AMD64 Zen6 Instruction Based Sampling (IBS) Extensions and Features

Publication # **69205** Revision: **1.00**

Issue Date: **March 2026**

Revision History

Date	Revision	Change Description
March 2026	1.00	Initial release

Zen6 Instruction Based Sampling (IBS) Extensions and Features

This document describes the following Instruction Based Sampling (IBS) features and extensions available on some Zen6 products:

- General extensions for Fetch and Execution IBS
- New IBS Filtering capabilities
- IBS Buffering
- IBS Memory Profiler

Instruction Based Sampling Extensions

This section describes the following general extensions to Instruction Based Sampling (IBS).

- Alternate Fetch and Execution IBS disable bits
- IBS Filtering support for fetch latency
- IBS Filtering support for virtual address bit 63 for Fetch and Execution IBS
- Streaming (non-temporal) store and remote socket indication for Execution IBS

Alternate Fetch and Execution IBS Disable Bits

Processors which set CPUID Fn8000_0001B_EAX[IbsDis] (bit 13) = 1 support alternate Fetch and Execution IBS disable bits. For Fetch IBS the alternate disable bit is bit 0 of IBS Fetch Control Register 2 and for Execution IBS it is bit 0 of IBS Execution Control Register 2. These bits allow software to start and stop IBS via writes to registers not updated by the processor thus eliminating the read-modify-write hazard which exists for modifying bits in IBS Fetch Control Register and IBS Execution Control Register, respectively, while IBS is active.

IBS Filtering Extensions

Zen6 adds the following new IBS Filter capabilities:

- Fetch IBS Filtering for fetch latency. Controlled by IbsFetchCtl2[IbsFetchLatFilter] and supported when CPUID Fn8000_0001B_EAX[IbsFetchLatencyFiltering] (bit 14) = 1
- IBS Filtering for instruction virtual address bit 63 being set or cleared. Controlled by IbsFetchCtl2[IbsFetchExclAddr63Eq0] and IbsFetchCtl2[IbsFetchExclAddr63Eq1] for IBS Fetch Sampling, IbsOpCtl2[IbsOpExclAddr63Eq0] and IbsOpCtl2[IbsOpExclAddr63Eq1] for IBS Execution Sampling and IbsMemCtl2[IbsMemExclAddr63Eq0] and IbsMemCtl2[IbsMemExclAddr63Eq1] for IBS Memory Profiler. Support is indicated by CPUID Fn8000_0001B_EAX[IbsAddrBit63Filtering] (bit 15) = 1.
- IBS Filtering for streaming stores. Controlled by IbsOpCtl2[IbsOpStrmStFilter] for IBS Execution Sampling and IbsMemCtl2[IbsMemStrmStFilter] for IBS Memory Profiler. Support is indicated by CPUID Fn8000_0001B_EAX[IbsStrmStAndRmtSocket] (bit 16) = 1.

The following pseudocode illustrates the processor's Fetch and Execution IBS Filtering behavior considering IBS Filter controls available on some processors.

The action `restart IBS` refers to the processor discarding a sample and restarting the respective IBS facility as described in "*AMD64 Architecture Programmer's Manual*", Volume 2, Section 13.3.5 *"IBS Filtering"*. The action `report valid IBS sample` refers to the processor setting the respective IBS sample valid (IbsFetchCtl[IbsFetchVal], IbsOpCtl[IbsOpVal] or IbsMemCtl[IbsMemVal]) and signaling an IBS interrupt. CPUID checks for the support of IBS filtering capabilities have been omitted for clarity.

See a description of the filter control bits in the "Registers for Zen6 Instruction Based Sampling Extensions" section below for CPUID enumeration of individual filter capabilities.

Fetch IBS Filtering Pseudocode

```
// fetch address bit 63 filtering
IF (   IbsFetchCtl2[IbsFetchExclAddr63Eq0]==1 &&
       IbsFetchLinAd[63]==0)

       restart IBS

// fetch address bit 63 filtering
ELSIF ( IbsFetchCtl2[IbsFetchExclAddr63Eq1]==1 &&
       IbsFetchLinAd[63]==1)

       restart IBS

// fetch latency filtering
ELSIF ( IbsFetchCtl[IbsFetchLat] < IbsFetchCtl2[IbsFetchLatFilter]*128 )

       restart IBS

// no L3-miss filtering enabled
elsif (IbsFetchCtl[IbsL3MissOnly]==0)

       report valid IBS sample

// L3-miss filtering
elsif (IbsFetchCtl[IbsL3MissOnly]==1 &&
       IbsFetchCtl[IbsFetchL3Miss]==1

       report valid IBS sample

ELSE
       restart IBS
```

