RW	Private	N/A	Shared	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT entry	SEPT Entry	RW	Private	N/A	Exclusive(h)	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT trees	N/A	RW	Private	N/A	Shared(i)	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT entries	SEPT Entry	RW	Private	N/A	Exclusive(i)	N/A	N/A

3.1.5.5. Completion Status Codes

Table 3.23: TDH.EXPORT.RESTORE Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_INCOMPATIBLE_EXPORT_MODE_TD_NON_ACCESSIBLE	
TDX_INTERRUPTED_RESUMABLE	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	

Completion Status Code	Description
TDX_SUCCESS	The operation is successful. **Note:** Processing of some GPA list entries may have encountered errors, but this did not cause an abort of the overall operation. The number of such errors is reported in R8.
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.6. TDH.EXPORT.STATE.IMMUTABLE Leaf

TDH.EXPORT.STATE.IMMUTABLE starts a new export session and exports the TD's immutable state as a multi-page migration bundle.

TDH.EXPORT.STATE.IMMUTABLE is also used for starting a new S4 hibernation session.

3.1.6.1. Input Operands

Table 3.24: TDH.EXPORT.STATE.IMMUTABLE Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 72
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	TDR	Source TD handle and flags		
		Bits	**Name**	**Description**
		0	EXPORT_TYPE	0: TD Export 1: S4 Hibernation
		11:1	Reserved	Must be 0
		51:12	TDR HPA	HPA[51:12] of the source TD's TDR page (HKID bits must be 0)
		63:52	Reserved	Must be 0
R8	MBMD	HPA and size of memory of a memory buffer to use for MBMD:		
		Bits	**Name**	**Description**
		51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
		63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.

Operand	Name	Description		
R9	PAGE_LIST_INFO	Migration buffers list information, in HPA_LIST_INFO format (see the [ABI Spec]) FORMAT (bits 2:0) and FIRST_ENTRY (bits 11:3) must be 0. See enumeration details below.		
R10	MIG_STREAM	Migration stream and resume flag:		
		Bits	**Name**	**Description**
		15:0	MIGS_INDEX	Migration stream index – must be 0
		62:16	RESERVED	Reserved: must be 0
		63	RESUME	0: This is a new invocation 1: This is resumption of a previously interrupted operation

3.1.6.2. Output Operands

Table 3.25: TDH.EXPORT.STATE.IMMUTABLE Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RDX	NUM_EXPORTED	If STATUS is TDX_SUCCESS, RDX returns the number of exported 4KB migration buffers. Else, RDX value should be ignored.
AVX, AVX2 and AVX512 state		May be reset to the architectural RESET state
Other		Unmodified

3.1.6.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.6.3.1. Overview

TDH.EXPORT.STATE.IMMUTABLE starts a new export session. It exports the TD's immutable state as a migration bundle, which includes an MBMD and a set of 4KB pages, encrypted with the migration session key. The migration bundle is protected by a MAC that is stored in the MBMD.

TDH.EXPORT.STATE.IMMUTABLE is also used for starting a new S4 hibernation session.

3.1.6.3.2. Enumeration

Availability of TDH.EXPORT.STATE.IMMUTABLE is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.EXPORT.STATE.IMMUTABLE returns a TDX_OPERAND_INVALID(RAX) status.

The required number of 4KB pages for exporting the immutable TD state is enumerated by NUM_IMMUTABLE_STATE_PAGES.

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.6.3.3. Preconditions

- The TD build and measurement have been finalized, or the TD has been imported, and no export session is in progress, i.e., the TD's OP_STATE is either RUNNABLE or LIVE_IMPORT.
- The TD is migratable: its ATTRIBUTES.MIGRATABLE is set to 1.
- Any previous aborted export session has been cleaned up by TDH.EXPORT.RESTORE, TDH.MEM.SCAN(EXPORT_RESTORE) and/or TDH.EXPORT.UNBLOCKW.
- A Migration TD has been bound to the TD.
- The migration TD has written the migration decryption key and migration protocol version metadata fields.
- Migration stream 0 has been created by TDH.MIG.STREAM.CREATE.

3.1.6.3.4. Interruptibility

If a pending interrupt is detected during operation, TDH.EXPORT.STATE.IMMUTABLE returns with a TDX_INTERRUPTED_RESUMABLE status in RAX. The host VMM is expected to resume it, by calling it with R10.RESUME set to 1. If the host VMM cannot resume TDH.EXPORT.STATE.IMMUTABLE for some reason, it should abort the export session (TDH.EXPORT.ABORT).

3.1.6.3.5. Aspects of MAC Calculation Compatibility on TDX Module Update

If TDH.EXPORT.STATE.IMMUTABLE was interrupted, and later the TDX Module was updated before TDH.EXPORT.STATE.IMMUTABLE was resumed, there could be incompatibility issues with the format of the intermediate MRTD MAC context, held as part of the migration stream context (MIGSC), between the original and the new TD module.

On TDH.SYS.UPDATE, the host VMM can configure the TDX Module to detect such incompatibility. If detected, TDH.EXPORT.STATE.IMMUTABLE returns a TDX_INCOMPATIBLE_MBMD_MAC_CONTEXT status. In this case, the host VMM is expected to abort the migration session and may restart it. For details, see the [Base Spec] section titled "Compatibility Aspects of TD-Preserving Update".

If the host VMM did not configure the TDX Module to detect such incompatibility, it will not be detected by TDH.EXPORT.STATE.IMMUTABLE. However, on import, TDH.IMPORT.STATE.IMMUTABLE will detect that the MAC is incorrect and will return a TDX_INCORRECT_MBMD_MAC error.

3.1.6.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.26: TDH.EXPORT.STATE.IMMUTABLE Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	RW	Opaque	4KB	Shared	Shared	Shared
Explicit	R8	HPA	Memory to use for MBMD	MBMD	RW	Shared	128B	None	None	None
Explicit	R9	HPA	Page list	PAGE_LIST	RW	Shared	4KB	None	None	None
Explicit	R10	N/A	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Explicit	N/A	HPA	Destination pages (via page list)	Blob	RW	Shared	4KB	None	None	None
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive	N/A	N/A

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	PL.S4_STATE	N/A	RW	Hidden	N/A	Exclusive	N/A	N/A

3.1.6.5. Completion Status Codes

Table 3.27: TDH.EXPORT.STATE.IMMUTABLE Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_INCOMPATIBLE_EXPORT_MODE_TD_NON_ACCESSIBLE	
TDX_INCOMPATIBLE_MBMD_MAC_CONTEXT	
TDX_INTERRUPTED_RESUMABLE	
TDX_INVALID_RESUMPTION	
TDX_IOMMU_IOTLB_TRACKING_NOT_DONE	Applicable only if TDX Connect is supported
TDX_MAX_EXPORTS_EXCEEDED	
TDX_METADATA_LIST_OVERFLOW	
TDX_MIGRATION_DECRYPTION_KEY_NOT_SET	
TDX_MIGRATION_STREAM_STATE_INCORRECT	
TDX_MIN_MIGS_NOT_CREATED	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_PREVIOUS_EXPORT_CLEANUP_INCOMPLETE	
TDX_RND_NO_ENTROPY	Failed to generate a random migration encryption key. This is typically caused by an entropy error of the CPU's random number generator, and may be impacted by RDSEED, RDRAND or PCONFIG executing on other LPs. The operation should be retried.
TDX_SUCCESS	Operation is successful
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_HAS_ATTACHED_DEVICES	Applicable only if TDX Connect is supported
TDX_TD_KEYS_NOT_CONFIGURED	

Completion Status Code	Description
TDX_TD_NOT_MIGRATABLE	
TDX_TDCS_NOT_ALLOCATED	

3.1.7. TDH.EXPORT.STATE.TD Leaf

TDH.EXPORT.STATE.TD exports a paused TD's mutable state as a multi-page migration bundle.

3.1.7.1. Input Operands

Table 3.28: TDH.EXPORT.STATE.TD Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 73
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	TDR	HPA of the source TD's TDR page (HKID bits must be 0)		
R8	MBMD	HPA and size of memory of a memory buffer to use for MBMD:		
		Bits	**Name**	**Description**
		51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
		63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.
R9	PAGE_LIST_INFO	Migration buffers list information, in HPA_LIST_INFO format (see the [ABI Spec]) FORMAT (bits 2:0) and FIRST_ENTRY (bits 11:3) must be 0. See enumeration details below.		
R10	MIG_STREAM	Migration stream and resume flag:		
		Bits	**Name**	**Description**
		15:0	MIGS_INDEX	Migration stream index – must be 0
		62:16	RESERVED	Reserved: must be 0
		63	RESUME	0: This is a new invocation 1: This is resumption of a previously interrupted operation

5

3.1.7.2. Output Operands

Table 3.29: TDH.EXPORT.STATE.TD Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RDX	NUM_EXPORTED	If STATUS is TDX_SUCCESS, RDX returns the number of exported 4KB migration buffers. Else, RDX value should be ignored.
AVX, AVX2 and AVX512 state		May be reset to the architectural RESET state
Other		Unmodified

3.1.7.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.7.3.1. Overview

TDH.EXPORT.STATE.TD exports the TD's mutable state as a migration bundle, which includes an MBMD and a set of 4KB pages, encrypted with the migration session key. The migration bundle is protected by a MAC that is stored in the MBMD. The TD must have been paused by a TDH.EXPORT.PAUSE.

3.1.7.3.2. Enumeration

Availability of TDH.EXPORT.STATE.TD is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.EXPORT.STATE.TD returns a TDX_OPERAND_INVALID(RAX) status.

The required number of 4KB pages for exporting the mutable TD state is enumerated by NUM_TD_STATE_PAGES.

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.7.3.3. Preconditions

An export session is in progress, and the TD has been paused: its OP_STATE is PAUSED_EXPORT.

3.1.7.3.4. Interruptibility

If a pending interrupt is detected during operation, TDH.EXPORT.STATE.TD returns with a TDX_INTERRUPTED_RESUMABLE status in RAX. The host VMM is expected to resume it, by calling it with R10.RESUME set to 1. If the host VMM cannot resume TDH.EXPORT.STATE.TD for some reason, it should abort the export session (TDH.EXPORT.ABORT).

3.1.7.3.5. Aspects of MAC Calculation Compatibility on TDX Module Update

If TDH.EXPORT.STATE.TD was interrupted, and later the TDX Module was updated before TDH.EXPORT.STATE.TD was resumed, there could be incompatibility issues with the format of the intermediate MRTD MAC context, held as part of the migration stream context (MIGSC), between the original and the new TD module.

On TDH.SYS.UPDATE, the host VMM can configure the TDX Module to detect such incompatibility. If detected, TDH.EXPORT.STATE.TD returns a TDX_INCOMPATIBLE_ MBMD_MAC_CONTEXT status. In this case, the host VMM is expected to abort the migration session and may restart it. For details, see the [Base Spec] section titled "Compatibility Aspects of TD-Preserving Update".

If the host VMM did not configure the TDX Module to detect such incompatibility, it will not be detected by TDH.EXPORT.STATE.TD. However, on import, TDH.IMPORT.STATE.TD will detect that the MAC is incorrect and will return a TDX_INCORRECT_MBMD_MAC error.

3.1.7.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.30: TDH.EXPORT.STATE.TD Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	R8	HPA	Memory to use for MBMD	MBMD	RW	Shared	128B	None	None	None
Explicit	R9	HPA	Page list	PAGE_LIST	R	Shared	4KB	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Explicit	N/A	HPA	Destination pages (via page list)	Blob	RW	Shared	4KB	None	None	None
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Exclusive(h)	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	None	N/A	N/A

5

3.1.7.5. Completion Status Codes

Table 3.31: TDH.EXPORT.STATE.TD Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_INCOMPATIBLE_MBMD_MAC_CONTEXT	
TDX_INTERRUPTED_RESUMABLE	
TDX_INVALID_RESUMPTION	
TDX_METADATA_LIST_OVERFLOW	
TDX_MIGRATION_STREAM_STATE_INCORRECT	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	

Completion Status Code	Description
TDX_SUCCESS	Operation is successful
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.8. TDH.EXPORT.STATE.VP Leaf

TDH.EXPORT.STATE.VP exports a paused TD's VCPU mutable state as a multi-page migration bundle.

3.1.8.1. Input Operands

Table 3.32: TDH.EXPORT.STATE.VP Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 74
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	TDVPR	HPA of the source TD VCPU's TDVPR page (HKID bits must be 0)		
R8	MBMD	HPA and size of memory of a memory buffer to use for MBMD:		
		Bits	**Name**	**Description**
		51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
		63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.
R9	PAGE_LIST_INFO	Migration buffers list information, in HPA_LIST_INFO format (see the [ABI Spec]) FORMAT (bits 2:0) and FIRST_ENTRY (bits 11:3) must be 0. See enumeration details below.		
R10	MIG_STREAM	Migration stream and resume flag:		
		Bits	**Name**	**Description**
		15:0	MIGS_INDEX	Migration stream index If N migration streams have been created by TDH.MIG.STREAM.CREATE, then MIGS_INDEX must be lower than N-1.
		62:16	RESERVED	Reserved: must be 0

Operand	Name	Description		
		63	RESUME	0: This is a new invocation 1: This is resumption of a previously interrupted operation

3.1.8.2. Output Operands

Table 3.33: TDH.EXPORT.STATE.VP Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RDX	NUM_EXPORTED	If STATUS is TDX_SUCCESS, RDX returns the number of exported 4KB migration buffers. Else, RDX value should be ignored.
AVX, AVX2 and AVX512 state		May be reset to the architectural RESET state
Other		Unmodified

3.1.8.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.8.3.1. Overview

TDH.EXPORT.STATE.VP exports a TD's VCPU mutable state as a migration bundle, which includes an MBMD and a set of 4KB pages, encrypted with the migration session key. The migration bundle is protected by a MAC that is stored in the MBMD. The TD must have been paused by a TDH.EXPORT.PAUSE.

3.1.8.3.2. Enumeration

Availability of TDH.EXPORT.STATE.VP is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.EXPORT.STATE.VP returns a TDX_OPERAND_INVALID(RAX) status.

The required number of 4KB pages for exporting the mutable VCPU state is enumerated by NUM_VP_STATE_PAGES.

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.8.3.3. Preconditions

- The specified migration stream has been created by TDH.MIG.STREAM.CREATE.
- An export session is in progress.
- The TD has been paused by TDH.EXPORT.PAUSE.
- The mutable TD-scope state has been exported by TDH.EXPORT.STATE.TD.

3.1.8.3.4. Interruptibility

If a pending interrupt is detected during operation, TDH.EXPORT.STATE.VP returns with a TDX_INTERRUPTED_RESUMABLE status in RAX. The host VMM is expected to resume it, by calling it with R10.RESUME set to 1. If the host VMM cannot resume TDH.EXPORT.STATE.VP for some reason, it should abort the export session (TDH.EXPORT.ABORT).

3.1.8.3.5. Aspects of MAC Calculation Compatibility on TDX Module Update

If TDH.EXPORT.STATE.VP was interrupted, and later the TDX Module was updated before TDH.EXPORT.STATE.VP was resumed, there could be incompatibility issues with the format of the intermediate MRTD MAC context, held as part of the migration stream context (MIGSC), between the original and the new TD module.

On TDH.SYS.UPDATE, the host VMM can configure the TDX Module to detect such incompatibility. If detected, TDH.EXPORT.STATE.VP returns a TDX_INCOMPATIBLE_ MBMD_MAC_CONTEXT status. In this case, the host VMM is expected to abort the migration session and may restart it. For details, see the [Base Spec] section titled "Compatibility Aspects of TD-Preserving Update".

If the host VMM did not configure the TDX Module to detect such incompatibility, it will not be detected by TDH.EXPORT.STATE.VP. However, on import, TDH.IMPORT.STATE.VP will detect that the MAC is incorrect and will return a TDX_INCORRECT_MBMD_MAC error.

3.1.8.3.6. VCPU Association

TDH.EXPORT.VP associates the TD VCPU with the current LP. This requires that the VCPU will not be associated with another LP – for details, see the [TDX Module Base Spec].

3.1.8.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.34: TDH.EXPORT.STATE.VP Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDVPR page	TDVPS	RW	Opaque	4KB	Exclusive	Shared	Shared
Explicit	R8	HPA	Memory to use for MBMD	MBMD	RW	Shared	128B	None	None	None
Explicit	R9	HPA	Page list	PAGE_LIST	R	Shared	4KB	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Explicit	N/A	HPA	Destination pages (via page list)	Blob	RW	Shared	4KB	None	None	None
Implicit	N/A	HPA	TDR page	TDR	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	None	N/A	N/A

3.1.8.5. Completion Status Codes

Table 3.35: TDH.EXPORT.STATE.VP Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_INCOMPATIBLE_MBMD_MAC_CONTEXT	

Completion Status Code	Description
TDX_INTERRUPTED_RESUMABLE	
TDX_INVALID_RESUMPTION	
TDX_METADATA_LIST_OVERFLOW	
TDX_MIGRATION_STREAM_STATE_INCORRECT	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SUCCESS	Operation is successful
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	
TDX_VCPU_ALREADY_EXPORTED	
TDX_VCPU_STATE_INCORRECT	The VCPU is expected to be in the VCPU_READY state. For details, see the [Base Spec] discussion of VPCU transitions.

3.1.9. TDH.EXPORT.TRACK Leaf

TDH.EXPORT.TRACK ends the current export epoch and starts a new one. It either starts a new in-order phase epoch or starts the out-of-order phase. In both cases, TDH.EXPORT.TRACK generates an epoch token to be exported to the destination platform.

3.1.9.1. Input Operands

Table 3.36: TDH.EXPORT.TRACK Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 71
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	HPA of the source TD's TDR page (HKID bits must be 0)		
R8	HPA and size of memory of a memory buffer to use for MBMD:		
	Bits	**Name**	**Description**
	51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
	63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.
R10	Migration stream and flags:		
	Bits	**Name**	**Description**
	15:0	MIGS_INDEX	Migration stream index – must be 0
	62:16	RESERVED	Reserved: must be 0
	63	IN_ORDER_DONE	Indicates that the in-order export phase is done, and a start token should be generated

3.1.9.2. Output Operands

Table 3.37: TDH.EXPORT.TRACK Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code

Operand	Description
AVX, AVX2 and AVX512 state	May be reset to the architectural RESET state
Other	Unmodified

3.1.9.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.9.3.1. Overview

If R10.IN_ORDER_DONE is 0, TDH.EXPORT.TRACK starts a new export epoch.

Else (R10.IN_ORDER_DONE is 1), TDH.EXPORT.TRACK checks that no memory exported so far needs to be re-exported. If so, it ends the in-order export phase and starts the out-of-order phase.

In both cases, TDH.EXPORT.TRACK generates an epoch token, to be exported on the specified migration stream.

When called as part of S4 hibernation, R10.IN_ORDER_DONE must be 1.

3.1.9.3.2. Enumeration

Availability of TDH.EXPORT.TRACK is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.EXPORT.TRACK returns a TDX_OPERAND_INVALID(RAX) status.

3.1.9.3.3. Preconditions

- An export session is in progress.
- A start token has not been generated during this export session: TDH.EXPORT.TRACK has not been called with IN_ORDER_DONE set to 1.

Additional preconditions for start token generation are discussed below.

3.1.9.3.4. Epoch Token Generation

TDH.EXPORT.TRACK generates an epoch token, to be exported. The epoch token, once successfully imported on the destination side, indicates that a new migration epoch has started.

3.1.9.3.5. End of the In-Order Phase and Start Token Generation

An R10.IN_ORDER_DONE value of 1 indicates that TDH.EXPORT.TRACK is requested to end the in-order export phase, start the out-of-order phase and generate a start token to be exported. A start token is a private case of an epoch token. Once successfully imported on the destination side, the start token enables the host VMM to commit the migration and start running the TD on the destination.

TDH.EXPORT.TRACK checks the completeness of memory export:

- TDH.EXPORT.TRACK checks that the TD has been paused by TDH.EXPORT.PAUSE.
- If the TDX module is configured for non-blocking export, TDH.EXPORT.TRACK checks that a comprehensive scan was successfully done using TDH.MEM.SCAN.COMP(DCHECK).
- TDH.EXPORT.TRACK checks that no TD private memory exported so far needs to be re-exported.
- If either the TD is configured for TDX Connect or the TDX module does not support post-copy (this is the case if the TDX module is configured for non-blocking export), TDH.EXPORT.TRACK checks that all the TD private memory has been exported.

When called as part of S4 hibernation, R10.IN_ORDER_DONE must be 1.

3.1.9.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.38: TDH.EXPORT.TRACK Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	R8	HPA	Memory to use for MBMD	MBMD	RW	Shared	128B	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	Comprehensive scan state	N/A	R	Opaque	N/A	Shared	N/A	N/A

5

3.1.9.5. Completion Status Codes

Table 3.39: TDH.EXPORT.TRACK Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_EXPORTED_DIRTY_PAGES_REMAIN	
TDX_MEM_SCAN_DCHECK_NOT_DONE	
TDX_MIGRATION_EPOCH_OVERFLOW	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX _PAGE_METADATA_INCORRECT	
TDX_SUCCESS	Operation is successful
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	
TDX_UNEXPORTED_MEMORY_REMAINS	

TD Migration Application Binary Interface (ABI) Reference

3.1.10. TDH.EXPORT.UNBLOCKW Leaf

Remove the write-blocking of a 4KB TD private page previously blocked by TDH.EXPORT.BLOCKW.

3.1.10.1. Input Operands

Table 3.40: TDH.EXPORT.UNBLOCKW Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 75
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	EPT mapping information:		
	Bits	**Name**	**Description**
	2:0	Level	Level of the Secure EPT entry that maps the page to be blocked for writing – see the [ABI Spec]: must be 0 (4KB)
	11:3	Reserved	Reserved: must be 0
	51:12	GPA	Bits 51:12 of the GPA to be unblocked for writing
	63:52	Reserved	Reserved: must be 0
RDX	Host physical address of the parent TDR page (HKID bits must be 0)		

3.1.10.2. Output Operands

Table 3.41: TDH.EXPORT.UNBLOCKW Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
RCX	Extended error information part 1 In case of EPT walk error, Secure EPT entry where the error was detected In other cases, RCX returns 0
RDX	Extended error information part 2 In case of EPT walk error, EPT level where the error was detected In other cases, RDX returns 0
Other	Unmodified

3.1.10.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.10.3.1. Overview

TDH.EXPORT.UNBLOCKW finds the write blocked Secure EPT entry for the given GPA and level. It verifies that the entry has been blocked for writing and TLB tracking has been done, then marks the entry as non-blocked for writing (MAPPED, PENDING, EXPORTED_DIRTY or PENDING_EXPORTED_DIRTY as appropriate). If the page has any L2 mappings, TDH.EXPORT.UNBLOCKW unblocks them.

3.1.10.3.2. Enumeration

Availability of TDH.EXPORT.UNBLOCKW is enumerated by TDX_FEATURES0.TD_MIGRATION (bit 0), readable by TDH.SYS.RD*. If not supported, calling TDH.EXPORT.UNBLOCKW returns a TDX_OPERAND_INVALID(RAX) status.

3.1.10.3.3. Preconditions

Either one of the following is true:

- An export session is in progress, or
- The TD is allowed to run (its OP_STATE is either RUNNABLE, LIVE EXPORT, PAUSED_EXPORT or POST_EXPORT). In these states, TDH.EXPORT.UNBLOCKW is used to clean up after an aborted export session.

3.1.10.3.4. Export Mode

TDH.EXPORT.UNBLOCKW is only available if the TDX module is configured for write-blocking based export (the default). If not available, calling TDH.EXPORT.UNBLOCKW returns a TDX_OPERAND_INVALID(RAX) status.

3.1.10.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.42: TDH.EXPORT.UNBLOCKW Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	GPA and Level	Secure EPT page or TD private page	Blob	None	Private	$2^{12+9*Level}$ Bytes	None	None	None
Explicit	RDX	HPA	TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	None	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT tree	N/A	RW	Private	N/A	Shared	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT entry	SEPT Entry	RW	Private	N/A	Exclusive(h)	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT trees	N/A	RW	Private	N/A	Shared(i)	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT entries	SEPT Entry	RW	Private	N/A	Exclusive(i)	N/A	N/A

3.1.10.5. Completion Status Codes

Table 3.43: TDH.EXPORT.UNBLOCKW Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_EPT_ENTRY_STATE_INCORRECT	
TDX_EPT_WALK_FAILED	
TDX_NOT_WRITE_BLOCKED	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SUCCESS	TDH.EXPORT.UNBLOCKW is successful
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	
TDX_TLB_TRACKING_NOT_DONE	

3.1.11. TDH.IMPORT.ABORT Leaf

Abort an import session; after this the target TD can only be destroyed. Generate an abort token that is to be consumed by the source platform.