Execution and Memory Profiler IBS Filtering Pseudocode

```
IF (   IbsOpCtl2[IbsOpExclRip63Eq0]==1 &&
       (IbsOpRip[63]==0 || IbsOpData1[IbsRipInvalid]==1))

       restart IBS

ELSIF (IbsOpCtl2[IbsOpExclRip63Eq1]==1 &&
       (IbsOpRip[63]==1 || IbsOpData1[IbsRipInvalid]==1))

       restart IBS
```

```
// no load latency, L3-miss or streaming store filtering is active
ELSIF (   IbsOpCtl[IbsOpLatFltEn]==0 &&
          IbsOpCtl1[IbsOpL3MissOnly]==0 &&
          IbsOpCtl2[IbsOpStrmStFilter]==0)

       report valid IBS sample

// load latency filtering without L3 miss filtering
ELSIF (   IbsOpCtl[IbsOpLatFltEn]==1 &&
          IbsOpCtl[IbsOpL3MissOnly]==0 &&
          IbsOpData3[IbsLdOp]==1 &&
          IbsOpData3[IbsSwPf]==0 &&
          (IbsOpData3[IbsDcMissLat] > (IbsOpData3[IbsOpLatThrsh]+1) * 128)

       report valid IBS sample

// load latency filtering with L3 miss filtering
ELSIF (   IbsOpCtl[IbsOpLatFltEn]==1 &&
          IbsOpCtl[IbsOpL3MissOnly]==1 &&
          IbsOpData3[IbsL2Miss]==1 &&
          IbsOpData2[DataSrc[4:0]] > 01h) &&
          IbsOpData3[IbsLdOp]==1 &&
          IbsOpData3[IbsSwPf]==0 &&
          (IbsOpData3[IbsDcMissLat] > (IbsOpCtl[IbsOpLatThrsh]+1) * 128)

       report valid IBS sample

// L3-miss filtering without load latency filtering
ELSIF (   IbsOpCtl[IbsOpLatFltEn]==0 &&
          IbsOpCtl[IbsOpL3MissOnly]==1 &&
          IbsOpData3[IbsL2Miss]==1 &&
          IbsOpData2[DataSrc[4:0]] > 01h)

       report valid IBS sample

// streaming store filtering
ELSIF (   IbsOpCtl2[IbsOpStrmStFilter]==1 &&
          IbsOpData2[StrmSt]==1)

       report valid IBS sample

ELSE

       restart IBS
```

Registers for Zen6 Instruction Based Sampling Extensions

The following is a list of new registers and register bits or fields added to existing registers for the Instruction Based Sampling extensions described in this document.

IBS Fetch Control Register 2 (IbsFetchCtl2) (MSR C001_103Fh)

Bits	Mnemonic	Description	Access Type
63:7	Reserved		RAZ
6	IbsFetchExclAddr63Eq0	Exclude IBS Fetch samples with fetch address bit 63 cleared	R/W
5	IbsFetchExclAddr63Eq1	Exclude IBS Fetch samples with fetch address bit 63 set	R/W
4:1	IbsFetchLatFilter	IBS Fetch Latency filter threshold	R/W
0	IbsFetchDis	IBS Fetch Sampling Disable	R/W

The fields of IbsFetchCtl2 are further described below:

- *IbsFetchExclAddr63Eq0 (Exclude IBS Fetch samples with fetch address bit 63 cleared)* – Bit 6, read/write. This bit controls IBS Fetch Filtering. When set, Fetch IBS excludes samples when the logical fetch address (IbsFetchLinAd) has bit 63 cleared. IbsFetchExclAddr63Eq0 is supported when CPUID Fn8000_0001B_EAX[IbsAddrBit63Filtering] (bit 15) = 1.
- *IbsFetchExclAddr63Eq1 (Exclude IBS Fetch samples with fetch address bit 63 set)* – Bit 5, read/write. This bit controls IBS Fetch Filtering. When set, Fetch IBS excludes samples when the logical fetch address (IbsFetchLinAd) has bit 63 set. IbsFetchExclAddr63Eq1 is supported when CPUID Fn8000_0001B_EAX[IbsAddrBit63Filtering] (bit 15) = 1.
- *IbsFetchLatFilter (IBS Fetch Latency filter threshold)* – Bits 4:1, read/write. This bit controls IBS Fetch Filtering. When programmed with a non-zero value, Fetch IBS only reports samples when the fetch latency (IbsFetchCtl[IbsFetchLat]) is greater or equal to this field (IbsFetchLatFilter) times 128. When programmed with a non-zero value and IBS L3 Miss Filtering is active IbsFetchCtl[IbsL3MissOnly] = 1), the Fetch IBS sample must be an L3 miss IbsFetchCtl[IbsL3Miss] = 1) and the fetch latency threshold condition must be met for Fetch IBS to report a sample. This field is supported when CPUID Fn8000_0001B_EAX[IbsFetchLatencyFiltering] (bit 14) = 1
- *IbsFetchDis (IBS Fetch Sampling Disable)* – Bit 0, read/write. When set, IBS Fetch Sampling is disabled. IBS Fetch Sampling is only enabled if IbsFetchDis = 0 (if supported), IbsFetchCtl[IbsFetchEn] = 1, and IbsFetchCtl[IbsFetchVal] = 0. This bit is supported when CPUID Fn8000_0001B_EAX[IbsDis] (bit 13) = 1.