3.1.11.1. Input Operands

Table 3.44: TDH.IMPORT.ABORT Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 80
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	HPA of the destination TD's TDR page (HKID bits must be 0)		
R8	HPA and size of memory of a memory buffer to use for MBMD:		
	Bits	**Name**	**Description**
	51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
	63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.
R10	Migration stream index – must be 0		

3.1.11.2. Output Operands

Table 3.45: TDH.IMPORT.ABORT Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
AVX, AVX2 and AVX512 state	May be reset to the architectural RESET state
Other	Unmodified

3.1.11.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.11.3.1. Overview

TDH.IMPORT.ABORT generates an abort token MBMD and sets the destination TD's OP_STATE to IMPORT_FAILED. In this state, the destination TD will not run; it can only be destroyed. This is indicated by the FATAL bit (61) of the completion status returned in RAX.

3.1.11.3.2. Enumeration

Availability of TDH.IMPORT.ABORT is enumerated by TDX_FEATURES0.TD_MIGRATION (bit 0), readable by TDH.SYS.RD*. If not supported, calling TDH.IMPORT.ABORT returns a TDX_OPERAND_INVALID(RAX) status.

3.1.11.3.3. Preconditions

An import session is in progress but has not been committed yet by TDH.IMPORT.COMMIT.

3.1.11.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.46: TDH.IMPORT.ABORT Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	R	Opaque	4KB	Exclusive	Shared	Shared
Explicit	R8	HPA	Memory to use for MBMD	MBMD	RW	Shared	128B	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A

3.1.11.5. Completion Status Codes

The table below details the status values returned in RAX. The FATAL bit (61) indicates that the import session has been aborted. In such cases, the status code name is appended with a "_FATAL" suffix.

Table 3.47: TDH.IMPORT.ABORT Completion Status Codes (Returned in RAX) Definition

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_OP_STATE_INCORRECT	0	
TDX_OPERAND_ADDR_RANGE_ERROR	0	
TDX_OPERAND_BUSY	0	
TDX_OPERAND_INVALID	0	
TDX_PAGE_METADATA_INCORRECT	0	
TDX_SUCCESS_FATAL	1	TDH.IMPORT.ABORT is successful, the import session is aborted, and the TD should be torn down

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_SYS_NOT_READY	0	
TDX_SYS_SHUTDOWN	0	
TDX_TD_FATAL	1	
TDX_TD_KEYS_NOT_CONFIGURED	0	
TDX_TDCS_NOT_ALLOCATED	0	

3.1.12. TDH.IMPORT.COMMIT Leaf

Commit an import session and allow the imported TD to run.

3.1.12.1. Input Operands

Table 3.48: TDH.IMPORT.COMMIT Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 82
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	HPA of the destination TD's TDR page (HKID bits must be 0)		

3.1.12.2. Output Operands

Table 3.49: TDH.IMPORT.COMMIT Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
Other	Unmodified

3.1.12.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.12.3.1. Overview

TDH.IMPORT.COMMIT commits an import session and allows the important TD to run. Post-copy memory import may continue.

3.1.12.3.2. Enumeration

Availability of TDH.IMPORT.COMMIT is enumerated by TDX_FEATURES0.TD_MIGRATION (bit 0), readable by TDH.SYS.RD*. If not supported, calling TDH.IMPORT.COMMIT returns a TDX_OPERAND_INVALID(RAX) status.

3.1.12.3.3. Preconditions

- An import session is in progress and has not been committed yet by TDH.IMPORT.COMMIT.
- A start token has been imported by TDH.IMPORT.TRACK.

3.1.12.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.50: TDH.IMPORT.COMMIT Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	R	Opaque	4KB	Exclusive	Shared	Shared
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A

3.1.12.5. Completion Status Codes

Table 3.51: TDH.IMPORT.COMMIT Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SUCCESS	Operation is successful
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

5

3.1.13. TDH.IMPORT.END Leaf

End an import session.

3.1.13.1. Input Operands

Table 3.52: TDH.IMPORT.END Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 81
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	HPA of the destination TD's TDR page (HKID bits must be 0)		

3.1.13.2. Output Operands

Table 3.53: TDH.IMPORT.END Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
Other	Unmodified

3.1.13.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.13.3.1. Overview

TDH.IMPORT.END ends an import session and allows the important TD to run (if not already allowed by TDH.IMPORT.COMMIT).

When called as part of an S4 resumption session, TDH.IMPORT.END must be called after no future replay is prevented by calling TDH.SYS.S4_END.

3.1.13.3.2. Enumeration

Availability of TDH.IMPORT.END is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.IMPORT.END returns a TDX_OPERAND_INVALID(RAX) status.

3.1.13.3.3. Preconditions

- An import session is in progress.
- A start token has been imported by TDH.IMPORT.TRACK.

3.1.13.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.54: TDH.IMPORT.END Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	R	Opaque	4KB	Exclusive	Shared	Shared
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A

5

3.1.13.5. Completion Status Codes

Table 3.55: TDH.IMPORT.END Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SUCCESS	Operation is successful
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.14. TDH.IMPORT.MEM Leaf

TDH.IMPORT.MEM imports a list of TD private pages contents and/or cancellation requests based on a migration bundle in shared memory.

3.1.14.1. Input Operands

5 **Table 3.56: TDH.IMPORT.MEM Input Operands Definition**

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 83
		23:16	Version Number	Selects the SEAMCALL interface function version Version may be 0 or 1. See enumeration details below.
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	GPA_LIST_INFO	HPA of a GPA list page in shared memory, and first and last entries to process, as defined in 2.2.2 On a new invocation, FIRST_ENTRY must be 0. On a resumed invocation, FIRST_ENTRY must be the index of the next GPA list entry to export.		
RDX	TD_HANDLE_AND_FLAGS	Destination TD handle and flags		
		Bits	**Name**	**Description**
		0	NO_REOWN	Indicates that no change of physical page ownership between the host VMM and the guest TD must happen See enumeration details below
		11:1	Reserved	Must be 0
		51:12	TDR HPA	HPA[51:12] of the destination TD's TDR page (HKID bits must be 0)
		63:52	Reserved	Must be 0
R8	MBMD	HPA and size of memory of an MBMD structure in memory:		
		Bits	**Name**	**Description**
		51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
		63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.

<table>
<tr><th>Operand</th><th>Name</th><th>Description</th></tr>
<tr><td>R9</td><td>MIG_BUFF_LIST</td><td>HPA (including HKID bits) of a migration buffer list in shared memory, corresponding to the GPA list pointed by RCX – see 2.2.3.
No migration buffers are required for PENDING pages and for migration cancellation requests. The list entries for such pages are skipped.</td></tr>
<tr><td rowspan="4">R10</td><td rowspan="4">MIG_STREAM</td><td>Migration stream and resume flag:</td></tr>
<tr><td>
<table>
<tr><th>Bits</th><th>Name</th><th>Description</th></tr>
<tr><td>15:0</td><td>MIGS_INDEX</td><td>Migration stream index
If N migration streams have been created by TDH.MIG.STREAM.CREATE, then MIGS_INDEX must be lower than N-1.</td></tr>
<tr><td>62:16</td><td>RESERVED</td><td>Reserved: must be 0</td></tr>
<tr><td>63</td><td>RESUME</td><td>0: This is a new invocation
1: This is resumption of a previously interrupted operation</td></tr>
</table>
</td></tr>
<tr><td>R11</td><td>MAC_LIST_0</td><td>HPA (including HKID bits) of a MAC list in shared memory, corresponding to the first 256 entries of the GPA list pointed by RCX – see 2.2.3.
If GPA_LIST_INFO.FIRST_ENTRY >= 256, then MAC_LIST_0 is ignored.</td></tr>
<tr><td>R12</td><td>MAC_LIST_1</td><td>HPA (including HKID bits) of a MAC list in shared memory, corresponding to the last 256 entries of the GPA list pointed by RCX – see 2.2.3.
If GPA_LIST_INFO.LAST_ENTRY < 256, then MAC_LIST_1 is ignored.</td></tr>
<tr><td>R13</td><td>PAGE_LIST</td><td>HPA (including HKID bits) of a destination page list in shared memory.
The page list is optional; if not provided, PAGE_LIST must be NULL_PA (all 1’s). The page list is not required if NO_REOWN is set to 1.
See the description below for details.</td></tr>
<tr><td>R14</td><td>ATTRIB_LIST</td><td>If GPA_LIST_INFO.FORMAT is GPA_AND_L2_ATTR, then R14 contains the HPA (including HKID bits) of a page L2 attributes list in shared memory – see 2.2.4.
Else, R14 is ignored.
An ATTRIB_LIST is mandatory for importing partitioned TDs (which contain one or more L2 VMs).
Enumeration: Support of this feature is enumerated by TDX_FEATURES0.PARTITIONED_TD_MIGRATION, readable by TDH.SYS.RD*.</td></tr>
</table>

3.1.14.2. Output Operands

Table 3.57: TDH.IMPORT.MEM Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RCX	GPA_LIST_INFO	Same as the input value, except that FIRST_ENTRY is updated to the index of the next entry to be processed. If all entries have been processed, FIRST_ENTRY is updated to (LAST_ENTRY + 1) Modulo 512.

Operand	Name	Description
R8	LIST_ERR_COUNT	If TDH.IMPORT.MEM was called with version 1 or higher, R8 returns the number of GPA list entries where an error was encountered. Else, R8 is unmodified.
AVX, AVX2 and AVX512 state		May be reset to the architectural RESET state
Other		Unmodified

3.1.14.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.14.3.1. Overview

TDH.IMPORT.MEM imports a list of up to 512 TD private 4KB pages based on a migration bundle, which includes an MBMD, set of 4KB pages encrypted with the migration session key, a 4KB page containing the GPA and attributes list, an optional 4KB containing page attributes list, and two 4KB pages containing page MACs.

For each page in the migration bundle's GPA list, the requested operation may either be to import the page, to re-import a newer version of the page (after a previous import) or to cancel a previous page import. It is also possible to skip entries in the list by requesting no operation for specific entries. The GPA list format is described in 2.2.2.

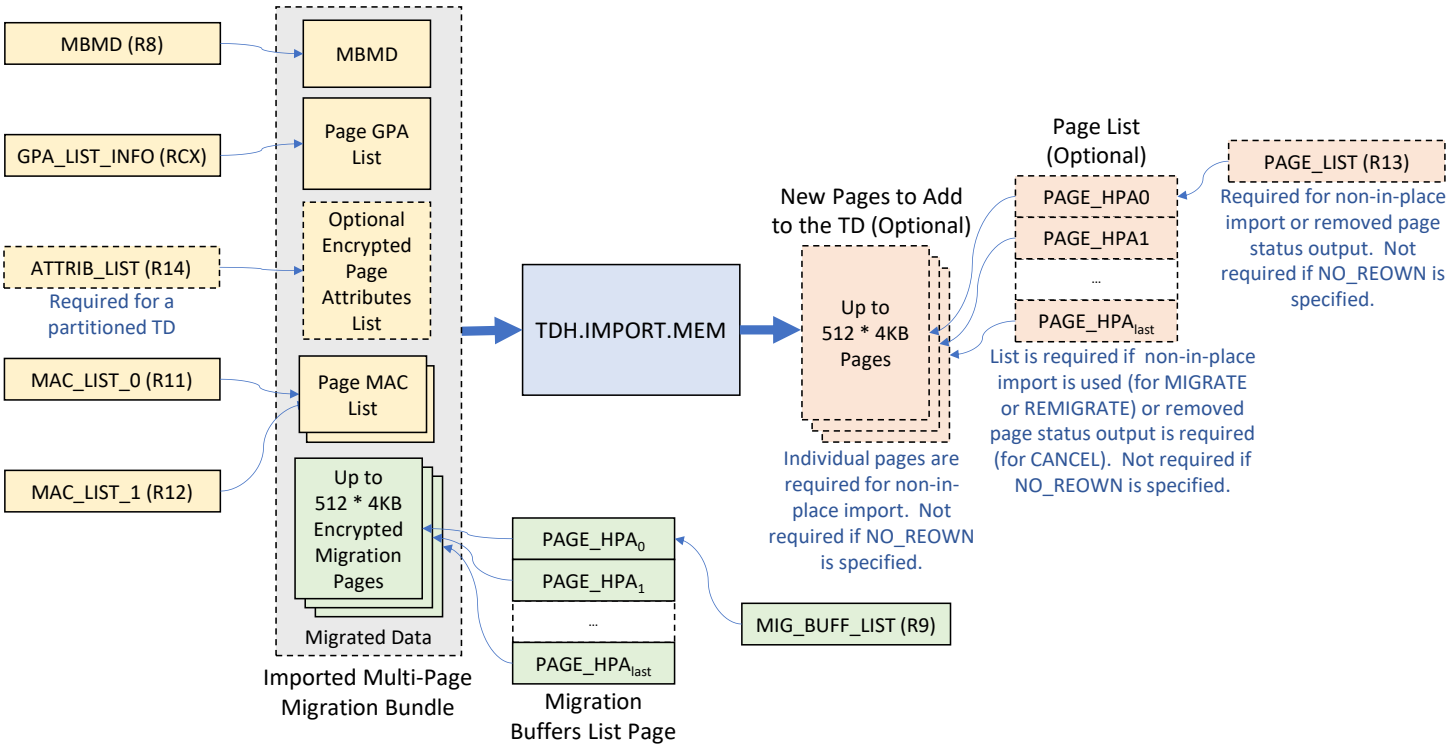

Figure 3.2: TDH.IMPORT.MEM Inputs and Outputs

Entries in the Page GPA List (pointed by RCX), Page Attributes List (pointed by R14), Page MAC List (pointed by R11 and R12), Migration Buffers List (pointed by R9) and Page List (pointed by R13) are sorted in the same order. I.e., entry N in each of those lists applies to the same imported page.

3.1.14.3.2. Enumeration

Availability of TDH.IMPORT.MEM is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.IMPORT.MEM returns a TDX_OPERAND_INVALID(RAX) status.

TDX_FEATURES0.PARTITIONED_TD_MIGRATION (bit 21) enumerates TDX Module support of migrating partitioned TDs (which contain one or more L2 VMs).

TDX_FEATURES0.IMPORT_PAGE_STATUS (bit 44) enumerates update of page list entries to indicate status (see below).

Support of TDH.EXPORT.MEM version 1 or higher is enumerated by TDX_FEATURES0.LIST_ERROR_COUNT (bit 54).

Support of the NO_REOWN input flag and partial GPA list processing is enumerated by TDX_FEATURES0.ENHANCED_IMPORT (bit 60).

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.14.3.3. Preconditions

- An import session is in progress.
- The specified migration stream has been created by TDH.MIG.STREAM.CREATE.

3.1.14.3.4. Page List

The page list is optional. If not provided, PAGE_LIST (R13) must be set to NULL_PA (all 1's). Specifically, the page list is not used if NO_REOWN is set to 1.

If a page list is provided, i.e., PAGE_LIST (R13) contains a valid shared HPA of a page containing page list entry, then each entry in the list is processed as follows:

A page list entry, defined in in the [ABI Spec] section titled "Private Page List", is used by TDH.IMPORT.MEM on input as follows:

- If the INVALID bit (63) is 0, the entry should contain a valid shared HPA of a page that can be converted to a TD private page (HKID bits must be 0). TDH.IMPORT.MEM uses this page for a MIGRATE operation.
- Else (the INVALID bit is 1), the entry doesn't point to a page. For a MIGRATE operation, or for a REMIGRATE operation where the previously imported page was locally removed by TDH.MEM.PAGE.REMOVE, TDH.IMPORT.MEM uses the applicable migration buffer for in-place import (see below). To facilitate parsing of the list entry on output (see below) it is recommended that the host VMM will set such entries to NULL_PA (all 1's).

A page list entry is updated by TDH.IMPORT.MEM on output as follows:

- If an entry was skipped due to a NOP operation or due to an error, the INVALID bit (63) is set to 1.

The following applies only if TDX_FEATURES0.IMPORT_PAGE_STATUS (bit 44) is enumerated as 1:

- If a page was provided on input (i.e., the INVALID bit was 0) but was not used by TDH.IMPORT.MEM as a TD private page, the INVALID bit is set to 1.
- If the import operation was CANCEL and it successfully removed a TD private page, and a page was not provided on input (the INVALID bit was 1), the page list entry is updated as follows:
  - Bits 51:0 contain the removed page HPA (HKID bits are 0).
  - The REMOVED bit (61) is set to 1 to indicate a removed page.
  - If the TDX Module is configured for Dynamic PAMT, the PAMT_REMOVAL_HINT bit (62) provides a hint that the PAMT pages mapping the removed page are empty and can be removed.
  - The INVALID bit (63) is cleared to 0.

  The above does not apply if the NO_REOWN input flag is set. In this case, a CANCEL operation does not remove the TD private page; instead, it converts it to a pre-allocated Pending page.

If a page list is not provided, i.e., R13 contains NULL_PA (all 1's), then MIGRATE operations are restricted as follows:

- Either an in-place import is used, or
- The page has been pre-allocated by TDH.MEM.PAGE.AUG.

Other import operations (NOP, REMIGRATE, CANCEL) never require a new TD private page.

3.1.14.3.5. Re-Import

Re-import is only allowed during the in-order import phase. The imported pages replace an older version of the same pages, if the SEPT entry state is compatible:

- If the old SEPT state is PENDING, it may be overwritten by a new version that is either PENDING or MAPPED.
- If the old SEPT state is MAPPED, then if the TDX module supports page release by the guest TD, as enumerated by TDX_FEATURES0.PAGE_RELEASE, it may be overwritten by a new version that is either MAPPED or PENDING. Else, it may only be overwritten by a new version that is MAPPED.

Page attributes (e.g., RWX etc.) of a new page version may be different than those of a previously imported version.

If the out-of-order import phase, the imported pages may not overwrite an older version of the same pages.

3.1.14.3.6. In-Place Import

In the following cases, import may be done in-place; the same physical pages that are provided as input are converted to TD private pages:

- First-time import (MIGRATE operation) of a page during the current import session.
- Import (MIGRATE operation) following a previous cancellation (CANCEL operation).
- Re-import (REMIGRATE operation) following a local removal (TDH.MEM.PAGE.REMOVE) of the page.

Alternatively, a list of 4KB pages to be used as the destination TD new private pages may be provided, as described above. In any case, either a migration buffer or a new page must be provided, even if the imported page is PENDING and no content is imported.

3.1.14.3.7. Import over an Existing Page

In the following cases, import is done over an existing page:

- First-time import (MIGRATE operation) of a page during the current import session, following page pre-allocation by TDH.MEM.PAGE.AUG.
- Import (MIGRATE operation) following a previous cancellation (CANCEL operation) where NO_REOWN was specified, i.e., the page was converted to a Pending pre-allocated page.
- Re-import (REMIGRATE operation) following a previous MIGRATE or a REMIGRATE of the same page.

In all cases, the new page, if provided, is not used.

3.1.14.3.8. Avoiding Change of Physical Page Ownership

If the NO_REOWN input flag is set, TDH.IMPORT.MEM does not allow the physical page ownership to change. This means the following:

- A MIGRATE operation will only succeed if the page is already owned by the guest TD, i.e., it has been pre-allocated by TDH.MEM.PAGE.AUG.
- A REMIGRATE operation will only succeed if the page is already owned by the guest TD, i.e., it was imported before and either has not been locally removed by TDH.MEM.PAGE.REMOVE, or was locally removed and then pre-allocated by TDH.MEM.PAGE.AUG.
- A CANCEL operation will not remove the page; instead, it will convert it to a pre-allocated PENDING page.

If the host VMM sets NO_REOWN, the page list (specified by R13) is not used.

3.1.14.3.9. GPA_LIST_INFO

On a new invocation, GPA_LIST_INFO.FIRST_ENTRY must be 0. On a resumed invocation, GPA_LIST_INFO.FIRST_ENTRY must be the index of the next GPA list entry to export.

If supported by the TDX Module, as enumerated by TDX_FEATURES0.ENHANCED_IMPORT (bit 60), the host VMM may specify that only part of the GPA list will be processed by TDH.IMPORT.MEM in the current invocation. This is done by specifying a GPA_LIST_INFO.LAST_ENTRY value that does not cover the whole GPA list (the number of GPAs in the list is provided in the imported MBMD header's NUM_GPAS field). In this case, TDH.IMPORT.MEM returns with a TDX_INTERRUPTED_LIST_END status; the host VMM can later resume the operation and process more entries in the GPA list. See 3.1.14.3.13 below.

3.1.14.3.10. GPA List Entry Error Handling

If a page can't be imported for a reason that is specific to that page, TDH.IMPORT.MEM marks its GPA list entry as unsuccessful.

When a page is not imported, its GPA list entry is updated as follows:

- The OPERATION field is written as NOP, indicating that the page has not been imported.
- The STATUS field is written with the status code, as shown below.

In some cases, listed in the table below, the TDX Module aborts the import session because the memory state of the imported TD can't be guaranteed to be correct. The target TD is marked as IMPORT_FAILED and, by design, will not run. This is indicated by the FATAL bit (61) of the completion status returned in RAX.

In other cases, listed in the table below, the TDX Module aborts the import session only if it has not been committed yet (by THD.IMPORT.COMMIT). If the import session has been committed, GPA list processing continues to the next entry, if applicable.

If TDH.IMPORT.MEM was called with version 1 or higher, R8 returns the number of GPA list entries where an error was encountered since it was last invoked, either as a new invocation or as a resumption of a previously interrupted operation. Else, R8 is unmodified.

The following table shows the possible GPA list entry STATUS values written by TDH.IMPORT.MEM. Refer to 2.2.2.4.7 for STATUS definition.

Table 3.58: GPA List Entry STATUS Values Returned by TDH.IMPORT.MEM

Value	Name	Description
0	SUCCESS	The GPA list entry was processed successfully.
1	SKIPPED	The GPA list entry was skipped because NOP was requested; this is not an error.
2	SEPT_WALK_FAILED	If the import session has been committed (TDH.IMPORT.COMMIT), the GPA list entry was skipped. Else, the import session is aborted.
3	SEPT_ENTRY_BUSY_HOST_PRIORITY	If the import session has been committed (TDH.IMPORT.COMMIT), the GPA list entry was skipped. Else, the import session is aborted.
4	SEPT_ENTRY_STATE_INCORRECT	If the import session has been committed (TDH.IMPORT.COMMIT), the GPA list entry was skipped. Else, the import session is aborted.
6	OP_STATE_INCORRECT	Import session is aborted.
7	MIGRATED_IN_CURRENT_EPOCH	Import session is aborted.
8	MIG_BUFFER_NOT_AVAILABLE	Import session is aborted.
9	NEW_PAGE_NOT_AVAILABLE	If the provided HPA is valid, then if the import session has been committed (TDH.IMPORT.COMMIT), the GPA list entry was skipped. Else, the import session is aborted. Else (HPA is invalid) the import session is aborted unconditionally.
10	INVALID_PAGE_MAC	Import session is aborted.
11	DISALLOWED_IMPORT_OVER_REMOVED	Import session is aborted.
13	L2_SEPT_WALK_FAILED	If the import session has been committed (TDH.IMPORT.COMMIT), the GPA list entry was skipped. Else, the import session is aborted.
14	ATTR_LIST_ENTRY_INVALID	Import session is aborted.
15	GPA_LIST_ENTRY_INVALID	Import session is aborted.
18	REOWN_DISALLOWED	Import session is aborted.

3.1.14.3.11. S4 Resumption

If TDH.IMPORT.MEM is called as part of an S4 hibernation, it only supports the out-of-order import phase. As a result, the GPA list may not contain CANCEL and REMIGRATE operations.

In case of an import error, then in addition to aborting the import of the current TD, an abort of an S4 resumption session also generates a new S4 anti-replay nonce, preventing a new S4 resumption session from starting.

3.1.14.3.12. Dynamic PAMT

If the TDX Module is configured for dynamic PAMT, the PAMT hierarchy can be built on demand. If a missing PAMT page pair is detected during operation, TDH.IMPORT.MEM returns with a TDX_INTERRUPTED_PAMT status in RAX. RCX is updated with the list entry for which the error happened. The host VMM is expected to add the missing PAMT page pair using TDH.PHYMEM.PAMT.ADD, then re-invoke TDH.IMPORT.MEM, keeping the same inputs as the original invocation except using the updated RCX value and setting R10.RESUME to 1.

3.1.14.3.13. Interruption and Resumption

TDH.IMPORT.MEM is interruptible. An interruption occurs in the following cases:

- If a pending interrupt is detected during operation, TDH.IMPORT.MEM returns with a TDX_INTERRUPTED_RESUMABLE status in RAX.
- If the TDX Module is configured for dynamic PAMT, and a missing PAMT page pair is detected during operation, TDH.IMPORT.MEM returns with a TDX_INTERRUPTED_PAMT status in RAX.
- If supported by the TDX Module, and the specified GPA_LIST_INFO.LAST_ENTRY is lower than imported MBMD's NUM_GPAS minus 1, TDH.IMPORT.MEM returns with a TDX_INTERRUPTED_LIST_END status in RAX. See 3.1.14.3.9 above.

RCX is updated with the next list entry index to process.

The host VMM should re-invoke TDH.IMPORT.MEM after handling the interruption reason, keeping the same inputs (and updated value in RCX) except setting R10.RESUME to 1 and optionally updating GPA_LIST_INFO.LAST_ENTRY. If the host VMM cannot resume TDH.IMPORT.MEM for some reason, it should abort the import session (TDH.IMPORT.ABORT).

3.1.14.3.14. Interrupt Latency

TDH.IMPORT.MEM may exceed the normal interrupt response latency limit if the core frequency is minimal. However, since TD Migration is a CPU-intensive operation, this is condition is not expected to happen in real-life scenarios. See the [Base Spec] section titled "Latency of the Intel TDX Interface Functions".

3.1.14.3.15. Removed Page Initialization

On platforms which do not use ACT, after any private pages have been removed by a CANCEL operation, the host VMM should initialize their content before they are reused as non-private pages, as described in the [Base Spec].

3.1.14.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.59: TDH.IMPORT.MEM Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions			
								Operand	Contain. 2MB	Contain. 1GB	ACT 2MB[4]
Explicit	RCX	HPA	GPA List page	GPA_LIST	RW	Shared	4KB	None	None	None	None
Explicit	RDX	HPA	TDR page	TDR	RW	Opaque	4KB	Shared	Shared	Shared	None
Explicit	R8	HPA	MBMD	MBMD	R	Shared	128B	None	None	None	None

[4] ACT is enumerated by TDX_FEATURES0.ACT, readable using TDH.SYS.RD.

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions			
								Operand	Contain. 2MB	Contain. 1GB	ACT 2MB[4]
Explicit	R9	HPA	Migration buffer list	PAGE_LIST	R	Shared	4KB	None	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A	N/A
Explicit	R11	HPA	MAC list page 1	MAC list	R	Shared	4KB	None	None	None	None
Explicit	R12	HPA	MAC list page 2	MAC list	R	Shared	4KB	None	None	None	None
Explicit	R13	HPA	Destination page list	Blob	RW	Private	4KB	Exclusive	Shared	Shared	None
Explicit	R14	HPA	L2 attributes list page	L2 page attributes	R	Shared	4KB	None	None	None	None
Explicit	N/A	GPA	TD private pages (via GPA list)	Blob	None	Private	4KB	None	None	None	None
Explicit	N/A	HPA	Migration buffer pages (via page list)	Blob	RW	Shared	4KB	None	None	None	None
Explicit	N/A	HPA	Destination pages (via page list)	Blob	RW	Private	4KB	Exclusive	Shared	Shared	Exclusive
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Shared	N/A	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Shared	N/A	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT tree	N/A	RW	Private	N/A	Shared	N/A	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT entries	SEPT Entry	RW	Private	N/A	Exclusive	N/A	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT trees	N/A	RW	Private	N/A	Shared(i)	N/A	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT entries	SEPT Entry	RW	Private	N/A	Exclusive(i)	N/A	N/A	N/A

3.1.14.5. Completion Status Codes

The table below details the status values returned in RAX. The FATAL bit (61) indicates that the import session has been aborted. In such cases, the status code name is appended with a "_FATAL" suffix.

Table 3.60: TDH.IMPORT.MEM Completion Status Codes (Returned in RAX) Definition

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_EPT_ENTRY_STATE_INCORRECT	0	Applicable to cases where we allow skipping of import if the state was changed locally during out-of-order import. DETAILS_L2: OPERAND_ID: GPA_LIST_ENTRY GPA List Entry STATUS: SEPT_ENTRY_STATE_INCORRECT
TDX_EPT_ENTRY_STATE_INCORRECT_FATAL	1	DETAILS_L2: OPERAND_ID: GPA_LIST_ENTRY GPA List Entry STATUS: SEPT_ENTRY_STATE_INCORRECT, TDX_IMPORT_MISMATCH
TDX_EPT_WALK_FAILED	0	Applicable to the out-of-order import phase DETAILS_L2: OPERAND_ID: GPA_LIST_ENTRY GPA List Entry STATUS: SEPT_WALK_FAILED
TDX_EPT_WALK_FAILED_FATAL	1	Applicable to the in-order import phase DETAILS_L2: OPERAND_ID: GPA_LIST_ENTRY GPA List Entry STATUS: SEPT_WALK_FAILED
TDX_INCORRECT_MBMD_MAC	0	
TDX_INTERRUPTED_LIST_END	0	
TDX_INTERRUPTED_PAMT	0	DETAILS_L2: GPA list entry index
TDX_INTERRUPTED_RESUMABLE	0	
TDX_INVALID_MBMD	0	
TDX_INVALID_PAGE_MAC_FATAL	1	DETAILS_L2: OPERAND_ID: GPA_LIST_ENTRY GPA List Entry STATUS: INVALID_PAGE_MAC
TDX_INVALID_RESUMPTION	0	
TDX_L2_SEPT_WALK_FAILED	0	Applicable to the out-of-order import phase DETAILS_L2: VM and Level GPA List Entry STATUS: L2_SEPT_WALK_FAILED
TDX_L2_SEPT_WALK_FAILED_FATAL	1	Applicable to the in-order import phase DETAILS_L2: VM and Level GPA List Entry STATUS: L2_SEPT_WALK_FAILED
TDX_MIGRATED_IN_CURRENT_EPOCH_FATAL	1	DETAILS_L2: GPA_LIST_ENTRY GPA List Entry STATUS: MIGRATED_IN_CURRENT_EPOCH
TDX_MIGRATION_STREAM_STATE_INCORRECT	0	
TDX_OP_STATE_INCORRECT	0	DETAILS_L2: OP_STATE
TDX_OP_STATE_INCORRECT_FATAL	1	Applicable to OP_STATE restrictions for specific import operations, e.g., CANCEL is not allowed during the out-of-order phase. DETAILS_L2: OP_STATE
TDX_OPERAND_ADDR_RANGE_ERROR	0	DETAILS_L2: OPERAND_ID: RDX
TDX_OPERAND_ADDR_RANGE_ERROR_FATAL	1	DETAILS_L2: OPERAND_ID: NEW_PAGE_LIST_ENTRY GPA List Entry STATUS: NEW_PAGE_NOT_AVAILABLE
		DETAILS_L2: RDX, MIG, MIGSC, SEPT_TREE

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_OPERAND_BUSY	0	Applicable to the out-of-order import phase DETAILS_L2: OPERAND_ID: GPA_LIST_ENTRY GPA List Entry STATUS: SEPT_ENTRY_BUSY_HOST_PRIORITY, TD_PAGE_BUSY_HOST_PRIORITY
		DETAILS_L2: OPERAND_ID: NEW_PAGE_LIST_ENTRY GPA List Entry STATUS: NEW_PAGE_NOT_AVAILABLE
TDX_OPERAND_BUSY_FATAL	1	Applicable to the in-order import phase DETAILS_L2: OPERAND_ID: GPA_LIST_ENTRY GPA List Entry STATUS: SEPT_ENTRY_BUSY_HOST_PRIORITY, TD_PAGE_BUSY_HOST_PRIORITY
		Applicable to the in-order import phase DETAILS_L2: OPERAND_ID: NEW_PAGE_LIST_ENTRY GPA List Entry STATUS: NEW_PAGE_NOT_AVAILABLE
TDX_OPERAND_INVALID	0	DETAILS_L2: OPERAND_ID: RCX, RDX, R8 – R14
		Applicable to the out-of-order import phase DETAILS_L2: OPERAND_ID: NEW_PAGE_LIST_ENTRY GPA List Entry STATUS: NEW_PAGE_NOT_AVAILABLE
TDX_OPERAND_INVALID_FATAL	1	DETAILS_L2: OPERAND_ID: GPA_LIST_ENTRY GPA List Entry STATUS: GPA_LIST_ENTRY_INVALID, ATTR_LIST_ENTRY_INVALID
		DETAILS_L2: OPERAND_ID: MIG_BUFF_LIST_ENTRY GPA List Entry STATUS: MIG_BUFFER_NOT_AVAILABLE
		Applicable to the in-order import phase DETAILS_L2: OPERAND_ID: NEW_PAGE_LIST_ENTRY GPA List Entry STATUS: NEW_PAGE_NOT_AVAILABLE
TDX_PAGE_METADATA_INCORRECT	0	DETAILS_L2: RDX
		Applicable to the out-of-order import phase DETAILS_L2: NEW_PAGE_LIST_ENTRY GPA List Entry STATUS: NEW_PAGE_NOT_AVAILABLE
TDX_PAGE_METADATA_INCORRECT_FATAL	1	Applicable to the in-order import phase DETAILS_L2: NEW_PAGE_LIST_ENTRY GPA List Entry STATUS: NEW_PAGE_NOT_AVAILABLE
TDX_REOWN_DISALLOWED_FATAL	1	DETAILS_L2: GPA list entry index GPA List Entry STATUS: DISALLOWED_IMPORT_OVER_REMOVED
TDX_SUCCESS	0	The operation is successful. Processing of some GPA list entries may have encountered errors, but this did not cause an abortion of the overall operation. The number of such errors is reported in R8.
TDX_SYS_NOT_READY	0	

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_SYS_SHUTDOWN	0	
TDX_TD_FATAL	1	DETAILS_L2: OPERAND_ID: RDX
TDX_TD_KEYS_NOT_CONFIGURED	0	DETAILS_L2: OPERAND_ID: RDX
TDX_TDCS_NOT_ALLOCATED	0	DETAILS_L2: OPERAND_ID: RDX

3.1.15. TDH.IMPORT.STATE.IMMUTABLE Leaf

TDH.IMPORT.STATE.IMMUTABLE starts a new import session and exports the TD's immutable state as a multi-page migration bundle.

TDH.IMPORT.STATE.IMMUTABLE is also used for starting a new S4 resumption session.

3.1.15.1. Input Operands

Table 3.61: TDH.IMPORT.STATE.IMMUTABLE Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 85
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	TD Handle	Destination TD handle and flags		
		Bits	**Name**	**Description**
		0	IMPORT_TYPE	0: TD Import 1: S4 Resumption See enumeration details below.
		11:1	Reserved	Must be 0
		51:12	TDR HPA	HPA[51:12] of the destination TD's TDR page (HKID bits must be 0)
		63:52	Reserved	Must be 0
R8	MBMD	HPA and size of memory of an MBMD structure in memory:		
		Bits	**Name**	**Description**
		51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
		63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.

Operand	Name	Description		
R9	PAGE_LIST_INFO	Migration buffers list information, in HPA_LIST_INFO format (see the [ABI Spec]) FORMAT (bits 2:0) and FIRST_ENTRY (bits 11:3) must be 0.		
R10	MIG_STREAM	Migration stream and resume flag:		
		Bits	**Name**	**Description**
		15:0	MIGS_INDEX	Migration stream index – must be 0
		62:16	RESERVED	Reserved: must be 0
		63	RESUME	0: This is a new invocation 1: This is resumption of a previously interrupted operation

3.1.15.2. Output Operands

Table 3.62: TDH.IMPORT.STATE.IMMUTABLE Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RCX	EXTENDED_ ERROR_INFO1	In case of an interruption, as indicated by RAX returning TDX_INTERRUPTED_RESUMABLE, RCX is unmodified. In case of an error related to non-memory state field import, as indicated by RAX, RCX contains the offending field identifier. See the status codes table below. In other cases, RCX returns 0.
RDX	EXTENDED_ ERROR_INFO2	In case of an interruption, as indicated by RAX returning TDX_INTERRUPTED_RESUMABLE, RDX is unmodified. In other cases, RDX returns 0.
AVX, AVX2 and AVX512 state		May be reset to the architectural RESET state
Other		Unmodified

3.1.15.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.15.3.1. Overview

TDH.IMPORT.STATE.IMMUTABLE starts a new import session. It imports the TD's immutable state migration bundle previously exported by TDH.EXPORT.STATE.IMMUTABLE. The migration bundle includes an MBMD and a set of 4KB pages.

TDH.IMPORT.STATE.IMMUTABLE is also used for starting a new S4 resumption session.

TD immutable state is verified by TDH.IMPORT.STATE.IMMUTABLE against target platform capabilities and Intel TDX Module version, capabilities and configuration. The checks are similar, but not identical, to the TD_PARAMS checks done on the source platform by TDH.MNG.INIT.

3.1.15.3.2. Enumeration

Availability of TDH.IMPORT.STATE.IMMUTABLE is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.IMPORT.STATE.IMMUTABLE returns a TDX_OPERAND_INVALID(RAX) status.

IMPORT_TYPE value 0 (TD Import) is supported if TDX_FEATURES0.TD_MIGRATION (bit 0) is 1.

IMPORT_TYPE value 1 (S4 Resumption) is supported if TDX_FEATURES0.S4 (bit 13) is 1.

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.15.3.3. Preconditions

- The TD has been created but not initialized by TDH.MNG.INIT.
- There is no migration session in progress.
- A Migration TD has been bound to the source TD.
- The migration TD has written the migration decryption key and migration protocol version metadata fields.
- Migration stream 0 has been created by TDH.MIG.STREAM.CREATE.

3.1.15.3.4. Interruptibility

If a pending interrupt is detected during operation, TDH.IMPORT.STATE.IMMUTABLE returns with a TDX_INTERRUPTED_RESUMABLE status in RAX. The host VMM is expected to resume it, by calling it with R10.RESUME set to 1. If the host VMM cannot resume TDH.IMPORT.STATE.IMMUTABLE for some reason, it should abort the import session (TDH.IMPORT.ABORT).

3.1.15.3.5. Aspects of MAC Calculation Compatibility on TDX Module Update

If TDH.IMPORT.STATE.IMMUTABLE was interrupted, and later the TDX Module was updated before TDH.IMPORT.STATE.IMMUTABLE was resumed, there could be incompatibility issues with the format of the intermediate MRTD MAC context, held as part of the migration stream context (MIGSC), between the original and the new TD module.

On TDH.SYS.UPDATE, the host VMM can configure the TDX Module to detect such incompatibility. If detected, TDH.IMPORT.STATE.IMMUTABLE returns a TDX_INCOMPATIBLE_ MBMD_MAC_CONTEXT status. In this case, the host VMM is expected to abort the migration session and may restart it. For details, see the [Base Spec] section titled "Compatibility Aspects of TD-Preserving Update".

If the host VMM did not configure the TDX Module to detect such incompatibility, it will not be detected by TDH.IMPORT.STATE.IMMUTABLE. However, TDH.IMPORT.STATE.IMMUTABLE will detect that the MAC is incorrect and will return a TDX_INCORRECT_MBMD_MAC error.

3.1.15.3.6. Import Abort

A failed TDH.IMPORT.STATE.IMMUTABLE marks (except in cases where the imported TD state has not been modified) the target TD as IMPORT_FAILED; by design, it will not run. This is indicated by the FATAL bit (61) of the completion status returned in RAX.

3.1.15.3.7. S4 Resumption Abort

In addition to aborting the import of the current TD, an abort of an S4 resumption session also generates a new S4 anti-replay nonce, preventing a new S4 resumption session from starting.

3.1.15.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.63: TDH.IMPORT.STATE.IMMUTABLE Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	RW	Opaque	4KB	Exclusive	Shared	Shared

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	R8	HPA	MBMD	MBMD	R	Shared	128B	None	None	None
Explicit	R9	HPA	Page list	PAGE_LIST	R	Shared	4KB	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Explicit	N/A	HPA	Source pages (via page list)	Blob	R	Shared	4KB	None	None	None
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	PL.S4_STATE	N/A	RW	Hidden	N/A	Exclusive	N/A	N/A

3.1.15.5. Completion Status Codes

The table below details the status values returned in RAX. The FATAL bit (61) indicates that the import session has been aborted. In such cases, the status code name is appended with a "_FATAL" suffix.

Table 3.64: TDH.IMPORT.STATE.IMMUTABLE Completion Status Codes (Returned in RAX) Definition

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_CPUID_LEAF_1F_FORMAT_UNRECOGNIZED_FATAL	1	
TDX_INCOMPATIBLE_MBMD_MAC_CONTEXT_FATAL	1	
TDX_INCORRECT_MBMD_MAC_FATAL	1	
TDX_INTERRUPTED_RESUMABLE	0	
TDX_INVALID_MBMD_FATAL	1	
TDX_INVALID_METADATA_LIST_HEADER_FATAL	1	
TDX_INVALID_MIGRATION_DECRYPTION_KEY	0	
TDX_INVALID_RESUMPTION	0	Applicable if the RESUME flag was set while no resumption is expected or clear while resumption is expected
TDX_INVALID_RESUMPTION_FATAL	1	Applicable if the interrupted interface function is not TDH.IMPORT.STATE.IMMUTABLE or PAGE_LIST_INFO is different than it was on interruption
TDX_METADATA_FIELD_ID_INCORRECT_FATAL	1	Field ID or sequence header is returned in RCX
TDX_METADATA_FIELD_NOT_WRITABLE_FATAL	1	Field ID is returned in RCX
TDX_METADATA_FIELD_VALUE_NOT_VALID_FATAL	1	Field ID is returned in RCX
TDX_METADATA_LIST_OVERFLOW_FATAL	1	

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_MIGRATION_DECRYPTION_KEY_NOT_SET	0	
TDX_MIGRATION_STREAM_STATE_INCORRECT	0	
TDX_MIN_MIGS_NOT_CREATED	0	
TDX_NUM_MIGS_HIGHER_THAN_CREATED_FATAL	1	
TDX_OP_STATE_INCORRECT	0	DETAILS_L2: OP_STATE
TDX_OPERAND_ADDR_RANGE_ERROR	0	DETAILS_L2: OPERAND_ID: RCX
TDX_OPERAND_BUSY	0	DETAILS_L2: OPERAND_ID: RCX, MIGSC
TDX_OPERAND_INVALID	0	DETAILS_L2: OPERAND_ID: RCX, R10, PAGE (if list buffer size is invalid)
TDX_OPERAND_INVALID_FATAL	1	DETAILS_L2: OPERAND_ID: R8, R9, PAGE (if HPA is invalid), CPUID_FIXED0_BITMAP
TDX_PAGE_METADATA_INCORRECT	0	DETAILS_L2: OPERAND_ID: RCX
TDX_REQUIRED_METADATA_FIELD_MISSING_FATAL	1	Required field ID is returned in RCX
TDX_RND_NO_ENTROPY	0	Failed to generate a random migration encryption key. This is typically caused by an entropy error of the CPU's random number generator, and may be impacted by RDSEED, RDRAND or PCONFIG executing on other LPs. The operation should be retried.
TDX_SERVTD_EXT_MISMATCH_FATAL	1	Rebind SERVTD_EXT hash did not match
TDX_SUCCESS	0	Operation is successful
TDX_S4_STATE_INCORRECT	0	Applicable only if TDX_FEATURES0.S4 (bit 13) is 1 and IMPORT_TYPE is 1 (S4 Resumption).
TDX_SYS_BUSY	0	Applicable only if TDX_FEATURES0.S4 (bit 13) is 1 and IMPORT_TYPE is 1 (S4 Resumption).
TDX_SYS_NOT_READY	0	
TDX_SYS_SHUTDOWN	0	
TDX_TD_FATAL	1	DETAILS_L2: OPERAND_ID: RCX
TDX_TD_KEYS_NOT_CONFIGURED	0	DETAILS_L2: OPERAND_ID: RCX
TDX_TDCS_NOT_ALLOCATED	0	DETAILS_L2: OPERAND_ID: RCX
TDX_VIRTUAL_MSR_VALUE_NOT_VALID_FATAL	1	DETAILS_L2: MSR index

3.1.16. TDH.IMPORT.STATE.TD Leaf

TDH.IMPORT.STATE.TD imports the TD-scope mutable state as a multi-page migration bundle.

3.1.16.1. Input Operands

Table 3.65: TDH.IMPORT.STATE.TD Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 86
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	TDR	HPA of the destination TD TDR page (HKID bits must be 0)		
R8	MBMD	HPA and size of memory of an MBMD structure in memory:		
		Bits	**Name**	**Description**
		51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
		63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.
R9	PAGE_LIST_INFO	Migration buffers list information, in HPA_LIST_INFO format (see the [ABI Spec]) FORMAT (bits 2:0) and FIRST_ENTRY (bits 11:3) must be 0.		
R10	MIG_STREAM	Migration stream and resume flag:		
		Bits	**Name**	**Description**
		15:0	MIGS_INDEX	Migration stream index – must be 0
		62:16	RESERVED	Reserved: must be 0
		63	RESUME	0: This is a new invocation 1: This is resumption of a previously interrupted operation

5

3.1.16.2. Output Operands

Table 3.66: TDH.IMPORT.STATE.TD Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RCX	EXTENDED_ ERROR_INFO1	In case of an interruption, as indicated by RAX returning TDX_INTERRUPTED_RESUMABLE, RCX is unmodified. In case of an error related to non-memory state field import, as indicated by RAX, RCX contains the offending field identifier. See the status codes table below. In other cases, RCX returns 0.
RDX	EXTENDED_ ERROR_INFO2	In case of an interruption, as indicated by RAX returning TDX_INTERRUPTED_RESUMABLE, RDX is unmodified. In other cases, RDX returns 0.
AVX, AVX2 and AVX512 state		May be reset to the architectural RESET state
Other		Unmodified

3.1.16.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.16.3.1. Overview

TDH.IMPORT.STATE.TD imports the TD-scope mutable state migration bundle previously exported by TDH.EXPORT.STATE.TD. The migration bundle includes an MBMD and a set of 4KB pages.

TD-scope mutable state is verified by TDH.IMPORT.STATE.TD against target platform capabilities and Intel TDX Module version, capabilities and configuration.

3.1.16.3.2. Enumeration

Availability of TDH.IMPORT.STATE.TD is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.IMPORT.STATE.TD returns a TDX_OPERAND_INVALID(RAX) status.

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.16.3.3. Preconditions

An import session is in progress, but TD-scope mutable state has not been imported yet by TDH.IMPORT.STATE.TD.

3.1.16.3.4. Interruptibility

If a pending interrupt is detected during operation, TDH.IMPORT.STATE.TD returns with a TDX_INTERRUPTED_RESUMABLE status in RAX. The host VMM is expected to resume it, by calling it with R10.RESUME set to 1. If the host VMM cannot resume TDH.IMPORT.STATE.TD for some reason, it should abort the import session (TDH.IMPORT.ABORT).

3.1.16.3.5. Import Abort

A failed TDH.IMPORT.STATE.TD marks (except in cases where the imported TD state has not been modified) the target TD as IMPORT_FAILED; by design, it will not run. This is indicated by the FATAL bit (61) of the completion status returned in RAX.

3.1.16.3.6. Aspects of MAC Calculation Compatibility on TDX Module Update

If TDH.IMPORT.STATE.TD was interrupted, and later the TDX Module was updated before TDH.IMPORT.STATE.TD was resumed, there could be incompatibility issues with the format of the intermediate MRTD MAC context, held as part of the migration stream context (MIGSC), between the original and the new TD module.

On TDH.SYS.UPDATE, the host VMM can configure the TDX Module to detect such incompatibility. If detected, TDH.IMPORT.STATE.TD returns a TDX_INCOMPATIBLE_ MBMD_MAC_CONTEXT status. In this case, the host VMM is expected to abort the migration session and may restart it. For details, see the [Base Spec] section titled "Compatibility Aspects of TD-Preserving Update".

If the host VMM did not configure the TDX Module to detect such incompatibility, it will not be detected by TDH.IMPORT.STATE.TD. However, TDH.IMPORT.STATE.TD will detect that the MAC is incorrect and will return a TDX_INCORRECT_MBMD_MAC error.

3.1.16.3.7. S4 Resumption Abort

In addition to aborting the import of the current TD, an abort of an S4 resumption session also generates a new S4 anti-replay nonce, preventing a new S4 resumption session from starting.

3.1.16.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.67: TDH.IMPORT.STATE.VP Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	R	Opaque	4KB	Exclusive	Shared	Shared
Explicit	R8	HPA	MBMD	MBMD	R	Shared	128B	None	None	None
Explicit	R9	HPA	Page list	PAGE_LIST	R	Shared	4KB	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Explicit	N/A	HPA	Source pages (via page list)	Blob	R	Shared	4KB	None	None	None
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	Service TD bindings table	N/A	R	Hidden	N/A	Exclusive(i)	N/A	N/A

3.1.16.5. Completion Status Codes

The table below details the status values returned in RAX. The FATAL bit (61) indicates that the import session has been aborted. In such cases, the status code name is appended with a "_FATAL" suffix.

Table 3.68: TDH.IMPORT.STATE.TD Completion Status Codes (Returned in RAX) Definition

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_INCOMPATIBLE_MBMD_MAC_CONTEXT_FATAL	1	
TDX_INCORRECT_MBMD_MAC_FATAL	1	
TDX_INTERRUPTED_RESUMABLE	0	
TDX_INVALID_MBMD_FATAL	1	
TDX_INVALID_METADATA_LIST_HEADER_FATAL	1	
TDX_INVALID_RESUMPTION	0	Applicable if the RESUME flag was set while no resumption is expected or clear while resumption is expected
TDX_INVALID_RESUMPTION_FATAL	1	Applicable if the interrupted interface function is not TDH.IMPORT.STATE.TD or PAGE_LIST_INFO is different than it was on interruption
TDX_METADATA_FIELD_ID_INCORRECT_FATAL	1	Field ID or sequence header is returned in RCX
TDX_METADATA_FIELD_NOT_WRITABLE_FATAL	1	Field ID is returned in RCX
TDX_METADATA_FIELD_VALUE_NOT_VALID_FATAL	1	Field ID is returned in RCX
TDX_METADATA_LIST_OVERFLOW_FATAL	1	
TDX_MIGRATION_STREAM_STATE_INCORRECT	0	
TDX_OP_STATE_INCORRECT	0	DETAILS_L2: OP_STATE
TDX_OPERAND_ADDR_RANGE_ERROR	0	DETAILS_L2: OPERAND_ID: RCX
TDX_OPERAND_BUSY	0	DETAILS_L2: OPERAND_ID: RCX, MIGSC
TDX_OPERAND_INVALID	0	DETAILS_L2: OPERAND_ID: RCX, R8, R10
TDX_OPERAND_INVALID_FATAL	1	DETAILS_L2: OPERAND_ID: R9, PAGE
TDX_PAGE_METADATA_INCORRECT	0	DETAILS_L2: OPERAND_ID: RCX
TDX_REQUIRED_METADATA_FIELD_MISSING_FATAL	1	Required field ID is returned in RCX
TDX_SUCCESS	0	
TDX_TD_FATAL	1	DETAILS_L2: OPERAND_ID: RCX
TDX_TD_KEYS_NOT_CONFIGURED	0	DETAILS_L2: OPERAND_ID: RCX
TDX_TDCS_NOT_ALLOCATED	0	DETAILS_L2: OPERAND_ID: RCX

3.1.17. TDH.IMPORT.STATE.VP Leaf

TDH.IMPORT.STATE.VP imports the VCPU-scope mutable state as a multi-page migration bundle.

3.1.17.1. Input Operands

Table 3.69: TDH.IMPORT.STATE.VP Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 87
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	TDVPR	HPA of the destination TD VCPU's TDVPR page (HKID bits must be 0)		
R8	MBMD	HPA and size of memory of an MBMD structure in memory:		
		Bits	**Name**	**Description**
		51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
		63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.
R9	PAGE_LIST_INFO	Migration buffers list information, in HPA_LIST_INFO format (see the [ABI Spec]) FORMAT (bits 2:0) and FIRST_ENTRY (bits 11:3) must be 0.		
R10	MIG_STREAM	Migration stream and resume flag:		
		Bits	**Name**	**Description**
		15:0	MIGS_INDEX	Migration stream index If N migration streams have been created by TDH.MIG.STREAM.CREATE, then MIGS_INDEX must be lower than N-1.
		62:16	RESERVED	Reserved: must be 0
		63	RESUME	0: This is a new invocation

Operand	Name	Description		
				1: This is resumption of a previously interrupted operation

3.1.17.2. Output Operands

Table 3.70: TDH.IMPORT.STATE.VP Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RCX	EXTENDED_ ERROR_INFO1	In case of an interruption, as indicated by RAX returning TDX_INTERRUPTED_RESUMABLE, RCX is unmodified. In case of an error related to non-memory state field import, as indicated by RAX, RCX contains the offending field identifier. See the status codes table below. In other cases, RCX returns 0.
RDX	EXTENDED_ ERROR_INFO2	In case of an interruption, as indicated by RAX returning TDX_INTERRUPTED_RESUMABLE, RDX is unmodified. In other cases, RDX returns 0.
AVX, AVX2 and AVX512 state		May be reset to the architectural RESET state
Other		Unmodified

3.1.17.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.17.3.1. Overview

TDH.IMPORT.STATE.VP imports the VCPU-scope mutable state migration bundle previously exported by TDH.EXPORT.STATE.VP. The migration bundle includes an MBMD and a set of 4KB pages.

VCPU-scope mutable state is verified by TDH.IMPORT.STATE.VP against target platform capabilities and Intel TDX Module version, capabilities and configuration.

3.1.17.3.2. Enumeration

Availability of TDH.IMPORT.STATE.VP is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.IMPORT.STATE.VP returns a TDX_OPERAND_INVALID(RAX) status.

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.17.3.3. Preconditions

- An import session is in progress.
- TD-scope mutable state has been imported by TDH.IMPORT.STATE.TD.
- The requested migration stream has been created by TDH.MIG.STREAM.CREATE.
- The VCPU has been created by TDGH.VP.CREATE and the required number of pages has been added by TDH.VP.ADDCX.

3.1.17.3.4. Interruptibility

If a pending interrupt is detected during operation, TDH.IMPORT.STATE.VP returns with a TDX_INTERRUPTED_RESUMABLE status in RAX. The host VMM is expected to resume it, by calling it with R10.RESUME set to 1. If the host VMM cannot resume TDH.IMPORT.STATE.VP for some reason, it should abort the import session (TDH.IMPORT.ABORT).

3.1.17.3.5. Aspects of MAC Calculation Compatibility on TDX Module Update

If TDH.IMPORT.STATE.VP was interrupted, and later the TDX Module was updated before TDH.IMPORT.STATE.VP was resumed, there could be incompatibility issues with the format of the intermediate MRTD MAC context, held as part of the migration stream context (MIGSC), between the original and the new TD module.

On TDH.SYS.UPDATE, the host VMM can configure the TDX Module to detect such incompatibility. If detected, TDH.IMPORT.STATE.VP returns a TDX_INCOMPATIBLE_ MBMD_MAC_CONTEXT status. In this case, the host VMM is expected to abort the migration session and may restart it. For details, see the [Base Spec] section titled “Compatibility Aspects of TD-Preserving Update”.

If the host VMM did not configure the TDX Module to detect such incompatibility, it will not be detected by TDH.IMPORT.STATE.VP. However, TDH.IMPORT.STATE.VP will detect that the MAC is incorrect and will return a TDX_INCORRECT_MBMD_MAC error.

3.1.17.3.6. Import Abort

A failed TDH.IMPORT.STATE.VP marks (except in cases where the imported TD state has not been modified) the target TD as IMPORT_FAILED; by design, it will not run. This is indicated by the FATAL bit (61) of the completion status returned in RAX.

3.1.17.3.7. S4 Resumption Abort

In addition to aborting the import of the current TD, an abort of an S4 resumption session also generates a new S4 anti-replay nonce, preventing a new S4 resumption session from starting.

3.1.17.3.8. VCPU Association

TDH.IMPORT.VP associates the TD VCPU with the current LP. This requires that the VCPU will not be associated with another LP – for details, see the [TDX Module Base Spec].

3.1.17.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.71: TDH.IMPORT.STATE.VP Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDVPR page	TDVPS	RW	Opaque	4KB	Exclusive	Shared	Shared
Explicit	R8	HPA	MBMD	MBMD	R	Shared	128B	None	None	None
Explicit	R9	HPA	Page list	PAGE_LIST	R	Shared	4KB	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Explicit	N/A	HPA	Source pages (via page list)	Blob	R	Shared	4KB	None	None	None
Implicit	N/A	HPA	TDR page	TDR	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	None	N/A	N/A

3.1.17.5. Completion Status Codes

The table below details the status values returned in RAX. The FATAL bit (61) indicates that the import session has been aborted. In such cases, the status code name is appended with a "_FATAL" suffix.

Table 3.72: TDH.IMPORT.STATE.VP Completion Status Codes (Returned in RAX) Definition

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_ALL_VCPUS_IMPORTED_FATAL	1	
TDX_INCOMPATIBLE_MBMD_MAC_CONTEXT_FATAL	1	
TDX_INCORRECT_MBMD_MAC_FATAL	1	
TDX_INTERRUPTED_RESUMABLE	0	
TDX_INVALID_MBMD_FATAL	1	
TDX_INVALID_METADATA_LIST_HEADER_FATAL	1	
TDX_INVALID_RESUMPTION	0	Applicable if the RESUME flag was set while no resumption is expected or clear while resumption is expected
TDX_INVALID_RESUMPTION_FATAL	1	Applicable if the interrupted interface function is not TDH.IMPORT.STATE.VP or PAGE_LIST_INFO is different than it was on interruption
TDX_METADATA_FIELD_ID_INCORRECT_FATAL	1	Field ID or sequence header is returned in RCX
TDX_METADATA_FIELD_NOT_WRITABLE_FATAL	1	Field ID is returned in RCX
TDX_METADATA_FIELD_VALUE_NOT_VALID_FATAL	1	Field ID is returned in RCX
TDX_METADATA_LIST_OVERFLOW_FATAL	1	
TDX_OP_STATE_INCORRECT	0	DETAILS_L2: OP_STATE
TDX_OPERAND_ADDR_RANGE_ERROR	0	DETAILS_L2: OPERAND_ID: RCX
TDX_OPERAND_BUSY	0	DETAILS_L2: OPERAND_ID: RCX, TDR, OP_STATE
TDX_OPERAND_INVALID	0	DETAILS_L2: OPERAND_ID: RCX, R10, MIGSC
TDX_OPERAND_INVALID_FATAL	1	DETAILS_L2: OPERAND_ID: R8, R9, PAGE
TDX_PAGE_METADATA_INCORRECT	0	DETAILS_L2: OPERAND_ID: RCX
TDX_REQUIRED_METADATA_FIELD_MISSING_FATAL	1	Required field ID is returned in RCX
TDX_SUCCESS	0	Operation is successful
TDX_SYS_NOT_READY	0	
TDX_SYS_SHUTDOWN	0	
TDX_TD_FATAL	0	

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_TD_KEYS_NOT_CONFIGURED	0	
TDX_TDCS_NOT_ALLOCATED	0	
TDX_VCPU_ASSOCIATED_FATAL	1	
TDX_VCPU_STATE_INCORRECT_FATAL	1	DETAILS_L2: VCPU state If this is the initial call to TDH.IMPORT.STATE.VP for this VCPU, the VCPU is expected to be in the VCPU_READY state. If this is a resumption of an interrupted call, the VCPU is expected to be in the VCPU_IMPORT state . For details, see the [Base Spec] discussion of VPCU transitions.
TDX_X2APIC_ID_NOT_UNIQUE_FATAL	1	

3.1.18. TDH.IMPORT.TRACK Leaf

TDH.IMPORT.TRACK consumes an epoch token received from the source platform. It ends the current in-order import phase epoch and either starts a new epoch or starts the out-of-order import phase.

3.1.18.1. Input Operands

Table 3.73: TDH.IMPORT.TRACK Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 84
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	HPA of the destination TD's TDR page (HKID bits must be 0)		
R8	HPA and size of memory of an MBMD structure in memory:		
	Bits	**Name**	**Description**
	51:0	HPA	Bits 51:0 of the host physical address (including HKID bits) HPA must be aligned on 128 bytes.
	63:52	Size	Size of the memory buffer containing MBMD, in bytes The buffer size must be at least the maximum MBMD size, which is 128 bytes. See 2.2.1 for details.
R10	Migration stream index – must be 0		

3.1.18.2. Output Operands

Table 3.74: TDH.IMPORT.TRACK Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
AVX, AVX2 and AVX512 state	May be reset to the architectural RESET state
Other	Unmodified

3.1.18.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.18.3.1. Overview

TDH.IMPORT.TRACK parses an epoch token received from the source platform. It checks that the epoch number indicated by the token is correct, and that all migration bundles indicated by the token have been received.

If successful, it ends the current import epoch, and as indicated by the epoch token either starts a new epoch or starts the out-of-order import phase.

3.1.18.3.2. Enumeration

Availability of TDH.IMPORT.TRACK is enumerated by either TDX_FEATURES0.TD_MIGRATION (bit 0) or TDX_FEATURES0.S4 (bit 13), readable by TDH.SYS.RD*, being set to 1. If not supported, calling TDH.IMPORT.TRACK returns a TDX_OPERAND_INVALID(RAX) status.

3.1.18.3.3. Preconditions

- An import session is in progress
- TDH.IMPORT.TRACK has not imported a start token yet.
- If a start token is being imported:
  - Mutable TD-scope state has been imported by TDH.IMPORT.STATE.TD.
  - Mutable VCPU-scope state has been imported by TDH.IMPORT.STATE.VP for all VCPUs.

3.1.18.3.4. Import Abort

A failure may mark the target TD as IMPORT_FAILED; by design, it will not run. This is indicated by the FATAL bit (61) of the completion status returned in RAX.

3.1.18.3.5. S4 Resumption

When called during an S4 resumption session, TDH.IMPORT.TRACK only supports transitioning to the out-of-order import phase, with a start token MBMD generated by TDH.EXPORT.TRACK with IN_ORDER_DONE specified.

If an error is encountered, then in addition to aborting the import of the current TD, an abort of an S4 resumption session also generates a new S4 anti-replay nonce, preventing a new S4 resumption session from starting.

3.1.18.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.75: TDH.IMPORT.TRACK Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	R	Opaque	4KB	Exclusive	Shared	Shared
Explicit	R8	HPA	MBMD buffer	MBMD	R	Shared	128B	None	None	None
Explicit	R10	Index	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A

3.1.18.5. Completion Status Codes

The table below details the status values returned in RAX. The FATAL bit (61) indicates that the import session has been aborted. In such cases, the status code name is appended with a "_FATAL" suffix.

Table 3.76: TDH.IMPORT.TRACK Completion Status Codes (Returned in RAX) Definition

Completion Status Code (RAX[63:32])	Import Abort FATAL Flag (RAX[61])	Description
TDX_INCORRECT_MBMD_MAC_FATAL	1	
TDX_INVALID_MBMD_FATAL	1	
TDX_MIGRATION_STREAM_STATE_INCORRECT	0	
TDX_OP_STATE_INCORRECT	0	DETAILS_L2: OP_STATE
TDX_OPERAND_ADDR_RANGE_ERROR	0	DETAILS_L2: OPERAND_ID: RCX
TDX_OPERAND_BUSY	0	DETAILS_L2: OPERAND_ID: RCX, MIGSC
TDX_OPERAND_INVALID	0	DETAILS_L2: OPERAND_ID: RCX, R8, R10
TDX_PAGE_METADATA_INCORRECT	0	DETAILS_L2: OPERAND_ID: RCX
TDX_SOME_VCPUS_NOT_MIGRATED_FATAL	1	
TDX_SUCCESS	0	Operation is successful
TDX_SYS_NOT_READY	0	
TDX_SYS_SHUTDOWN	0	
TDX_TD_FATAL	0	DETAILS_L2: OPERAND_ID: RCX
TDX_TD_KEYS_NOT_CONFIGURED	0	DETAILS_L2: OPERAND_ID: RCX
TDX_TDCS_NOT_ALLOCATED	0	DETAILS_L2: OPERAND_ID: RCX

5

3.1.19. TDH.MEM.SCAN.COMP/RANGE – Common

This section contains definitions that are common to TDH.MEM.SCAN.COMP and TDH.MEM.SCAN.RANGE.

3.1.19.1. GPA List-of-Lists Processing

3.1.19.1.1. Overview

GPA list-of-lists is defined in 2.2.2. It enables the host VMM to specify up to 262144 (i.e., 512^2) GPA list entries into which the scan results are written. The figure below shows how it is used by TDH.MEM.SCAN.COMP and TDH.MEM.SCAN.RANGE.

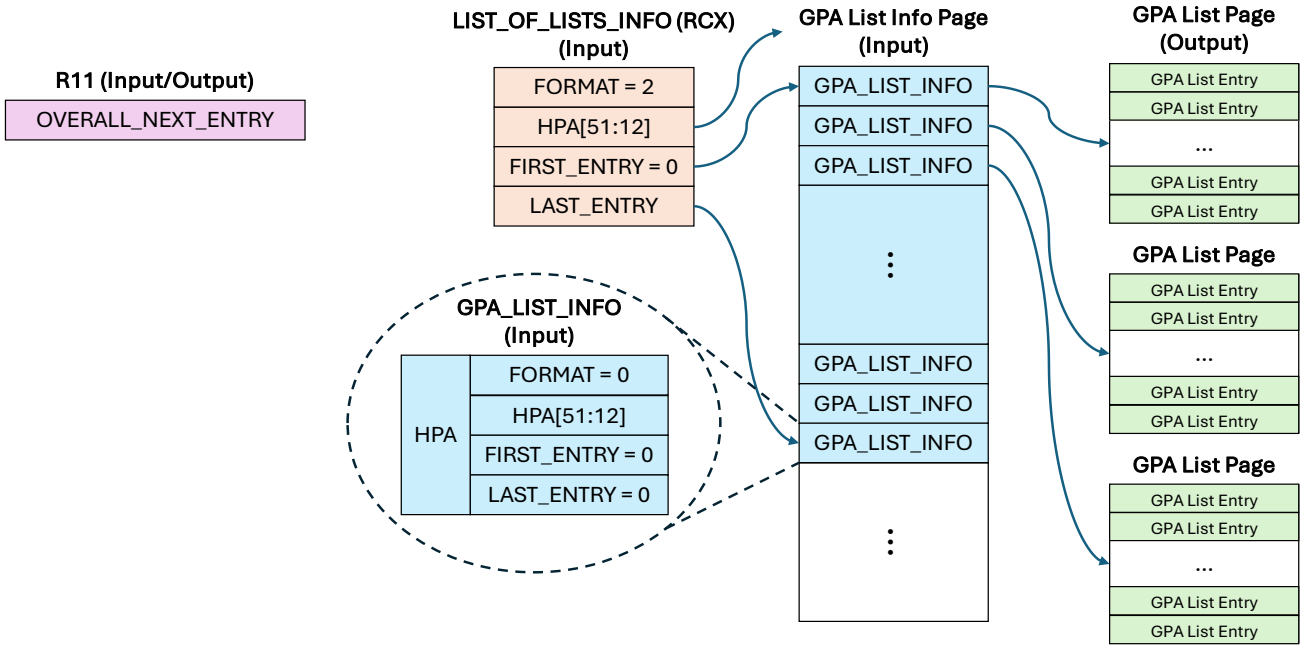

Figure 3.3: GPA List of Lists as Used by TDH.MEM.SCAN.COMP/RANGE

The host VMM can allocate up to 512 4KB GPA List pages, referenced by a single 4KB GPA List Info Page. Each GPA List page contains 512 entries.

3.1.19.1.2. LIST_OF_LISTS_INFO (Input)

The host VMM sets the LIST_OF_LISTS_INFO input (in RCX) as follows:

- HPA contains bits 51:12 of the GPA List Info page shared HPA (including HKID).
- FIRST_ENTRY must be 0.
- LAST_ENTRY contains the index of the last valid entry in the GPA List Info page.
- FORMAT must be LIST_OF_LISTS (2).

LIST_OF_LISTS_INFO is not modified by TDH.MEM.SCAN.COMP/RANGE.

3.1.19.1.3. GPA_LIST_INFO Entries (Input)

The host VMM sets each GPA_LIST_INFO entry with the 4KB-aligned shared HPA (including HKID) of a GPA List Page.

Note: This is compliant with the generic GPA_LIST_INFO format defined in 2.2.2.2:
- The HPA field contains bits 51:12 of the page HPA.
- FORMAT, FIRST_ENTRY and LAST_ENTRY must be 0.

GPA_LIST_INFO entries are not modified by TDH.MEM.SCAN.COMP/RANGE.

3.1.19.1.4. GPA List Pages (Output)

TDH.MEM.SCAN.COMP/RANGE writes entries in the GPA list pages, starting with the next available entry.

3.1.19.1.5. OVERALL_NEXT_ENTRY (Input/Output)

OVERALL_NEXT_ENTRY, provided as an input in R11, specifies the next overall (between 0 and 262143) index of the GPA List entry to be written by TDH.MEM.SCAN.COMP/RANGE. On output, R11 is updated by TDH.MEM.SCAN.COMP/RANGE

to the index of the next available entry. Thus, the output value can be used as-is as the input for the next call to TDH.MEM.SCAN.COMP/RANGE, to continue filling the GPA list-of-lists. See below for details.

Given OVERALL_NEXT_ENTRY, the following can be calculated:

- The next available GPA_LIST_INFO entry index in the GPA List Info page, which is also the index of the GPA List Page that is being filled, can be calculated as OVERALL_NEXT_ENTRY / 512.
- All previous GPA List Pages are already filled with 512 GPA List Entries.
- In the GPA List Page that being filled, the next available GPA List Entry index can be calculated as OVERALL_NEXT_ENTRY % 512.

On input, OVERALL_NEXT_ENTRY's value must be between 0 and 512 * (LIST_OF_LISTS_INFO.LAST_ENTRY + 1) - 1.

Note: On a list full condition (status is TDX_INTERRUPTED_LIST_FULL), OVERALL_NEXT_ENTRY returns with a value that is higher than the last available entry.

3.1.19.1.6. Providing an Empty GPA List-of-Lists

To start with a new, empty GPA list-of-lists, the host VMM sets OVERALL_NEXT_ENTRY to 0.

This typically needs to be done after a previous GPA list-of-lists was completely filled, i.e., the previous call to TDH.MEM.SCAN.COMP/RANGE returned with and TDX_INTERRUPTED_LIST_FULL status (see below).

3.1.19.1.7. Continuing a Partially Filled GPA List-of-Lists

TDH.MEM.SCAN.COMP/RANGE may be called to continue writing to a partially filled GPA list-of-lists, with the input LIST_OF_LISTS_INFO and OVERALL_NEXT_ENTRY values as returned by the previous call to TDH.MEM.SCAN.COMP/RANGE.

Specifically, this can be done when resuming TDH.MEM.SCAN.COMP/RANGE after an interruption, i.e., the returned status was TDX_INTERRUPTED_RESUMABLE or TDX_INTERRUPTED_BUSY.

3.1.19.1.8. List Full Interruption and Resumption

A list full condition happens when all the available GPA List entries have been written by TDH.MEM.SCAN.COMP/RANGE, and there is a need to write another entry. In this case, TDH.MEM.SCAN.COMP/RANGE returns with a TDX_INTERRUPTED_LIST_FULL status. The host VMM is expected to provide a GPA list-of-lists with available entries and resume TDH.MEM.SCAN.COMP/RANGE.

3.1.20. TDH.MEM.SCAN.COMP Leaf

Do a comprehensive scan of the TD's private GPA space and perform the requested operation.

3.1.20.1. Input Operands

Table 3.77: TDH.MEM.SCAN.COMP Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 93
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	LIST_OF_LISTS_ INFO	Information for GPA list-of-lists in shared memory. For details, see the description below, in 3.1.19.1 and in 2.2.2. LIST_OF_LISTS_INFO is only applicable for operations that require a GPA list input, e.g., when OPERATION is DCHECK (see below). For other operations (e.g., PRECLEAR), LIST_OF_LISTS_INFO must be 0. If applicable: • FORMAT must be LIST_OF_LISTS (2). • FIRST_ENTRY must be 0. • LAST_ENTRY contains the index of the last valid entry in the GPA List Info page, i.e., the provided number of GPA List Info pages - 1.		
RDX	TDR	HPA of the source TD's TDR page (HKID bits must be 0)		
R8	CONTROLS	Controls fields:		
		Bits	**Field**	**Description**
		7:0	OPERATION	Identifies the requested operation – see the table below **Enumeration:** See below for details.
		15:8	QUALIFIER	Additional qualification of the requested operation
		31:16	RESERVED	Reserved, must be 0
		47:32	CONTEXT_ID	Identifies the context of this invocation

Operand	Name	Description		
				CONTEXT_ID must be between 0 and NUM_MEM_SCAN_CONTEXTS - 1. See enumeration details below. Each concurrent invocation of TDH.MEM.SCAN.COMP must be provided with a unique CONTEXT_ID.
		55:48	RANGE_ID	Identifies the GPA range (previously configured by TDH.SCAN.CONFIG) for this invocation RANGE_ID must be between 0 and NUM_RANGES - 1 (NUM_RANGES is configured by TDH.SCAN.CONFIG).
		62:56	RESERVED	Reserved, must be 0
		63	RESUME	0: This is a new invocation of TDH.MEM.SCAN.COMP 1: This is resumption of a previously interrupted TDH.MEM.SCAN.COMP operation
R11	OVERALL_NEXT_ENTRY	The next overall index of the GPA List entry to be written by TDH.MEM.SCAN.COMP. For details, see the description below and in 3.1.19.1. OVERALL_NEXT_ENTRY is only applicable for operations that require a GPA list input, e.g., when OPERATION is DCHECK (see below). For other operations (e.g., PRECLEAR), OVERALL_NEXT_ENTRY must be 0. If applicable, OVERALL_NEXT_ENTRY must be between 0 and 512 * (LIST_OF_LISTS_INFO.LAST_ENTRY + 1) - 1.		

3.1.20.2. Output Operands

Table 3.78: TDH.MEM.SCAN.COMP Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
R11	OVERALL_NEXT_ENTRY	The overall index of the next available GPA List entry. For details, see the description below and in 3.1.19.1. OVERALL_NEXT_ENTRY is only applicable for operations that require a GPA list input, e.g., when OPERATION is DCHECK (see below). For other operations (e.g., PRECLEAR), OVERALL_NEXT_ENTRY returns 0.
Other		Unmodified

3.1.20.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.20.3.1. Overview

TDH.MEM.SCAN.COMP scans the whole TD's GPA range and performs the requested operation. Most operations fill page information in the returned GPA lists.

3.1.20.3.2. Enumeration

TDH.MEM.SCAN.COMP is supported if any of its operations are enumerated as available. See the operation modes table below for details. If not supported, calling TDH.MEM.SCAN.COMP returns a TDX_OPERAND_INVALID(RAX) status.

The supported number of memory scan contexts is enumerated by NUM_MEM_SCAN_CONTEXTS.

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.20.3.3. Preconditions

- The TDX module is configured for non-blocking export.
- An export session must be in progress.
- The comprehensive scan has been configured by TDH.MEM.SCAN.CONFIG.
- If the requested operation is PRECLEAR:
  - The TD must not have been paused by TDH.EXPORT.PAUSE.
- If the requested operation is PRECHECK:
  - The TD must not have been paused by TDH.EXPORT.PAUSE.
  - A PRECLEAR operation completed successfully.
  - The host VMM did a TLB shootdown after running the PRECLEAR operation (see 3.1.20.3.8.2 below).
- If the requested operation is DCHECK:
  - The TD must have been paused by TDH.EXPORT.PAUSE.
  - A start token has not been generated by TDH.EXPORT.TRACK.
  - If DCHECK has operation had been done, the scan state was later reset by TDH.MEM.SCAN.RESET or reconfigured by TDH.MEM.SCAN.CONFIG.
  - If PRECHECK and PRECLEAR operations have been done, they completed successfully.

3.1.20.3.4. Operation Modes and Qualifiers

TDH.MEM.SCAN.COMP supports multiple operation modes. Each operation mode may have a qualifier. The tables below provide the details.

Table 3.79: TDH.MEM.SCAN.COMP Operation Modes

OPERATION Value	Name	Description
1	DCHECK	Do a comprehensive scan of the TD's GPA range for a final list of export candidates. DCHECK is used in the final round of export, after the TD has been paused and as a prerequisite to exporting the up-to-date memory image before calling TDH.EXPORT.TRACK(DONE). DCHECK is only available if the TDX module is configured for non-blocking export. An export session must be in progress and in the blackout phase, i.e., the TD must have been paused by TDH.EXPORT.PAUSE. **Enumeration:** DCHECK support is enumerated by TDX_FEATURES0.NON_BLOCKING_EXPORT (bit 41), readable by TDH.SYS.RD* (see [ref]). DCHECK sub modes are selected by the QUALIFIER input, described in the table below.
3	PRECLEAR	Do a comprehensive scan of the TD's GPA range and clear all the non-leaf SEPT entries' Accessed bits, as preparation for running PRECHECK. PRECLEAR is only available if the TDX module is configured for non-blocking export. PRECLEAR does not return a page list. An export session must be in progress and in the live export phase, i.e., the TD must not have been paused by TDH.EXPORT.PAUSE. **Enumeration:** PRECLEAR support is enumerated by TDX_FEATURES0.NON_BLOCKING_EXPORT (bit 41), readable by TDH.SYS.RD* (see [ref]). For PRECLEAR, the QUALIFIER input must be 0.

OPERATION Value	Name	Description
4	PRECHECK	Do a comprehensive scan of the TD's GPA range and set any non-leaf SEPT entry's Accessed bit if the SEPT sub-tree under that entry contains one or more export candidates. PRECHECK is used as preparation for running DCHECK. PRECHECK is only available if the TDX module is configured for non-blocking export. An export session must be in progress and in the live export phase, i.e., the TD must not have been paused by TDH.EXPORT.PAUSE. **Enumeration:** PRECHECK support is enumerated by TDX_FEATURES0.NON_BLOCKING_EXPORT (bit 41), readable by TDH.SYS.RD* (see [ref]). PRECHECK sub modes are selected by the QUALIFIER input, described in the table below.
Other	RESERVED	Reserved

Table 3.80: QUALIFIER Definition for PRECHECK

Bit(s)	Name	Description
0	REEXPORT	If an export candidate list is requested (see LIST below), REEXPORT selects whether unexported pages are detected: **0:** Scan and return a GPA list of memory pages that either have not been exported or need to be re-exported. **1:** Scan and return a GPA list of memory pages that need to be re-exported. Pages that have not been exported are still counted by UNEXPORT_COUNT. This mode is useful for implementing post-copy (if supported).
1	LIST	Selects whether an export candidate GPA list is written, to be used for page export by TDH.EXPORT.MEM. Regardless of the value of LIST, PRECHECK sets the ACCESSED bit of non-leaf SEPT entries which are the root of SEPT sub-trees that contain export candidates. **0:** No output GPA list is returned; PRECHECK works faster. **1:** A GPA list of export candidates is returned
Other	Reserved	Must be 0

Table 3.81: QUALIFIER Definition for DCHECK

Bit(s)	Name	Description
0	REEXPORT	Selects whether unexported pages are detected: **0:** Scan and return a GPA list of memory pages that either have not been exported or need to be re-exported. **1:** Scan and return a GPA list of memory pages that need to be re-exported. Pages that have not been exported are still counted by UNEXPORT_COUNT. This mode is useful for implementing post-copy (if supported).
Other	Reserved	Must be 0

3.1.20.3.5. Comprehensive Scan Details

The whole private GPA space is scanned by multiple, possibly concurrent, invocations of TDH.MEM.SCAN.COMP.

3.1.20.3.5.1. Multiple GPA Ranges

To allow optimization for NUMA configurations, multiple GPA ranges may be configured by the host VMM using TDH.MEM.SCAN.CONFIG.

The RANGE_ID input parameter identifies the GPA range for the current invocation. RANGE_ID must be between 0 and NUM_RANGES - 1 (NUM_RANGES is configured by TDH.SCAN.CONFIG). The host VMM can invoke concurrent scanning TDH.MEM.SCAN.COMP threads of each GPA range, typically executed on LPs that are close (in the NUMA sense) to the memory range being scanned.

3.1.20.3.5.2. Scan Concurrency within a GPA Range

Within each GPA range, the host VMM can invoke multiple concurrent scanning TDH.MEM.SCAN.COMP threads. The threads balance the work among themselves.

1. When invoked, TDH.MEM.SCAN.COMP allocates a sub-range to scan. The sub-range size is configured by the host VMM using TDH.MEM.SCAN.CONFIG.
2. When done with that sub-range, if there is still unscanned memory in the range, TDH.MEM.SCAN.COMP allocates a new sub-range to scan and loops to step 1 above.
3. If there is no more memory to scan, TDH.MEM.SCAN.COMP returns to the host VMM. If there are no more concurrent scans running in the current range, it indicates a successful scan of the range.

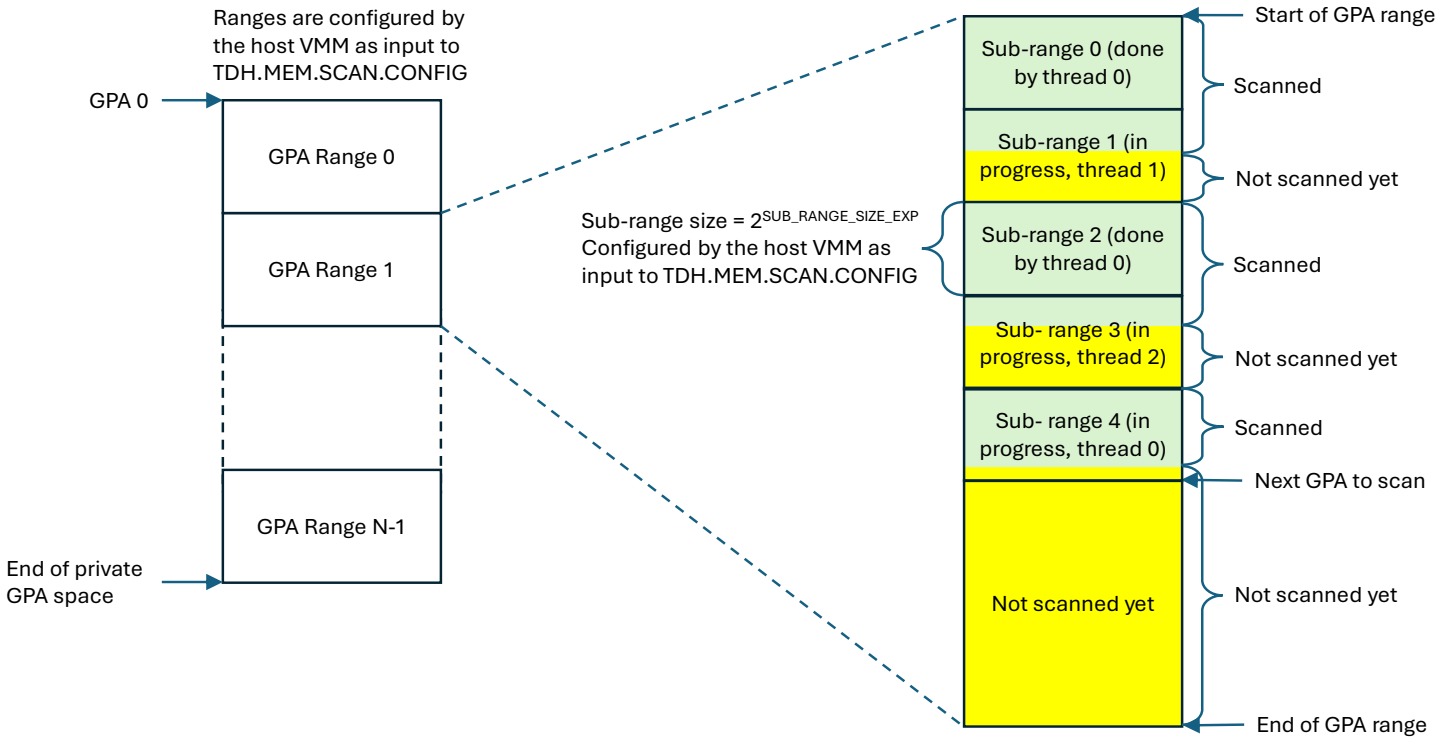

Figure 3.4: Concurrent Scan within a GPA Range

Notes:

- Each concurrent TDH.MEM.SCAN.COMP invocation must be provided with a unique CONTEXT_ID.
- The sub-range size and number of concurrent threads per range can be tuned for best performance, possibly taking into account the guest TD's memory allocation distributing within the GPA range.

3.1.20.3.5.3. Comprehensive Scan Start

Running PRECLEAR followed by PRECHECK is optional before running DCHECK. Therefore, the conditions described below identify the start of a new comprehensive scan

A new PRECLEAR comprehensive scan starts when the host VMM calls TDH.MEM.SCAN.COMP(PRECLEAR) to perform a comprehensive scan, and either of the following conditions is true:

- This is the first invocation of TDH.MEM.SCAN.COMP since TDH.MEM.SCAN.CONFIG was called, or
- This is the first invocation of TDH.MEM.SCAN.COMP since TDH.MEM.SCAN.RESET was called after TDH.MEM.SCAN.CONFIG had been called in the past.

A new PRECHECK comprehensive scan starts when the host VMM calls TDH.MEM.SCAN.COMP(PRECHECK) to perform a comprehensive scan, and the following condition is true:

- This is the first invocation of TDH.MEM.SCAN.COMP since TDH.MEM.SCAN.COMP(PRECLEAR) completed successfully.

A new DCHECK comprehensive scan starts when the host VMM calls TDH.MEM.SCAN.COMP(DCHECK) to perform a comprehensive scan, and any of the following conditions is true:

- This is the first invocation of TDH.MEM.SCAN.COMP since TDH.MEM.SCAN.CONFIG was called, or
- This is the first invocation of TDH.MEM.SCAN.COMP since TDH.MEM.SCAN.RESET was called after TDH.MEM.SCAN.CONFIG had been called in the past, or
- This is the first invocation of TDH.MEM.SCAN.COMP(DCHECK) since TDH.MEM.SCAN.COMP(PRECHECK) completed successfully.

3.1.20.3.5.4. Comprehensive Scan Context

To support multiple concurrent comprehensive scan threads, the TDX module holds multiple instances of scan contexts per TD, allocated by TDH.MEM.SCAN.CONFIG. The CONTEXT_ID parameter is used as a handle for the internally held context, to allow resumption after an incomplete operation or an interruption. It must be between 0 and NUM_MEM_SCAN_CONTEXTS - 1. See enumeration details above.

3.1.20.3.5.5. Comprehensive Scan Success

If TDH.MEM.SCAN.COMP executes successfully, the return status in RAX indicates the following:

TDX_SUCCESS	The current invocation of TDH.MEM.SCAN.COMP completed successfully, but there are concurrent instances of TDH.MEM.SCAN.COMP still scanning the current GPA range.
TDX_MEM_RANGE_SCAN_SUCCESS	Comprehensive scan of the current GPA range completed successfully, but there are concurrent instances of TDH.MEM.SCAN.COMP threads still scanning other GPA ranges. This status is returned at most once per GPA range.
TDX_MEM_SCAN_SUCCESS	Comprehensive scan of the entire GPA space completed successfully. This status is returned at most once per comprehensive scan.

3.1.20.3.5.6. Comprehensive Scan Failure

There might be cases where a TDH.MEM.SCAN.COMP instance fails in a way that fails the whole comprehensive scan. For example, this happens when a blocked SEPT entry that is discovered by TDH.MEM.SCAN.COMP(DCHECK). In such cases, TDH.MEM.SCAN.COMP returns a TDX_MEM_SCAN_FAILED_BLOCKED_RANGE status. Concurrently executing TDH.MEM.SCAN.COMP instances consequently also fail, returning a TDH_MEM_SCAN_FAILED(OTHER_THREAD_FAILED) status.

After a comprehensive scan failed, the host VMM may start a new comprehensive scan. To do that, it first needs to call TDH.COMP.SCAN.RESET to reset the internal comprehensive scan state.

3.1.20.3.5.7. Backward Compatibility

TDH.MEM.SCAN.COMP supports scanning the memory of a TD that was created by an older TDX module that didn't support TDH.MEM.SCAN.COMP, and the TDX module was later updated using a TD-preserving update. This scan may be somewhat less efficient since some information collected during the TD lifetime may not exist.

3.1.20.3.6. Concurrency

TDH.MEM.SCAN.COMP may be called concurrently on multiple LPs and may run concurrently with other functions (e.g., TDH.EXPORT.MEM). This is especially important for the DCHECK operation, which is done during the export blackout time and thus should be completed as fast as possible.

3.1.20.3.6.1. Memory Page Concurrency

For the PRECHECK operation, TDH.MEM.SCAN.COMP may not detect pages that are concurrently added to the TD, or whose state or attributes (including the Dirty bit) are concurrently modified, while scan is in progress. However, such changes set the Accessed bits in non-leaf SEPT entries, thus a following DCHECK operation will scan the applicable SEPT sub-trees and will detect the modifications.

For the DCHECK operation, concurrently changing page state in a way that impacts export correctness is prevented by the architectural restrictions on the allowed memory management operations.

3.1.20.3.6.2. SEPT Tree Concurrency

TDH.MEM.SCAN.COMP acquires a shared lock on the SEPT trees, to prevent blocking or changes to the tree structure (e.g., by TDH.MEM.RANGE.BLOCK or TDH.MEM.SEPT.REMOVE) while TDH.MEM.SCAN.COMP is running. However, this lock does not prevent SEPT tree structure modification when TDH.MEM.SCAN is not executing (e.g., it was interrupted).

For the PRECLEAR and PRECHECK operations, TDH.MEM.SCAN.COMP may prevent starvation of concurrent memory management functions by detecting that they failed to acquire an exclusive lock on the SEPT trees. In this case, TDH.MEM.SCAN.COMP yields and returns with a TDX_INTERRUPTED_BUSY Status in RAX. The host VMM is expected to resume TDH.MEM.SCAN.COMP. See the discussion on interruptibility below.

For the DCHECK operation, the non-blocking export architecture prevents SEPT tree structure change while the comprehensive scan is in progress, even if no instance of TDH.MEM.SCAN.COMP is running, by not allowing the applicable memory management functions to run.

3.1.20.3.6.3. SEPT Entry Concurrency

If TDH.MEM.SCAN.COMP fails to acquire a lock on an SEPT entry, it returns with a TDX_INTERRUPTED_BUSY status in RAX. The host VMM is expected to resume TDH.MEM.SCAN.COMP; see the discussion on interruptibility below.

3.1.20.3.7. GPA List-of-Lists Processing

Processing of the GPA list-of-list by TDH.MEM.SCAN.COMP and by TDH.MEM.SCAN.RANGE is similar; thus, it is described in 3.1.19.1.

3.1.20.3.8. TLB Tracking

3.1.20.3.8.1. Export Candidate Epoch Recording

For the PRECHECK operation, if an output GPA list is requested (QUALIFIER.LIST is 1), TDH.MEM.SCAN.COMP records the current TD epoch of each discovered export candidate (in the PAMT entry for that page). This allows TDH.EXPORT.MEM to check TLB tracking to help ensure that EPT translation caches have been flushed before the page is exported. Unlike write-blocking based export, TLB tracking is page-specific, allowing TDH.MEM.SCAN.COMP(PRECHECK) of a certain GPA range to run concurrently with TDH.EXPORT.MEM of another range.

For the PRECLEAR and DCHECK operations, TDH.MEM.SCAN.COMP does not record the current TD epoch.

3.1.20.3.8.2. Scan Epoch Recording and Checking

The host VMM is required to do a TLB shootdown after running the PRECLEAR operation and before running the PRECHECK operation. This is done so any change to the TD memory image during PRECHECK and after it will be recorded by non-leaf SEPT entries' Accessed bits. This is enforced as follows:

- After performing its scan operation, PRECLEAR records the TD epoch.
- Before starting its scan operation, PRECHECK uses the recorded epoch to check that TLB shootdown has been done.

3.1.20.3.9. Interruption and Resumption

TDH.MEM.SCAN.COMP is interruptible and resumable. An interruption occurs in the following cases:

- If a pending interrupt is detected during operation, TDH.MEM.SCAN.COMP returns with a TDX_INTERRUPTED_RESUMABLE status in RAX.
- If TDH.MEM.SCAN.COMP fails to acquire a lock on an SEPT entry, it returns with a TDX_INTERRUPTED_BUSY status in RAX.
- If TDH.MEM.SCAN.COMP executing a PRECLEAR or a PRECHECK operation detects that a concurrent function (such as TDH.MEM.PAGE.PROMOTE) has failed to acquire an exclusive lock on the SEPT trees, it may yield and return with a TDX_INTERRUPTED_BUSY status in RAX.
- If TDH.MEM.SCAN.COMP executing a PRECLEAR or a PRECHECK operation detects a blocked non-leaf SEPT entry, it returns with a TDX_INTERRUPTED_BLOCKED_RANGE status in RAX.

- If TDH.MEM.SCAN.COMP executing a PRECHECK operation detects a blocked leaf SEPT entry, it returns with a TDX_INTERRUPTED_BLOCKED_RANGE status in RAX[5].
- If TDH.MEM.SCAN.COMP detects that a concurrent function (such as TDH.EXPORT.ABORT) has failed to acquire an exclusive lock on the comprehensive memory scan state, it yields and returns with a TDX_INTERRUPTED_BUSY status in RAX.
- A GPA list-of-lists full condition has been detected. In this case, TDH.MEM.SCAN.COMP returns with a TDX_INTERRUPTED_LIST_FULL status.

In all cases, OVERALL_NEXT_ENTRY (in R11) is updated with the overall index of the next list entry to process. Except for the GPA list-of-lists full case, the host VMM may re-invoke TDH.MEM.SCAN.COMP soon after handling the interrupt. To resume after a GPA list-of-lists full case, the host VMM must provide a GPA list-of-lists with available entries. See 3.1.19.1 for details.

If an interrupted TDH.MEM.SCAN.COMP is resumed after an error was detected by a TDH.MEM.SCAN.COMP running in another context, it immediately returns with a TDX_MEM_SCAN_FAILED_OTHER_THREAD status.

Following resumption, TDH.MEM.SCAN.COMP will make some progress in its scanning of the GPA range before being interrupted again. However, some of the above interrupt conditions may prevent progress:

- A concurrent interface function holds a lock on the SEPT entry being processed.
- A concurrent interface function holds an exclusive lock on the comprehensive memory scan state.
- The GPA list-of-lists is full.

The host VMM is responsible for resolving such conditions in order for TDH.MEM.SCAN.COMP to progress.

3.1.20.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.82: TDH.MEM.SCAN.COMP Memory Operands Information

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	GPA list info page	HPA	R	Shared	4KB	None	None	None
Explicit	N/A	N/A	GPA list pages (via GPA list info page)	HPA	RW	Shared	4KB	None	None	None
Explicit	RDX	HPA	TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	R8	Unsigned Integer	Comprehensive scan context	N/A	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT tree	N/A	RW	Private	N/A	Shared	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT entry	SEPT Entry	RW	Private	N/A	Exclusive(h)	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT trees	N/A	RW	Private	N/A	Shared(i)	N/A	N/A

[5] PRECHECK may skip scanning some leaf SEPT entries if it has already set their ancestors' Accessed bits, if no GPA list output was requested.

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	GPA	L2 Secure EPT entries	SEPT Entry	RW	Private	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	N/A	Comprehensive scan state	N/A	RW	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Comprehensive scan start critical region	N/A	N/A	Hidden	N/A	Exclusive(t)	N/A	N/A

3.1.20.5. Completion Status Codes

Table 3.83: TDH.MEM.SCAN.COMP Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_INCOMPATIBLE_EXPORT_MODE_TD_NON_ACCESSIBLE	
TDX_MEM_RANGE_SCAN_SUCCESS	Comprehensive scan of the current GPA range (but not the whole TD private GPA address space) completed successfully.
TDX_MEM_SCAN_ALREADY_SUCCESSFUL	The comprehensive GPA space scan is already successful. The state needs to be reset after a previous scan, by calling TDH.COMP.SCAN.RESET. This status may be returned, in rare cases, because of concurrent TDH.MEM.SCAN.COMP invocation, if another thread has already completed the scan.
TDX_MEM_SCAN_CONFIG_REQUIRED	Comprehensive GPA space scan has not been configured by TDH.COMP.SCAN.CONFIG.
TDX_MEM_SCAN_FAILED_BLOCKED_RANGE	The DCHECK operation failed because a blocked memory range was detected.
TDX_MEM_SCAN_FAILED_OTHER_THREAD	A comprehensive GPA space scan failure was detected by TDH.MEM.SCAN running on another thread.
TDX_MEM_SCAN_SUCCESS	Comprehensive scan of the whole TD private GPA address space completed successfully.
TDX_INTERRUPTED_BLOCKED_RANGE	The PRECLEAR or PRECHECK operation encountered blocked memory range.
TDX_INTERRUPTED_BUSY	
TDX_INTERRUPTED_LIST_FULL	The output GPA lists are full.
TDX_INTERRUPTED_RESUMABLE	
TDX_INVALID_RESUMPTION	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_SUCCESS	Operation is successful.

Completion Status Code	Description
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.21. TDH.MEM.SCAN.CONFIG Leaf

Configure memory scan and add control structure pages.

3.1.21.1. Input Operands

Table 3.84: TDH.MEM.SCAN.CONFIG Input Operands Definition

Operand		Description		
RAX	LEAF_AND_VERSION	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 94
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Ignored
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	RANGE_LIST_INFO	The shared physical address (including HKID) of a of a page where memory scan control page will be created		
		Bits	**Field**	**Description**
		8:0	NUM_RANGES	Number of GPA ranges Must be between 1 and MAX_MEM_SCAN_RANGES, readable by TDH.SYS.RD*.
		11:9	RESERVED	Must be 0
		51:12	RANGE_LIST_HPA	Shared HPA (Incl. HKID) of a range list page
		63:52	RESERVED	Must be 0
RDX	TD_HANDLE	The physical address of the owner TDR page (HKID bits must be 0)		
R8	CX0_HPA	The physical address where memory scan control page 0 will be created If memory scan control page 0 has already been allocated on this platform by a previous call to TDH.MEM.SCAN.CONFIG, R8 may contain NULL_PA (-1). Else, R8 must contain a valid HPA (HKID bits must be 0).		
R9	CX1_HPA	The physical address where memory scan control page 1 will be created If the value of MEM_SCAN_CONTROL_PAGES, readable by TDH.SYS.RD*, is ≤ 1, R9 is ignored. Else, if memory scan control page 1 has already been allocated on this platform by a previous call to TDH.MEM.SCAN.CONFIG, R9 may contain NULL_PA (-1). Else, R9 must contain a valid HPA (HKID bits must be 0).		
R10	CX2_HPA	The physical address where memory scan control page 2 will be created If the value of MEM_SCAN_CONTROL_PAGES, readable by TDH.SYS.RD*, is ≤ 2, R10 is ignored.		

Operand		Description
		Else, if memory scan control page 1 has already been allocated on this platform by a previous call to TDH.MEM.SCAN.CONFIG, R10 may contain NULL_PA (-1). Else, R10 must contain a valid HPA (HKID bits must be 0).
R11	CX3_HPA	The physical address where memory scan control page 1 will be created If the value of MEM_SCAN_CONTROL_PAGES, readable by TDH.SYS.RD*, is ≤ 3, R11 is ignored. Else, if memory scan control page 3 has already been allocated on this platform by a previous call to TDH.MEM.SCAN.CONFIG, R11 may contain NULL_PA (-1). Else, R11 must contain a valid HPA (HKID bits must be 0).

3.1.21.2. Output Operands

Table 3.85: TDH.MEM.SCAN.CONFIG Output Operands Definition

Operand	Name	Description
RAX	Status	SEAMCALL instruction return code
R8	CX0_HPA	If TDH.MEM.SCAN.CONFIG terminated successfully, MEM_SCAN_CONTROL_PAGES is ≥ 1 and the page whose HPA was provided in R8 was not used as a memory scan control page (e.g., it has been allocated by a previous call to TDH.MEM.SCAN.CONFIG), R8 is updated with bit 63 set to 1. Else, R8 is unmodified.
R9	CX1_HPA	If TDH.MEM.SCAN.CONFIG terminated successfully, MEM_SCAN_CONTROL_PAGES is ≥ 2 and the page whose HPA was provided in R9 was not used as a memory scan control page (e.g., it has been allocated by a previous call to TDH.MEM.SCAN.CONFIG), R9 is updated with bit 63 set to 1. Else, R9 is unmodified.
R10	CX2_HPA	If TDH.MEM.SCAN.CONFIG terminated successfully, MEM_SCAN_CONTROL_PAGES is ≥ 3 and the page whose HPA was provided in R10 was not used as a memory scan control page (e.g., it has been allocated by a previous call to TDH.MEM.SCAN.CONFIG), R10 is updated with bit 63 set to 1. Else, R10 is unmodified.
R11	CX3_HPA	If TDH.MEM.SCAN.CONFIG terminated successfully, MEM_SCAN_CONTROL_PAGES is ≥ 4 and the page whose HPA was provided in R11 was not used as a memory scan control page (e.g., it has been allocated by a previous call to TDH.MEM.SCAN.CONFIG), R11 is updated with bit 63 set to 1. Else, R11 is unmodified.
Other		Unmodified

3.1.21.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

TDH.MEM.SCAN.CONFIG is called by the host VMM to configure the comprehensive memory scan, to be done by TDH.MEM.SCAN.COMP. It adds physical pages where scan control information used by TDH.MEM.SCAN.COMP will reside.

The scan configuration is not migrated. If a TD is migrated and then needs to be migrated again, TDH.MEM.SCAN.CONFIG must be called again. It can be invoked at any time after the TDCS pages have been allocated.

TDH.MEM.SCAN.CONFIG may be called to update a previous configuration. No instance of TDH.MEM.SCAN.COMP may be running when TDH.MEM.SCAN.CONFIG is invoked. TDH.MEM.SCAN.CONFIG resets the internal state held by the TDX module for comprehensive memory scans of the specified TD's GPA address space, similar to TDH.MEM.SCAN.RESET.

3.1.21.3.1. Enumeration

TDH.MEM.SCAN.CONFIG is supported if TDH.MEM.SCAN is supported. If not supported, calling TDH.MEM.SCAN.CONFIG returns a TDX_OPERAND_INVALID(RAX) status.

The required number of memory scan control pages is enumerated by NUM_MEM_SCAN_CONTROL_PAGES, readable by TDH.SYS.RD*.

The maximum number of GPA ranges is enumerated by MAX_MEM_SCAN_RANGES.

3.1.21.3.2. Preconditions

- The TD has been initialized by TDH.MNG.INIT, or imported and TDH.IMPORT.TRACK has imported a start token. An export session may be in progress but TDH.EXPORT.TRACK has not yet generated a start token.

3.1.21.3.3. Range List Page

The range list page holds a list of GPA ranges, to be used by TDH.MEM.SCAN. The list is formatted as an array of 64-bit entries, with entry N specifying the start of a GPA range N. The number of valid entries is specified by RCX.NUM_RANGES. See the description of TDH.MEM.SCAN.COMP in 0 for details.

Table 3.86: Range List Entry

Bits	Name	Description
20:0	RESERVED	Must be 0
50:21	RANGE_START	Bits 50:21 of the range start GPA RANGE_START must be a valid private GPA, lower than the maximum valid private GPA allowed for the TD – see the [TDX Module Base Spec] section on GPA Space Size Configuration and Virtualization. RANGE_START must be aligned on SUB_RANGE_SIZE, i.e., all bits lower than SUB_RANGE_SIZE_EXP must be 0. RANGE_START of entry 0 must be 0.
51	RESERVED	Must be 0
57:52	SUB_RANGE_SIZE_EXP	Specifies the sub-range size to be used by TDH.MEM.SCAN.COMP within the current range: $SUB\_RANGE\_SIZE = 2^{SUB\_RANGE\_SIZE\_EXP}$ SUB_RANGE_SIZE_EXP must be at between 21 (SUB_RANGE_SIZE of 2MB) and 51. Sub ranges are used by TDH.MEM.SCAN for multithreaded scanning. A SUB_RANGE_SIZE value equal or higher than the range size forces the range scan to be single threaded.
63:58	RESERVED	Must be 0

3.1.21.3.4. Memory Scan Control Pages

Up to 4 physical pages are allocated by TDH.MEM.SCAN.CONFIG, based on input parameters CX0_HPA through CX3_HPA, to hold control information for the comprehensive memory scan. The required number of scan control pages is enumerated by NUM_MEM_SCAN_CONTROL_PAGES, readable by TDH.SYS.RD*.

TDH.MEM.SCAN.CONFIG indicates whether a provided physical page is not used by it, e.g., because a scan control page has already been allocated by a previous call to TDH.MEM.SCAN.CONFIG, by setting bit 63 of the applicable output operand.

Physical control structure pages allocated by TDH.MEM.SCAN.CONFIG can only be reclaimed as part of the TD's teardown sequence.

3.1.21.3.5. Dynamic PAMT

If the TDX module is configured for dynamic PAMT, the PAMT hierarchy can be built on demand. A TDX_MISSING_PAMT_PAGE_PAIR status indicates that a PAMT page pair is missing for a control page. The host VMM may add it using TDH.PHYMEM.PAMT.ADD and retry the operation.

3.1.21.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.87: TDH.MEM.SCAN.CONFIG Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	Range list page	N/A	R	Shared	4KB	N/A	N/A	N/A
Explicit	RDX	HPA	TDR page	TDR	RW	Opaque	4KB	Shared	Shared	Shared
Explicit	R8	HPA	Control page 0[6]	N/A	RW	Opaque	4KB	Exclusive	Shared	Shared
Explicit	R9	HPA	Control page 1[6]	N/A	RW	Opaque	4KB	Exclusive	Shared	Shared
Explicit	R10	HPA	Control page 2[6]	N/A	RW	Opaque	4KB	Exclusive	Shared	Shared
Explicit	R11	HPA	Control page 3[6]	N/A	RW	Opaque	4KB	Exclusive	Shared	Shared
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	4KB	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Shared(h)	N/A	N/A
Implicit	N/A	N/A	Comprehensive scan state	N/A	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	All comprehensive scan contexts	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A
Implicit	N/A	HPA	Control page 0[7]	N/A	RW	Opaque	4KB	None	N/A	N/A
Implicit	N/A	HPA	Control page 1[7]	N/A	RW	Opaque	4KB	None	N/A	N/A
Implicit	N/A	HPA	Control page 2[7]	N/A	RW	Opaque	4KB	None	N/A	N/A
Implicit	N/A	HPA	Control page 3[7]	N/A	RW	Opaque	4KB	None	N/A	N/A

[6] Applies only if the provided HPA is used by TDH.MEM.SCAN.CONFIG to allocate a control page

[7] Applies in case the control page has been allocated by a previous call to TDH.MEM.SCAN.CONFIG

3.1.21.5. Completion Status Codes

Table 3.88: TDH.MEM.SCAN.CONFIG Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_MEM_SCAN_CONFIG_ALREADY_DONE	
TDX_MISSING_PAMT_PAGE_PAIR	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SUCCESS	Operation is successful.
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

5

3.1.22. TDH.MEM.SCAN.RANGE Leaf

Scan a range of the TD's private GPA space and perform the requested operation.

3.1.22.1. Input Operands

Table 3.89: TDH.MEM.SCAN.RANGE Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 92
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	LIST_OF_LISTS_ INFO	Information for GPA list-of-lists in shared memory. For details, see the description below, in 3.1.19.1 and in 2.2.2. LIST_OF_LISTS_INFO is only applicable for operations that require a GPA list input, i.e., when OPERATION is DSCAN (see below). For other operations (e.g., EXPORT_RESTORE), LIST_OF_LISTS_INFO must be 0. If applicable: • FORMAT must be LIST_OF_LISTS (2). • FIRST_ENTRY must be 0. • LAST_ENTRY contains the index of the last valid entry in the GPA List Info page, i.e., the provided number of GPA List Info pages - 1.		
RDX	TDR	HPA of the source TD's TDR page (HKID bits must be 0)		
R8	CONTROLS	Controls fields:		
		Bits	**Field**	**Description**
		7:0	OPERATION	Identifies the requested operation – see the table below **Enumeration:** See below for details.
		15:8	QUALIFIER	Additional qualification of the requested operation
		63:16	RESERVED	Reserved, must be 0
R9	RANGE_START	The start address of the private GPA range to scan RANGE_START must be a valid private GPA, aligned on 4KB.		

Operand	Name	Description
R10	RANGE_SIZE	The size of the GPA range to scan RANGE_SIZE must not be 0 and must be a multiple of 4KB. Bits 63:52 must be 0.
R11	OVERALL_NEXT_ENTRY	The next overall index of the GPA List entry to be written by TDH.MEM.SCAN.RANGE. For details, see the description below and in 3.1.19.1. OVERALL_NEXT_ENTRY is only applicable for operations that require a GPA list input, i.e., when OPERATION is DSCAN (see below). For other operations (e.g., EXPORT_RESTORE), OVERALL_NEXT_ENTRY must be 0. If applicable, OVERALL_NEXT_ENTRY must be between 0 and 512 * (LIST_OF_LISTS_INFO.LAST_ENTRY + 1) - 1.

3.1.22.2. Output Operands

Table 3.90: TDH.MEM.SCAN.RANGE Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
R9	NEXT_START	The first GPA for next scan, aligned on 4KB On any error or interruption: • If scan has not started yet, R9 is unmodified; it returns the RANGE_START value provided on input. • Else, R9 returns the last GPA successfully scanned so far + 1; see the description below for details. On successful end of the range scan, NEXT_START is set to the minimum of (RANGE_START + RANGE_SIZE, private GPA space size).
R10	REMAINING_SIZE	The remaining size of the GPA range to scan, in multiples of 4KB On any error or interruption: • If scan has not started yet, R10 is unmodified; it returns the RANGE_SIZE value provided on input. • Else, R10 returns RANGE_SIZE minus the size of the memory successfully scanned so far; see the description below for details. On successful end of the range scan, set to 0.
R11	OVERALL_NEXT_ENTRY	The overall index of the next available GPA List entry. For details, see the description below and in 3.1.19.1. OVERALL_NEXT_ENTRY is only applicable for operations that require a GPA list input, i.e., when OPERATION is DSCAN (see below). For other operations (e.g., EXPORT_RESTORE), OVERALL_NEXT_ENTRY returns 0.
Other		Unmodified

3.1.22.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.22.3.1. Overview

TDH.MEM.SCAN.RANGE scans the requested TD's GPA range and performs the requested operation. Most operations fill page information in the returned GPA lists.

3.1.22.3.2. Enumeration

TDH.MEM.SCAN.RANGE is supported if any of its operations are enumerated as available. See the operation modes table below for details. If not supported, calling TDH.MEM.SCAN returns a TDX_OPERAND_INVALID(RAX) status.

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.22.3.3. Preconditions

- If the requested operation is DSCAN:
  - The TDX module has been configured for non-blocking export.
- If the requested operation is EXPORT_RESTORE:
  - An export session must not be in progress.

3.1.22.3.4. Operation Modes and Qualifiers

TDH.MEM.SCAN.RANGE supports multiple operation modes as detailed below. Some TDH.MEM.SCAN.RANGE operation modes are available only if the TDX module is configured for non-blocking export. Operation modes may have qualifiers.

Table 3.91: TDH.MEM.SCAN.RANGE Operation Modes

OPERATION Value	Name	Description
0	DSCAN	Scan for export candidates. DSCAN is only available if the TDX module is configured for non-blocking export. **Enumeration:** DSCAN support is enumerated by TDX_FEATURES0.NON_BLOCKING_EXPORT (bit 41), readable by TDH.SYS.RD*. DSCAN has two sub modes, selected by the QUALIFIER input: **0: EXPORT:** Scan and return a GPA list of memory pages that either have not been exported or need to be re-exported. **1: REEXPORT:** Scan and return a GPA list of memory pages that need to be re-exported. **Other:** Reserved
2	EXPORT_RESTORE	Restore SEPT entries state after an export session has been aborted by TDH.EXPORT.ABORT. EXPORT_RESTORE does not return a page list. **Enumeration:** EXPORT_RESTORE support is enumerated by TDX_FEATURES0.SCAN_EXPORT_RESTORE (bit 43), readable by TDH.SYS.RD*. For EXPORT_RESTORE, QUALIFIER must be 0.
Other	RESERVED	Reserved

3.1.22.3.4.1. DSCAN Details

DSCAN skips SEPT sub-tree that are in a BLOCKED state, i.e.:

- If DSCAN encounters a blocked non-leaf SEPT entry, it skips it (and the whole GPA range that is mapped by it).
- If DSCAN encounters a blocked leaf SEPT entry, it skips that entry.

3.1.22.3.4.2. EXPORT_RESTORE Details

EXPORT_RESTORE stops if there is an SEPT sub-tree that is blocked. If EXPORT_RESTORE encounters non-leaf SEPT entry whose state is NL_BLOCKED, it aborts and returns a TDX_GPA_RANGE_BLOCKED status. The host VMM can determine the blocked range using the NEXT_START output operand.

EXPORT_RESTORE does not stop if a leaf SEPT entry is blocked.

3.1.22.3.5. Concurrency

TDH.MEM.SCAN.RANGE may be called concurrently on multiple LPs and may run concurrently with other functions (e.g., TDH.EXPORT.MEM).

3.1.22.3.5.1. Memory Page Concurrency

TDH.MEM.SCAN.RANGE may not detect pages that are concurrently added to the TD, or whose state or attributes (including the Dirty bit) are concurrently modified, while scan is in progress.

3.1.22.3.5.2. SEPT Tree Concurrency

TDH.MEM.SCAN.RANGE acquires a shared lock on the SEPT trees, to prevent blocking or changes to the tree structure (e.g., by TDH.MEM.RANGE.BLOCK or TDH.MEM.SEPT.REMOVE) while TDH.MEM.SCAN.RANGE is in progress. Note that this lock does not prevent SEPT tree structure modification when TDH.MEM.SCAN.RANGE is not executing (e.g., it was interrupted).

TDH.MEM.SCAN.RANGE may prevent starvation of concurrent memory management functions by detecting that they failed to acquire an exclusive lock on the SEPT trees. In this case, TDH.MEM.SCAN.RANGE yields and returns with a TDX_INTERRUPTED_BUSY Status in RAX. The host VMM is expected to resume TDH.MEM.SCAN.RANGE. See the discussion on interruptibility below.

3.1.22.3.5.3. SEPT Entry Concurrency

If TDH.MEM.SCAN.RANGE fails to acquire a lock on an SEPT entry, behavior depends on the requested operation:

- For a DSCAN operation, TDH.MEM.SCAN.RANGE skips the busy SEPT entry. No status is written to the GPA list.
  Note: Non-blocking export is designed to ensure that such entries will be scanned at least once by DCHECK. For details, see the [TD Migration Spec].
- For an EXPORT_RESTORE operation, TDH.MEM.SCAN.RANGE returns with a TDX_INTERRUPTED_BUSY Status in RAX. The host VMM is expected to resume TDH.MEM.SCAN.RANGE. See the discussion on interruptibility below.

3.1.22.3.6. GPA List-of-Lists Processing

Processing of the GPA list-of-list by TDH.MEM.SCAN.COMP and by TDH.MEM.SCAN.RANGE is similar; thus, it is described in 3.1.19.1.

3.1.22.3.7. TLB Tracking

For the DSCAN operation, TDH.MEM.SCAN.RANGE records the current TD epoch (in the PAMT entry for that page). This allows TDH.EXPORT.MEM to check TLB tracking to help ensure that EPT translation caches have been flushed before the page is exported. Unlike write-blocking based export, TLB tracking is page-specific, allowing TDH.MEM.SCAN.RANGE(DSCAN) of a certain GPA range to run concurrently with TDH.EXPORT.MEM of another range.

For the EXPORT_RESTORE operation, TDH.MEM.SCAN.RANGE does not record the current TD epoch.

3.1.22.3.8. Interruption and Resumption

TDH.MEM.SCAN.RANGE is interruptible and resumable. An interruption occurs in the following cases:

- If a pending interrupt is detected during operation, TDH.MEM.SCAN.RANGE returns with a TDX_INTERRUPTED_RESUMABLE status in RAX.
- If TDH.MEM.SCAN.RANGE fails to acquire a lock on an SEPT entry during an EXPORT_RESTORE operation, it returns with a TDX_INTERRUPTED_BUSY Status in RAX.
- If TDH.MEM.SCAN.RANGE detects that a concurrent function (such as TDH.MEM.PAGE.PROMOTE) has failed to acquire an exclusive lock on the SEPT trees, it may yield and return with a TDX_INTERRUPTED_BUSY Status in RAX.
- A GPA list-of-lists full condition has been detected. In this case, TDH.MEM.SCAN.COMP returns with a TDX_INTERRUPTED_LIST_FULL status.

In all cases, the following are updated with the range information and the next list entry index to process:

- NEXT_START (in R9)
- REMAINING_SIZE (in R10)
- OVERALL_NEXT_ENTRY (in R11)

Except for the GPA list-of-lists full case, the host VMM may re-invoke TDH.MEM.SCAN.RANGE soon after handling the interrupt. To resume after a GPA list-of-lists full case, the host VMM must provide a list-of-lists with available entries. See 3.1.19.1 for details.

3.1.22.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.92: TDH.MEM.SCAN.RANGE Memory Operands Information

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	GPA list info page	HPA	RW	Shared	4KB	None	None	None
Explicit	N/A	N/A	GPA list pages (via GPA list info page)	HPA	RW	Shared	4KB	None	None	None
Explicit	RDX	HPA	TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	R9	GPA	GPA range	GPA	R	Private	4KB	None	None	None
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT tree	N/A	RW	Private	N/A	Shared	N/A	N/A
Implicit	N/A	GPA	L1 Secure EPT entry	SEPT Entry	RW	Private	N/A	Exclusive(h)	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT trees	N/A	RW	Private	N/A	Shared(i)	N/A	N/A
Implicit	N/A	GPA	L2 Secure EPT entries	SEPT Entry	RW	Private	N/A	Exclusive(i)	N/A	N/A

3.1.22.5. Completion Status Codes

Table 3.93: TDH.MEM.SCAN.RANGE Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_GPA_RANGE_BLOCKED	Returned only if OPERATION is EXPORT_RESTORE
TDX_INCOMPATIBLE_EXPORT_MODE_TD_NON_ACCESSIBLE	
TDX_INTERRUPTED_BUSY	
TDX_INTERRUPTED_LIST_FULL	The output GPA lists are full.
TDX_INTERRUPTED_RESUMABLE	

Completion Status Code	Description
TDX_MEM_SCAN_FAILED_BLOCKED_RANGE	
TDX_MEM_SCAN_FAILED_OTHER_THREAD	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_SUCCESS	Operation is successful.
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.23. TDH.MEM.SCAN.RESET Leaf

Reset the TDX module's comprehensive memory scan internal state for the specified TD.

3.1.23.1. Input Operands

Table 3.94: TDH.MEM.SCAN.RESET Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Version	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 95
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Ignored
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RDX	TDR	HPA of the source TD's TDR page (HKID bits must be 0)		

3.1.23.2. Output Operands

Table 3.95: TDH.MEM.SCAN.RESET Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
Other		Unmodified

3.1.23.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.23.3.1. Overview

TDH.MEM.SCAN.RESET resets the internal state held by the TDX module for comprehensive memory scans of the specified TD's GPA address space. It is typically used if a scan by TDH.MEM.SCAN.COMP failed, before attempting a new scan.

3.1.23.3.2. Enumeration

TDH.MEM.SCAN.RESET is supported if TDH.MEM.SCAN.COMP is supported. If not supported, calling TDH.MEM.SCAN.RESET returns a TDX_OPERAND_INVALID(RAX) status.

3.1.23.3.3. Preconditions

- The TD has been finalized by TDH.MR.FINALIZE.
- A migration session may be in progress but TDH.EXPORT.TRACK has not yet generated a start token.

3.1.23.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.96: TDH.MEM.SCAN.RANGE Memory Operands Information

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RDX	HPA	TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	R9	GPA	GPA range	GPA	R	Private	4KB	None	None	None
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Comprehensive scan state	N/A	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	All comprehensive scan contexts	N/A	RW	Opaque	N/A	Exclusive(i)	N/A	N/A

5

3.1.23.5. Completion Status Codes

Table 3.97: TDH.MEM.SCAN.RESET Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_SUCCESS	Operation is successful.
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.24. TDH.MIG.SETUP Leaf

Unreleased Feature:	TDH.MIG.SETUP is an interface function that will be provided with the Migration Setup service, a feature which has not been released yet at the time of writing of this document. Details related to that feature serve as a preview and are subject to change.

Set up a TD migration session.

3.1.24.1. Input Operands

Table 3.98 TDH.MIG.SETUP Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Number	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 76
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	TD Handle	The physical address of the target TD's TDR page (HKID bits must be 0)		
RDX	CONTROLS	Control fields:		

Operand	Name	Description		
		Bit(s)	**Field**	**Description**
		31:0	Reserved	Must be 0
		32	DIRECTION	Direction of AKE: 0: INITIATOR – called on the migration source platform 1: RESPONDER – called on the destination platform DIRECTION must have the same value for all calls to TDH.MIG.SETUP done as part of the same session.
		39:33	IN_BUFF_TYPE	Type of the message in the host input buffer, as defined below If no input buffer is provided, IN_BUFF_TYPE must be EMPTY (0).
		63:40	Reserved	Must be 0
R8	IN_BUFF_HPA_LIST	The shared HPA of the first page in an HPA_LINKED_LIST (defined in the [ABI Spec]) of memory pages containing the input buffer If no input buffer is provided, IN_BUFF_HPA_LIST should be set to NULL_PA (-1). The page list entries themselves are Shared HPAs, aligned on 4KB. The input buffer format depends on IN_BUFF_TYPE. See below for details. See enumeration details below.		
R9	IN_BUFF_ACTUAL_SIZE	The actual size of host input buffer, in bytes If no input buffer is provided, IN_BUFF_ACTUAL_SIZE must be 0. IN_BUFF_ACTUAL_SIZE must not be higher than 4096 * MIG_SETUP_MAX_BUFF_PAGES. See enumeration details below.		
R10	OUT_BUFF_HPA_LIST	The shared HPA of the first page in an HPA_LINKED_LIST (defined in the [ABI Spec]) of memory pages containing the output buffer The page list entries themselves are Shared HPAs, aligned on 4KB. The output buffer format depends on the returned status. See below for details.		
R11	OUT_BUFF_SIZE	The overall size of host output buffer, in bytes OUT_BUFF_SIZE must be at least 4096 * MIG_SETUP_MAX_BUFF_PAGES. See enumeration details below.		

3.1.24.2. Output operands

Table 3.99: TDH.MIG.SETUP Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
RCX	OUT_BUFF_ACTUAL_SIZE	Actual size of the data in the output buffer, in bytes: • If STATUS is TDX_MIG_SETUP_MSG_EXCHANGE, RCX returns the size of Migration Setup Message.

Operand	Name	Description
		• If STATUS is TDX_MIG_SETUP_GET_QUOTE, RCX returns the size of provided TDREPORT_STRUCT in bytes. • In all other cases, RCX returns 0.
Other		Unmodified

3.1.24.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.24.3.1. Overview

TDH.MIG.SETUP is called multiple times as part of a Migration Stup session, to prepare for a TD migration session, including securely programming the migration keys between the source and destination upon successful completion.

A Migration Setup session is bound to the migrated TD. A session starts when TDH.MIG.SETUP is called and there is no currently active Migration Setup session. A session ends when TDH.MIG.SETUP returns TDX_SUCCESS or an error code such as TDX_MIG_POLICY_FAILURE (refer to the list of status codes below). A session can be aborted by the host VMM, by calling TDH.MIG.SETUP.ABORT.

For a general description of the TDX Module's session concept, see the [Base Spec] section titled "Sessions".

3.1.24.3.2. Enumeration

Support of TDH.MIG.SETUP is enumerated by FEATURES_ENABLE0.MIG_SETUP (bit 55), readable by TDH.SYS.RD*.
The maximum size of MIG_SETUP_MSG is enumerated by MIG_SETUP_MSG_MAX_SIZE.

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

The maximum value of IN_BUFF_ACTUAL_SIZE and the minimum value of OUT_BUF_SIZE (when a buffer is provided), in 4KB pages, is enumerated by MIG_SETUP_MAX_BUFF_PAGES.

3.1.24.3.3. Preconditions

- For export, on the source platform, TDH.MIG.SETUP should be called after the host VMM wrote the TD's MIG_SETUP_TD_POLICY_HASH using TDH.MNG.WR and finalized the TD using TDH.MR.FINALIZE, and before the export session starts by TDH.EXPORT.STATE.IMMUTABLE. TDH.MIG.SETUP may also be called during the post copy phase of an import session, to set up a new export session.
- For import, on the destination platform, TDH.MIG.SETUP should be called after the TD has been created and before the import session starts by TDH.IMPORT.STATE.IMMUTABLE.

3.1.24.3.4. Input and Output Data Types and Formats

The input data format depends on the specified IN_BUFF_TYPE, as described in the table below.

Table 3.100: Input Data Types and Formats

IN_BUFF_TYPE		Description		
Value	**Name**			
0	EMPTY	No input data.		
1	MIG_POLICIES	The input buffer contains one or two migration policies, as a set of the Tag-Length-Value (TLV) structures. Each TLV structure is formatted as follows:		
		Name	**Size (Bytes)**	**Description**
		TAG	4	TD migration policy type: **1:** Platform Migration Policy **2:** TD's Migration Policy **Other:** Reserved

IN_BUFF_TYPE		Description		
Value	Name			
		LENGTH	4	The length of the POLICY field below, in bytes
		POLICY	Value of LENGTH	The migration policy – detailed definitions are provided in **Error! Reference source not found.**.
2	MIG_SETUP_MSG	The input buffer contains a migration setup message blob from the TDX Module on the other side of the migration session.		
3	QUOTE	The input buffer contains a TDX Quote.		
Other	Reserved			

The output data format depends on the returned status, as described in the table below.

Table 3.101: Output Data Types and Formats

Returned STATUS	Description
TDX_MIG_SETUP_MSG_EXCHANGE, TDX_MIG_SETUP_MSG_SEND	The output buffer contains a migration setup message blob, to be sent to the TDX Module on the other side of the migration session.
TDX_MIG_SETUP_GET_QUOTE	The output buffer contains a TDREPORT_STRUCT to be used as an input form creating a TDX Quote.
Other	No output data.

3.1.24.3.5. Initial Invocation on the Source Platform

On the initial TDH.MIG.SETUP invocation on the source platform, the host VMM specifies it to be the INITIATOR of the AKE protocol by setting the DIRECTION input flag to 0.

The input buffer must contain migration policies, i.e., IN_BUFF_TYPE must be MIG_POLICIES (1):

- The migration policy associated with the TD must be provided.
- The source platform's migration policy may be provided.

TDH.MIG.SETUP verifies that the SHA-384 hash of the provided TD migration policy matches the TD's CUR_SERVTD_HASH. In case of a mismatch, a TDX_MIG_SETUP_POLICY_HASH_MISMATCH error code is returned.

3.1.24.3.6. Initial Invocation on the Destination Platform

On the initial TDH.MIG.SETUP invocation on the destination platform, the host VMM specifies it to be the RESPONDER of the AKE protocol by setting the DIRECTION input flag to 1.

The input buffer may either be empty (IN_BUFF_TYPE = EMPTY (0)) or contain the destination platform's migration policy (IN_BUFF_TYPE = MIG_POLICIES (1)).

3.1.24.3.7. Migration Setup Message Exchange

During the migration setup session, source and destination TDX Modules would need to exchange migration setup messages via a network communication channel provided by the host VMM. Message exchange is driven by status codes returned by TDH.MIG.SETUP, as described in the table below.

Table 3.102: TDH.MIG.SETUP Message Exchange Status Codes and Expected Host VMM Behavior

Completion Status Code	Expected Host VMM Behavior
TDX_MIG_SETUP_MSG_EXCHANGE	The host VMM is expected to send the message in the output buffer to the TDX Module on the other side on the migration session, wait for a response, then call TDH.MIG.SETUP again with the response in the input buffer, i.e., IN_BUFF_TYPE = MIG_SETUP_MSG (2).

Completion Status Code	Expected Host VMM Behavior
TDX_MIG_SETUP_MSG_SEND	The host VMM is expected to send the message in the output buffer to the TDX Module on the other side on the migration session, then call TDH.MIG.SETUP again with no input buffer, i.e., IN_BUFF_TYPE = EMPTY (0).
TDX_MIG_SETUP_MSG_RECEIVE	The output buffer is empty. The host VMM is expected to wait for a message from the TDX Module on the other side of the migration session, then call TDH.MIG.SETUP again with the message in the input buffer, i.e., IN_BUFF_TYPE = MIG_SETUP_MSG (2).
Other	No message exchange

3.1.24.3.8. Getting Quotes

During the migration setup session, both source and destination would need to request a TDX Quote to be generated by the platform's quoting service, based on a TDREPORT_STRUCT of the Migration Setup service.

To do this, TDH.MIG.SETUP returns a TDX_MIG_SETUP_GET_QUOTE status and writes the TDREPORT_STRUCT into the output buffer. Upon receiving this status, the host VMM is expected to use the platform's Quoting service to generate a Quote. Depending on the platform, this may be an SGX Quoting Enclave or a Quoting service provided by the TDX Module (TDH.QUOTE.GET). The host VMM should then invoke TDH.MIG.SETUP providing Quote in the input buffer.

3.1.24.3.9. Interruption and Resumption

TDH.MIG.SETUP is interruptible and resumable. If a pending interrupt is detected during operation, TDH.MIG.SETUP returns with a TDX_INTERRUPTED_RESUMABLE status in RAX.

Normally, the host VMM resumes TDH.MIG.SETUP by calling it again with the same TD Handle in RCX. All the other input operand are ignored.

If the host VMM decides not to resume TDH.MIG.SETUP, it needs to call TDH.MIG.SETUP.ABORT to abort the Migration Setup session.

3.1.24.3.10. Session Abort

The host VMM can abort an ongoing Migration Setup session at any time TDH.MIG.SETUP is not running by calling TDH.MIG.SETUP.ABORT. See 3.1.25 for details.

3.1.24.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.103: TDH.MIG.SETUP Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	RW	Opaque	4KB	Shared	Shared	Shared
Explicit	R8	HPA	Input buffers list	N/A	R	Shared	4KB	None	None	None
Explicit	N/A	HPA	Input pages	N/A	R	Shared	4KB	None	None	None
Explicit	R10	HPA	Output buffers list	N/A	R	Shared	4KB	None	None	None
Explicit	N/A	HPA	Output pages	N/A	RW	Shared	4KB	None	None	None
Implicit	N/A	Index	Migration setup context	MIG_SETUP_CONTEXTS	RW	Hidden	N/A	Exclusive(i)	N/A	N/A

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	4KB	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Shared(h)	N/A	N/A
Implicit	N/A	N/A	Migration setup	N/A	RW	Hidden	N/A	Exclusive	N/A	N/A

3.1.24.5. Completion Status Codes

Table 3.104: TDH.MIG.SETUP Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_EXT_FATAL_ERROR	Migration Setup session has ended unsuccessfully.
TDX_EXT_NOT_INITIALIZED	
TDX_EXT_NOT_READY	
TDX_INCONSISTENT_MSR	IA32_TSC_ADJUST MSR value is different than the value captured during the TDH.SYS.INIT interface function.
TDX_INTERRUPTED_RESUMABLE	See the interruption and resumption description in 3.1.24.3.9 above.
TDX_LIMIT_CPUID_MAXVAL_SET	IA32_MISC_ENABLES MSR bit 22 (Limit CPUID Maxval) is set.
TDX_MIG_SETUP_ABSENT_TD_POLICY	Migration Setup session has ended unsuccessfully.
TDX_MIG_SETUP_AKE_FAILURE	Migration Setup session has ended unsuccessfully.
TDX_MIG_SETUP_GET_QUOTE	See the quoting description in 3.1.24.3.8 above.
TDX_MIG_SETUP_MSG_EXCHANGE	See the message exchange description in 3.1.24.3.7 above.
TDX_MIG_SETUP_MSG_RECEIVE	See the message exchange description in 3.1.24.3.7 above.
TDX_MIG_SETUP_MSG_SEND	See the message exchange description in 3.1.24.3.7 above.
TDX_MIG_SETUP_PLATFORM_POLICY_INVALID	Migration Setup session has ended unsuccessfully.
TDX_MIG_SETUP_POLICY_FAILURE	Migration Setup session has ended unsuccessfully.
TDX_MIG_SETUP_QUOTE_INVALID	Migration Setup session has ended unsuccessfully.
TDX_MIG_SETUP_SESSION_ABORTING	
TDX_MIG_SETUP_SESSION_ACTIVE	
TDX_MIG_SETUP_TD_POLICY_HASH_MISMATCH	Migration Setup session has ended unsuccessfully.
TDX_MIG_SETUP_TD_POLICY_HASH_ZERO	Migration Setup session has not started.
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
TDX_OP_STATE_INCORRECT	

Completion Status Code	Description
TDX_SUCCESS	Migration Setup session has ended successfully.
TDX_TSC_ROLLBACK	

3.1.25. TDH.MIG.SETUP.ABORT Leaf

Unreleased Feature:	TDH.MIG.SETUP.ABORT is an interface function that will be provided with the Migration Setup service, a feature which has not been released yet at the time of writing of this document. Details related to that feature serve as a preview and are subject to change.

Abort a Migration Setup session.

3.1.25.1. Input Operands

Table 3.105 TDH.MIG.SETUP.ABORT Input Operands Definition

Operand	Name	Description		
RAX	Leaf and Number	SEAMCALL instruction leaf number and version		
		Bits	**Field**	**Description**
		15:0	Leaf Number	Selects the SEAMCALL interface function: 77
		23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
		24	INTERRUPT_MODE	Pending interrupts detection mode: 0: Pending interrupts are detected only if the host VMM's RFLAGS.IF is 1. 1: Pending interrupts are detected regardless of the host VMM's RFLAGS.IF status. See enumeration details below.
		62:25	Reserved	Must be 0
		63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	TD Handle	The physical address of the target TD's TDR page (HKID bits must be 0)		
RDX	CONTROLS	Control fields:		
		Bit(s)	**Field**	**Description**
		63:0	Reserved	Must be 0

3.1.25.2. Output operands

Table 3.106: TDH.MIG.SETUP.ABORT Output Operands Definition

Operand	Name	Description
RAX	STATUS	SEAMCALL instruction return code
Other		Unmodified

3.1.25.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.25.3.1. Overview

TDH.MIG.SETUP.ABORT aborts a Migration Setup session.

3.1.25.3.2. Enumeration

Support of TDH.MIG.SETUP.ABORT is enumerated by FEATURES_ENABLE0.MIG_SETUP (bit 55), readable by TDH.SYS.RD*.

Support of INTERRUPT_MODE is enumerated by TDX_FEATURES0.HOST_SIDE_INTERRUPT_MODE (bit 62).

3.1.25.3.3. Preconditions

- The Migration Setup session indicated by SESSION_ID is active.

3.1.25.3.4. Interruptibility

TDH.MIG.SETUP.ABORT is interruptible and resumable. If a pending interrupt is detected during operation, TDH.MIG.SETUP.ABORT returns with a TDX_INTERRUPTED_RESUMABLE status in RAX.

The host VMM is expected to resume TDH.MIG.SETUP.ABORT by calling it again with the same TD Handle in RCX and same SESSION_ID in RDX. All the other input operand are ignored.

3.1.25.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.107: TDH.MIG.SETUP.ABORT Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	TDR page	TDR	RW	Opaque	4KB	Shared	Shared	Shared
Explicit	RDX	Index	Migration setup context	N/A	RW	Hidden	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	4KB	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Shared(h)	N/A	N/A
Implicit	N/A	N/A	Migration setup	N/A	RW	Hidden	N/A	Exclusive	N/A	N/A
Implicit	N/A	Index	Migration setup context	N/A	RW	Hidden	N/A	Exclusive(i)	N/A	N/A

3.1.25.5. Completion Status Codes

Table 3.108: TDH.MIG.SETUP.ABORT Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_EXT_NOT_INITIALIZED	
TDX_INCONSISTENT_MSR	IA32_TSC_ADJUST MSR value is different than the value captured during the TDH.SYS.INIT interface function. Indicates an unsuccessful Migration Setup session end.
TDX_INTERRUPTED_RESUMABLE	
TDX_LIMIT_CPUID_MAXVAL_SET	IA32_MISC_ENABLES MSR bit 22 (Limit CPUID Maxval) is set.

Completion Status Code	Description
TDX_MIG_SETUP_SESSION_INACTIVE	
TDX_OPERAND_BUSY	
TDX_OPERAND_INVALID	
OP_STATE_INCORRECT	
TDX_SUCCESS	Indicates a successful Migration Setup session abort.
TDX_TSC_ROLLBACK	

3.1.26. TDH.MIG.STREAM.CREATE Leaf

Create a Migration Stream and its MIGSC control structure.

3.1.26.1. Input Operands

Table 3.109: TDH.MIG.STREAM.CREATE Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 96
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	The physical address of a page where MIGSC will be created		
RDX	The physical address of the owner TDR page (HKID bits must be 0)		

3.1.26.2. Output Operands

Table 3.110: TDH.MIG.STREAM.CREATE Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
Other	Unmodified

3.1.26.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.26.3.1. Overview

TDH.MIG.STREAM.CREATE creates a new Migration Stream and its MIGSC control structure. This function can be invoked at any time after the TDCS pages have been allocated.

TDH.MIG.STREAM.CREATE can only be successfully invoked if no migration session is in progress.

3.1.26.3.2. Enumeration

Availability of TDH.MIG.STREAM.CREATE is enumerated by TDX_FEATURES0.TD_MIGRATION (bit 0), readable by TDH.SYS.RD*. If not supported, calling TDH.EXPORT.ABORT returns a TDX_OPERAND_INVALID(RAX) status.

The maximum number of migration streams per TD is enumerated by MAX_MIGS.

3.1.26.3.3. Preconditions

- No migration session is in progress.
- The number of already created migration streams is lower than the maximum allowed.

3.1.26.3.4. Dynamic PAMT

If the TDX Module is configured for dynamic PAMT, the PAMT hierarchy can be built on demand. A TDX_MISSING_PAMT_PAGE_PAIR status indicates that a PAMT page pair is missing for the new MIGSC page. The host VMM may add it using TDH.PHYMEM.PAMT.ADD and retry the operation.

3.1.26.3.5. Control Structure Pages

Physical MIGSC pages allocated by TDH.MIG.STREAM.CREATE can only be reclaimed as part of the TD's teardown sequence.

3.1.26.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.111: TDH.MIG.STREAM.CREATE Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	MIGSC page	MIGSC	RW	Opaque	4KB	Exclusive	Shared	Shared
Explicit	RDX	HPA	TDR page	TDR	RW	Opaque	4KB	Shared	Shared	Shared
Implicit	N/A	N/A	TDCS structure	TDCS	RW	Opaque	4KB	Shared(i)	N/A	N/A
Implicit	N/A	N/A	TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Exclusive(h)	N/A	N/A
Implicit	N/A	N/A	Migration context	N/A	RW	Opaque	N/A	Exclusive	N/A	N/A
Implicit	N/A	N/A	Mig. Stream context	Mig. Stream context	RW	Opaque	N/A	Exclusive(i)	N/A	N/A

3.1.26.5. Completion Status Codes

Table 3.112: TDH.MIG.STREAM.CREATE Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_MISSING_PAMT_PAGE_PAIR	
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SUCCESS	TDH.MIG.STREAM.CREATE is successful.
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	

Completion Status Code	Description
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.27. TDH.SERVTD.BIND Leaf

Bind a service TD to a target TD.

3.1.27.1. Input Operands

Table 3.113: TDH.SERVTD.BIND Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 48
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	The physical address of the target TD's TDR page (HKID bits must be 0)		
RDX	The physical address of the service TD's TDR page (HKID bits must be 0)		
R8	Index (slot number) in the target TD's service TD binding table		
R9	SERVTD_TYPE: Service TD type		
R10	SERVTD_ATTR: Service TD attributes		

3.1.27.2. Output Operands

Table 3.114: TDH.SERVTD.BIND Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
RCX	Binding handle In case of an error, as indicated by RAX, RCX returns 0.
R10	TD_UUID bits 63:0 In case of an error, as indicated by RAX, R10 returns 0.
R11	TD_UUID bits 127:64 In case of an error, as indicated by RAX, R11 returns 0.
R12	TD_UUID bits 191:128 In case of an error, as indicated by RAX, R12 returns 0.
R13	TD_UUID bits 255:192 In case of an error, as indicated by RAX, R13 returns 0.

Operand	Description
AVX, AVX2 and AVX512 state	May be reset to the architectural RESET state
Other	Unmodified

3.1.27.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.27.3.1. Overview

TDH.SERVTD.BIND binds a service TD to a target TD.

3.1.27.3.2. Enumeration

Availability of TDH.SERVTD.BIND is enumerated by TDX_FEATURES0.SERVICE_TD (bit 2), readable by TDH.SYS.RD*. If not supported, calling TDH.SERVTD.BIND returns a TDX_OPERAND_INVALID(RAX) status.

3.1.27.3.3. Preconditions

- The target TD has been created and the TDCS pages have been allocated.
- If the service TD has been pre-bound by TDH.SERVTD.PERBID:
  - The target TD has not been paused for export and is not in the in-order import phase.
- Else:
  - The target TD has not been finalized by TDH.MR.FINALIZE.
- The service TD has been finalized by TDH.MR.FINALIZE.

3.1.27.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.115: TDH.SERVTD.BIND Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	Target TD's TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	RDX	HPA	Service TD's TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Implicit	N/A	N/A	Target TD's TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	Target TD's TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Binding table		RW	Opaque	N/A	Exclusive	None	None
Implicit	N/A	N/A	Service TD's TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	Service TD's TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared	N/A	N/A

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	N/A	Service TD's TDCS.RTMR	SHA384_ HASH	N/A	Opaque	N/A	Shared	N/A	N/A

3.1.27.5. Completion Status Codes

Table 3.116: TDH.SERVTD.BIND Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SERVTD_ATTR_MISMATCH	
TDX_SERVTD_CANNOT_BE_MIGRATABLE	
TDX_SERVTD_INFO_HASH_MISMATCH	
TDX_SERVTD_NESTING_NOT_ALLOWED	
TDX_SERVTD_TYPE_MISMATCH	
TDX_SERVTD_UUID_MISMATCH	
TDX_SUCCESS	TDH.SERVTD.BIND is successful.
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

5

3.1.28. TDH.SERVTD.REBIND Leaf

Rebind a new service TD to a target TD.

3.1.28.1. Input Operands

Table 3.117: TDH.SERVTD.REBIND Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 97
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	The physical address of the target TD's TDR page (HKID bits must be 0)		
RDX	The physical address of the service TD's TDR page (HKID bits must be 0)		
R8	Index (slot number) in the target TD's service TD binding table		
R9	SERVTD_TYPE: Service TD type		
R10	SERVTD_ATTR: Service TD attributes		

3.1.28.2. Output Operands

Table 3.118: TDH.SERVTD.REBIND Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
RCX	Binding handle In case of an error, as indicated by RAX, RCX returns 0.
R10	TD_UUID bits 63:0 In case of an error, as indicated by RAX, R10 returns 0.
R11	TD_UUID bits 127:64 In case of an error, as indicated by RAX, R11 returns 0.
R12	TD_UUID bits 191:128 In case of an error, as indicated by RAX, R12 returns 0.
R13	TD_UUID bits 255:192 In case of an error, as indicated by RAX, R13 returns 0.
AVX, AVX2 and AVX512 state	May be reset to the architectural RESET state

Operand	Description
Other	Unmodified

3.1.28.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.28.3.1. Overview

TDH.SERVTD.REBIND binds a new service TD to a target TD. Current bound service TD must approve the transition to the new service TD by calling the TDG.SERVTD.REBIND.APPROVE API prior to this call.

3.1.28.3.2. Enumeration

Availability of TDH.SERVTD.REBIND is enumerated by TDX_FEATURES0.SERVTD_REBIND (bit 48), readable by TDH.SYS.RD*. If not supported, calling TDH.SERVTD.REBIND returns a TDX_OPERAND_INVALID(RAX) status.

3.1.28.3.3. Preconditions

- The target TD has not been paused for export and is not in the in-order import phase.
- The target TD's measurements have been finalized (by TDH.MR.FINALIZE) or it is being imported, and import is in the out-of-order phase.

3.1.28.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.119: TDH.SERVTD.REBIND Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	Target TD's TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	RDX	HPA	Service TD's TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Implicit	N/A	N/A	Target TD's TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	Target TD's TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Binding table		RW	Opaque	N/A	Exclusive	None	None
Implicit	N/A	N/A	Service TD's TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	Service TD's TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Service TD's TDCS.RTMR	SHA384_ HASH	N/A	Opaque	N/A	Exclusive (h)	N/A	N/A

3.1.28.5. Completion Status Codes

Table 3.120: TDH.SERVTD.REBIND Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_REBIND_TOKEN_MISMATCH	Rebind session tokens did not match
TDX_SERVTD_ATTR_MISMATCH	
TDX_SERVTD_CANNOT_BE_MIGRATABLE	
TDX_SERVTD_INFO_HASH_MISMATCH	Rebind SERVTD_EXT hash did not match
TDX_SERVTD_NESTING_NOT_ALLOWED	
TDX_SERVTD_INCORRECT_BINDING_STATE	Target TD binding state is not in the expected state
TDX_SERVTD_INFO_HASH_MISMATCH	
TDX_SERVTD_TYPE_MISMATCH	
TDX_SERVTD_UUID_MISMATCH	
TDX_SUCCESS	TDH.SERVTD.REBIND is successful.
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.1.29. TDH.SERVTD.PREBIND Leaf

Pre-bind a service TD to a target TD.

3.1.29.1. Input Operands

Table 3.121: TDH.SERVTD.PREBIND Input Operands Definition

Operand	Description		
RAX	SEAMCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the SEAMCALL interface function: 49
	23:16	Version Number	Selects the SEAMCALL interface function version Must be 0
	24	INTERRUPT_MODE	Ignored
	62:25	Reserved	Must be 0
	63	P-SEAMLDR	Must be 0 for TDX Module interface functions
RCX	The physical address of the target TD's TDR page (HKID bits must be 0)		
RDX	The physical address (including HKID bits) of SERVTD_INFO_HASH, the expected SHA384 hash of the service TD's TDINFO_STRUCT		
R8	Index (slot number) in the target TD's service TD binding table		
R9	SERVTD_TYPE: Expected service TD type		
R10	SERVTD_ATTR: Expected service TD attributes		

3.1.29.2. Output Operands

Table 3.122: TDH.SERVTD.PREBIND Output Operands Definition

Operand	Description
RAX	SEAMCALL instruction return code
Other	Unmodified

3.1.29.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.1.29.3.1. Overview

TDH.SERVTD.PREBIND pre-binds a service TD to a target TD, by setting its expected binding parameters.

3.1.29.3.2. Enumeration

Availability of TDH.SERVTD.PREBIND is enumerated by TDX_FEATURES0.SERVICE_TD (bit 2), readable by TDH.SYS.RD*. If not supported, calling TDH.SERVTD.PREBIND returns a TDX_OPERAND_INVALID(RAX) status.

3.1.29.3.3. Preconditions

- The target TD has been created and the TDCS pages have been allocated.

- The target TD has not been finalized by TDH.MR.FINALIZE.
- The service TD has been finalized by TDH.MR.FINALIZE.

3.1.29.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.123: TDH.SERVTD.PREBIND Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	Target TD's TDR page	TDR	R	Opaque	4KB	Shared	Shared	Shared
Explicit	RDX	HPA	SERVTD_INFO_ HASH	SHA384_ HASH	R	Shared	64B	N/A	N/A	N/A
Implicit	N/A	N/A	Target TD's TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	N/A	N/A
Implicit	N/A	N/A	Target TD's TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Binding table		RW	Opaque	N/A	Exclusive	None	None

3.1.29.5. Completion Status Codes

Table 3.124: TDH.SERVTD.PREBIND Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_OP_STATE_INCORRECT	
TDX_OPERAND_ADDR_RANGE_ERROR	
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	
TDX_SERVTD_ALREADY_BOUND_FOR_TYPE	
TDX_SUCCESS	TDH.SERVTD.PREBIND is successful.
TDX_SYS_NOT_READY	
TDX_SYS_SHUTDOWN	
TDX_TD_FATAL	
TDX_TD_KEYS_NOT_CONFIGURED	
TDX_TDCS_NOT_ALLOCATED	

3.2. Guest-Side (TDCALL) Interface Functions

3.2.1. TDG.SERVTD.RD Leaf

As a service TD, read a metadata field (control structure field) of a target TD.

3.2.1.1. Input Operands

Table 3.125: TDG.SERVTD.RD Input Operands Definition

Operand	Description		
RAX	TDCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the TDCALL interface function: 18
	23:16	Version Number	Selects the TDCALL interface function version Must be 0
	63:24	Reserved	Must be 0
RCX	Binding handle		
RDX	Field identifier The LAST_ELEMENT_IN_FIELD and LAST_FIELD_IN_SEQUENCE components of the field identifier must be 0. WRITE_MASK_VALID, INC_SIZE, CONTEXT_CODE and ELEMENT_SIZE_CODE components of the field identifier are ignored. A value of -1 is a special case: it is not a valid field identifier; in this case the first readable field identifier is returned in RDX.		
R10	Target TD's TD_UUID bits 63:0		
R11	Target TD's TD_UUID bits 127:64		
R12	Target TD's TD_UUID bits 191:128		
R13	Target TD's TD_UUID bits 255:192		

3.2.1.2. Output Operands

Table 3.126: TDG.SERVTD.RD Output Operands Definition

Operand	Description
RAX	TDCALL instruction return code
RDX	RDX returns the next readable field identifier. A value of -1 indicates no next field identifier is available. If the input field identifier was -1, RDX returns the first readable field identifier. In case of another error, RDX returns -1. The ordering of field identifiers is discussed in the [ABI Spec].
R8	Contents of the field In case of an error, as indicated by RAX, R8 returns 0.
R10	Updated target TD's TD_UUID bits 63:0 – see the description below.

Operand	Description
R11	Updated target TD's TD_UUID bits 127:64 – see the description below.
R12	Updated target TD's TD_UUID bits 191:128 – see the description below.
R13	Updated target TD's TD_UUID bits 255:192 – see the description below.
Other	Unmodified

3.2.1.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.2.1.3.1. Overview

TDG.SERVTD.RD reads a metadata field (control structure field) of a target TD.

RDX returns the next host-side readable field identifier. This may be used by the Service TD to dump the target TD metadata readable by the Service TD. To read all the available fields, the service TD can invoke TDG.SERVTD.RD in a loop, starting with field identifier -1 as an input, until RDX returns -1. A status code of TDX_METADATA_FIELD_SKIP indicates that the returned value is not applicable.

3.2.1.3.2. Enumeration

Availability of TDG.SERVTD.RD is enumerated by TDX_FEATURES0.SERVICE_TD (bit 2), readable by TDG.SYS.RD*. If not supported, calling TDG.SERVTD.RD returns a TDX_OPERAND_INVALID(RAX) status.

3.2.1.3.3. TD_UUID Update

TD_UUID is updated when the target TD is imported. If the service TD binding to the target TD happened before the target TD was imported, the TD_UUID provided in R13:R10 may no longer be correct. In this case, if the TD_UUID provided in R13:R10 is equal to the pre-import TD_UUID of the target TD, TDG.SERVTD.RD returns TDX_TARGET_UUID_UPDATED status in RAX, and updates R13:R10 with the imported value of TD_UUID. The caller should retry the operation with the new TD_UUID.

3.2.1.3.4. Cross-TD Traps

Failure to access the metadata of the target TD may result in a cross-TD trap TD exit to the host VMM. This TD exit is trap like, meaning it happens after TDG.SERVTD.RD has completed its operation. On the following TDH.VP.ENTER, the host VMM may set a HOST_RECOVERABILITY_HINT flag, indicating that TDG.SERVTD.RD may be retried. From the guest TD's perspective, this flag appears in bit 60 of the status code returned in RAX. See the [TDX Module Base Spec] for details.

3.2.1.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.127 TDG.SERVTD.RD Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	Target TD's TDR page (from binding handle)	TDR	R	Opaque	N/A	Shared(h)	Shared(h)	Shared(h)
Implicit	N/A	N/A	Service (this) TD's TDR page	TDR	None	Opaque	N/A	Shared(i)	None	None

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	N/A	Service (this) TD's TDCS structure	TDCS	R	Opaque	N/A	Shared(i)	None	None
Implicit	N/A	N/A	Service (this) TD's TDCS.RTMR	SHA384_ HASH	N/A	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Service (this) TD's TDVPS structure	TDVPS	None	Opaque	N/A	Shared(i)	Shared(i)	Shared(i)
Implicit	N/A	N/A	Target TD's TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	None	None
Implicit	N/A	N/A	Target TD's TDCS.OP_STATE	OP_STATE	RW	Opaque	N/A	Shared(h)	N/A	N/A
Implicit	N/A	N/A	Target TD's Binding table		R	Opaque	N/A	Shared(h)	None	None
Implicit	N/A	N/A	Target TD's TD metadata	N/A	R	Opaque	N/A	None	None	None

3.2.1.5. Completion Status Codes

Table 3.128: TDG.SERVTD.RD Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_METADATA_FIELD_ID_INCORRECT	
TDX_METADATA_FIELD_NOT_READABLE	
TDX_METADATA_FIELD_SKIP	Indicates that the field value being read is not applicable and needs to be skipped. If called in a loop, use RDX as the identifier of the next field to be read, if any.
TDX_METADATA_FIRST_FIELD_ID_IN_CONTEXT	Indicates that the first field ID in context is returned
TDX_METADATA_FIELD_VALUE_NOT_VALID	
TDX_OP_STATE_INCORRECT	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_OPERAND_ADDR_RANGE_ERROR	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	

Completion Status Code	Description
TDX_PAGE_METADATA_INCORRECT	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SERVTD_CANNOT_BE_MIGRATABLE	
TDX_SERVTD_INFO_HASH_MISMATCH	This service TD's info hash doesn't match the service TD info hash in the target TD's binding information. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SERVTD_NESTING_NOT_ALLOWED	
TDX_SERVTD_NOT_BOUND	This service TD is not bound to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SERVTD_UUID_MISMATCH	This service TD's TD_UUID doesn't match the service TD UUID in the target TD's binding information. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SUCCESS	TDG.SERVTD.RD is successful.
TDX_TARGET_UUID_MISMATCH	The target TD's TD_UUID value provided in R13:R10 doesn't match the actual value.
TDX_TARGET_UUID_UPDATED	The target TD's TD_UUID value provided in R13:R10 doesn't match the current actual value, but it does match the TD_UUID that target TD had before it was imported. In this case, the current TD_UUID value is provided in R13:R10, and the operation can be retried.
TDX_TD_FATAL	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_TD_KEYS_NOT_CONFIGURED	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_TDCS_NOT_ALLOCATED	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.

3.2.2. TDG.SERVTD.REBIND.APPROVE

The TDG.SERVTD.REBIND.APPROVE API is called by the currently bound service TD to approve a new Service TD to be bound to the target TD.

3.2.2.1. Input Operands

Table 3.129: TDG.SERVTD.REBIND.APPROVE Input Operand Definition

Operand	Description		
RAX	TDCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the TDCALL interface function: 33
	23:16	Version Number	Selects the TDCALL interface function version Must be 0
	63:24	Reserved	Must be 0
RCX	Old Binding Handle		
RDX	Rebind Session Token [63:0]		
R8	Rebind Session Token [127:64]		
R9	Rebind Session Token [195:128]		
R10	Target TD's TD_UUID bits 63:0		
R11	Target TD's TD_UUID bits 127:64		
R12	Target TD's TD_UUID bits 191:128		
R13	Target TD's TD_UUID bits 255:192		
R14	Rebind Session Token [255:196]		

3.2.2.2. Output Operands

Table 3.130: TDG.SERVTD.REBIND.APPROVE Output Operand Definition

Operand	Description
RAX	TDCALL instruction return code – see 3.2.2.5
R10	Updated target TD's TD_UUID bits 63:0 – see the description below.
R11	Updated target TD's TD_UUID bits 127:64 – see the description below.
R12	Updated target TD's TD_UUID bits 191:128 – see the description below.
R13	Updated target TD's TD_UUID bits 255:192 – see the description below.
Other	Unmodified

3.2.2.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.2.2.3.1. Overview

TDG.SERVTD.REBIND.APPROVE allows the calling Service TD to approve a new Service TD to be bound to the target TD currently bound to it. It sets the target TD's TDCS.REBIND_SESSION_TOKENS[slot] to the provided input rebind session token, where the slot is the current service TD's binding slot.

For Service TD rebinding details, see the [Base Spec].

3.2.2.3.2. Enumeration

Availability of TDG.SERVTD.REBIND.APPROVE is enumerated by TDX_FEATURES0.SERVTD_REBIND (bit 48), readable by TDG.SYS.RD*. If not supported, calling TDG.SERVTD.REBIND.APPROVE returns a TDX_OPERAND_INVALID(RAX) status.

3.2.2.3.3. TD_UUID Update

TD_UUID is updated when the target TD is imported. If the service TD binding to the target TD happened before the target TD was imported, the TD_UUID provided in R13:R10 may no longer be correct. In this case, if the TD_UUID provided in R13:R10 is equal to the pre-import TD_UUID of the target TD, TDG.SERVTD.REBIND.APPROVE returns TDX_TARGET_UUID_UPDATED status in RAX, and updates R13:R10 with the imported value of TD_UUID. The caller should retry the operation with the new TD_UUID.

3.2.2.3.4. Cross-TD Traps

Failure to access the metadata of the target TD may result in a cross-TD trap TD exit to the host VMM. This TD exit is trap like, meaning it happens after TDG.SERVTD.RD has completed its operation. On the following TDH.VP.ENTER, the host VMM may set a HOST_RECOVERABILITY_HINT flag, indicating that TDG.SERVTD.REBIND.APPROVE may be retried. From the guest TD's perspective, this flag appears in bit 60 of the status code returned in RAX. See the [TDX Module Base Spec] for details.

3.2.2.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.131: TDG.SERVTD.REBIND.APPROVE Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	Target TD's TDR page (from binding handle)	TDR	RW	Opaque	N/A	Shared(h)	Shared(h)	Shared(h)
Implicit	N/A	N/A	Service (this) TD's TDR page	TDR	None	Opaque	N/A	Shared(i)	None	None
Implicit	N/A	N/A	Service (this) TD's TDCS structure	TDCS	R	Opaque	N/A	Shared(i)	None	None
Implicit	N/A	N/A	Service (this) TD's TDCS.RTMR	SHA384_ HASH	N/A	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Service (this) TD's TDVPS structure	TDVPS	None	Opaque	N/A	Shared(i)	Shared(i)	Shared(i)
Implicit	N/A	N/A	Target TD's TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	None	None

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	N/A	Service (this) TD's TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared(h)	N/A	N/A
Implicit	N/A	N/A	Target TD's Binding table		R	Opaque	N/A	Exclusive(h)	None	None
Implicit	N/A	N/A	Target TD's TD metadata	N/A	RW	Opaque	N/A	None	None	None

3.2.2.5. Completion Status Codes

Table 3.132: TDG.SERVTD.REBIND.APPROVE Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_SUCCESS	TDG.SERVTD.REBIND.APPROVE is successful.
TDX_OP_STATE_INCORRECT	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_OPERAND_ADDR_RANGE_ERROR	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	
TDX_PAGE_METADATA_INCORRECT	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SERVTD_CANNOT_BE_MIGRATABLE	
TDX_SERVTD_INFO_HASH_MISMATCH	This service TD's info hash doesn't match the service TD info hash in the target TD's binding information. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SERVTD_NESTING_NOT_ALLOWED	

Completion Status Code	Description
TDX_SERVTD_NOT_BOUND	This service TD is not bound to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SERVTD_UUID_MISMATCH	This service TD's TD_UUID doesn't match the service TD UUID in the target TD's binding information. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SUCCESS	TDG.SERVTD.REBIND.APPROVE is successful.
TDX_TARGET_UUID_MISMATCH	The target TD's TD_UUID value provided in R13:R10 doesn't match the actual value.
TDX_TARGET_UUID_UPDATED	The target TD's TD_UUID value provided in R13:R10 doesn't match the current actual value, but it does match the TD_UUID that target TD had before it was imported. In this case, the current TD_UUID value is provided in R13:R10, and the operation can be retried.
TDX_TD_FATAL	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.

3.2.3. TDG.SERVTD.WR Leaf

As a service TD, write a metadata field (control structure field) of a target TD.

3.2.3.1. Input Operands

Table 3.133: TDG.SERVTD.WR Input Operands Definition

Operand	Description		
RAX	TDCALL instruction leaf number and version		
	Bits	**Field**	**Description**
	15:0	Leaf Number	Selects the TDCALL interface function: 19
	23:16	Version Number	Selects the TDCALL interface function version Must be 0
	63:24	Reserved	Must be 0
RCX	Binding handle		
RDX	Field identifier The LAST_ELEMENT_IN_FIELD and LAST_FIELD_IN_SEQUENCE components of the field identifier must be 0. WRITE_MASK_VALID, INC_SIZE, CONTEXT_CODE and ELEMENT_SIZE_CODE components of the field identifier are ignored. A value of -1 is a special case: it is not a valid field identifier; in this case the first readable field identifier is returned in RDX.		
R8	Data to write to the field		
R9	A 64b write mask to indicate which bits of the value in R8 are to be written to the field		
R10	Target TD's TD_UUID bits 63:0		
R11	Target TD's TD_UUID bits 127:64		
R12	Target TD's TD_UUID bits 191:128		
R13	Target TD's TD_UUID bits 255:192		

3.2.3.2. Output Operands

Table 3.134: TDG.SERVTD.WR Output Operands Definition

Operand	Description
RAX	TDCALL instruction return code
R8	Previous contents of the field In case of an error, R8 returns 0.
R10	Updated target TD's TD_UUID bits 63:0 – see the description below.
R11	Updated target TD's TD_UUID bits 127:64 – see the description below.
R12	Updated target TD's TD_UUID bits 191:128 – see the description below.

Operand	Description
R13	Updated target TD's TD_UUID bits 255:192 – see the description below.
Other	Unmodified

3.2.3.3. Leaf Function Description

Note: The description below is provided at a high level. Actual details, order of checks, returned status codes, etc. may vary.

3.2.3.3.1. Overview

TDG.SERVTD.WR writes a metadata field (control structure field) of a target TD. The value (R8) is written as specified by the write mask (R9). Writing is subject to the field's internal write mask (per the TD's ATTRIBUTES.DEBUG bit). Writing of specific fields is also subject to additional rules.

Table 3.135: Metadata Field Write Rules

Write Mask Bit in R9	Internal Write Mask Bit	Value Bit in R8
0	N/A	Silently ignored
1	0	Must be the same as the current field's bit
1	1	Written to the current field's bit

3.2.3.3.2. Enumeration

Availability of TDG.SERVTD.WR is enumerated by TDX_FEATURES0.SERVICE_TD (bit 2), readable by TDG.SYS.RD*. If not supported, calling TDG.SERVTD.WR returns a TDX_OPERAND_INVALID(RAX) status.

3.2.3.3.3. TD_UUID Update

TD_UUID is updated when the target TD is imported. If the service TD binding to the target TD happened before the target TD was imported, the TD_UUID provided in R13:R10 may no longer be correct. In this case, if the TD_UUID provided in R13:R10 is equal to the pre-import TD_UUID of the target TD, TDG.SERVTD.WR returns TDX_TARGET_UUID_UPDATED status in RAX, and updates R13:R10 with the imported value of TD_UUID. The caller should retry the operation with the new TD_UUID.

3.2.3.3.4. Cross-TD Traps

Failure to access the metadata of the target TD may result in a cross-TD trap TD exit to the host VMM. This TD exit is trap like, meaning it happens after TDG.SERVTD.WR has completed its operation. On the following TDH.VP.ENTER, the host VMM may set a HOST_RECOVERABILITY_HINT flag, indicating that TDG.SERVTD.WR may be retried. From the guest TD's perspective, this flag appears in bit 60 of the status code returned in RAX. See the [TDX Module Base Spec] for details.

3.2.3.4. Operands Information

To understand the table and text below, please refer to the [TDX Module Base Spec] chapter discussing general aspects of the Intel TDX Module API.

Table 3.136 TDG.SERVTD.WR Operands Information Definition

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Explicit	RCX	HPA	Target TD's TDR page (from binding handle)	TDR	RW	Opaque	N/A	Shared(h)	Shared(h)	Shared(h)

Explicit/ Implicit	Reg.	Ref Type	Resource	Resource Type	Access	Access Semantics	Align Check	Concurrency Restrictions		
								Operand	Contain. 2MB	Contain. 1GB
Implicit	N/A	N/A	Service (this) TD's TDR page	TDR	None	Opaque	N/A	Shared(i)	None	None
Implicit	N/A	N/A	Service (this) TD's TDCS structure	TDCS	R	Opaque	N/A	Shared(i)	None	None
Implicit	N/A	N/A	Service (this) TD's TDCS.RTMR	SHA384_ HASH	N/A	Opaque	N/A	Shared	N/A	N/A
Implicit	N/A	N/A	Service (this) TD's TDVPS structure	TDVPS	None	Opaque	N/A	Shared(i)	Shared(i)	Shared(i)
Implicit	N/A	N/A	Target TD's TDCS structure	TDCS	RW	Opaque	N/A	Shared(i)	None	None
Implicit	N/A	N/A	Service (this) TD's TDCS.OP_STATE	OP_STATE	R	Opaque	N/A	Shared(h)	N/A	N/A
Implicit	N/A	N/A	Target TD's Binding table		R	Opaque	N/A	Shared(h)	None	None
Implicit	N/A	N/A	Target TD's TD metadata	N/A	RW	Opaque	N/A	None	None	None

3.2.3.5. Completion Status Codes

Table 3.137: TDG.SERVTD.WR Completion Status Codes (Returned in RAX) Definition

Completion Status Code	Description
TDX_METADATA_FIELD_ID_INCORRECT	
TDX_METADATA_FIELD_NOT_READABLE	
TDX_METADATA_FIELD_NOT_WRITABLE	
TDX_METADATA_FIELD_VALUE_NOT_VALID	
TDX_METADATA_WR_MASK_NOT_VALID	
TDX_OP_STATE_INCORRECT	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_OPERAND_ADDR_RANGE_ERROR	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_OPERAND_BUSY	Operation encountered a busy operand, indicated by the lower 32 bits of the status. In many cases, this can be resolved by retrying the operation.
TDX_OPERAND_INVALID	

Completion Status Code	Description
TDX_PAGE_METADATA_INCORRECT	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SERVTD_CANNOT_BE_MIGRATABLE	
TDX_SERVTD_INFO_HASH_MISMATCH	This service TD's info hash doesn't match the service TD info hash in the target TD's binding information. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SERVTD_NESTING_NOT_ALLOWED	
TDX_SERVTD_NOT_BOUND	This service TD is not bound to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SERVTD_UUID_MISMATCH	This service TD's TD_UUID doesn't match the service TD UUID in the target TD's binding information. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_SUCCESS	TDG.SERVTD.WR is successful.
TDX_TARGET_UUID_MISMATCH	The target TD's TD_UUID value provided in R13:R10 doesn't match the actual value.
TDX_TARGET_UUID_UPDATED	The target TD's TD_UUID value provided in R13:R10 doesn't match the current actual value, but it does match the TD_UUID that target TD had before it was imported. In this case, the current TD_UUID value is provided in R13:R10, and the operation can be retried.
TDX_TD_FATAL	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_TD_KEYS_NOT_CONFIGURED	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.
TDX_TDCS_NOT_ALLOCATED	This status code refers to the target TD. Bit 60 (HOST_RECOVERABILITY_HINT) contains a hint from the host VMM that the error condition has been resolved, and the service TD can retry the operation.