IBS Fetch Control Extended Register (IbsFetchCtlExt) (MSR C001_103Ch)

Support for IbsFetchCtlExt is indicated by CPUID Fn8000_0001B_EAX[IbsFetchCtlExt] (bit 9) = 1.

Bits	Mnemonic	Description	Access Type
63:16	Reserved		RAZ
15:0	IbsFetchItlbRefillLat	IBS Fetch ITLB Refill Latency	R/W

The fields of IbsFetchCtlExt are further described below

- *IbsFetchItlbRefillLat (IBS Fetch ITLB Refill Latency)* – Bits 15:0, read/write. This 16-bit field indicates the number of core clock cycles the fetch engine is stalled for an ITLB reload for the sampled fetch. If there is no reload, the reported latency is zero.

IBS Execution Control Register 2 (IbsOpCtl2) (MSR C001_103Eh)

Bits	Mnemonic	Description	Access Type
63:4	Reserved		RAZ
3	IbsOpStrmStFilter	IBS Op Streaming Store Filter Control	R/W
2	IbsOpExclRip63Eq1	Exclude IBS Op samples with RIP bit 63 set	R/W
1	IbsOpExclRip63Eq0	Exclude IBS Op samples with RIP bit 63 cleared	R/W
0	IbsOpDis	IBS Execution Sampling disable	R/W

The fields of IbsOpCtl2 are further described below:

- *IbsOpStrmStFilter (IBS Op Streaming Store Filter Control)* – Bit 3, read/write. This bit controls Execution IBS Filtering. When set, samples for streaming stores (IbsOpData2[StrmSt] = 1) are reported. For more detail on Zen6 IBS Filtering see section “IBS Filtering Extensions”
- *IbsOpExclRip63Eq1 (Exclude IBS Execution samples with RIP bit 63 set)* – Bit 2, read/write. This bit controls IBS Execution Filtering. When set, Execution IBS excludes samples when the logical instruction address (IbsOpRip) has bit 63 set or IbsOpRip is not valid (IbsOpData1[IbsRipInvalid] = 1). IbsOpExclRip63Eq1 is supported when CPUID Fn8000_0001B_EAX[IbsAddrBit63Filtering] (bit 15) = 1.
- *IbsOpExclRip63Eq0 (Exclude IBS Execution samples with RIP bit 63 cleared)* – Bit 1, read/write. This bit controls IBS Execution Filtering. When set, Execution IBS excludes samples when the logical instruction address (IbsOpRip) has bit 63 cleared or IbsOpRip is not valid (IbsOpData1[IbsRipInvalid] = 1). IbsOpExclRip63Eq0 is supported when CPUID Fn8000_0001B_EAX[IbsAddrBit63Filtering] (bit 15) = 1.
- *IbsOpDis (IBS Execution Sampling Disable)* – Bit 0, read/write. When set, IBS Execution Sampling is disabled. IBS Execution Sampling is only enabled if IbsOpDis = 0 (if supported), IbsOpCtl[IbsOpEn] = 1, and IbsOpCtl[IbsOpVal] = 0. This bit is supported when CPUID Fn8000_0001B_EAX[IbsDis] (bit 13) = 1.

IBS Op Data 2 (IbsOpData2) (MSR C001_1036h)

Only bits added with Zen6 IBS extensions are shown.

Bits	Mnemonic	Description	Access Type
9	RmtSocket	Remote Socket	R/W
8	StrmSt	Streaming Store	R/W

The new fields of IbsOpData2 are further described below.

- *StrmSt (Streaming Store)* – Bit 8, read/write. This bit is set if the tagged store operation was a streaming store. A streaming store is the result of a store instruction with a non-temporal hint (MOVNTI for example) or a store operation treated as non-temporal by the processor (for example stores in sufficiently long REP MOVSB operations). Streaming stores do not update other bits or fields in IbsOpData2. This bit is supported when CPUID Fn8000_0001B_EAX[IbsStrmStAndRmtSocket] (bit 16) = 1.

- *RmtSocket (Remote Socket)* – Bit 9, read/write. This bit is set if the data request for the tagged load or store operation was serviced by a different socket. This bit is supported when CPUID Fn8000_0001B_EAX[IbsStrmStAndRmtSocket] (bit 16) = 1.

Virtualization Support for Zen6 Instruction Based Sampling Extensions

Hardware support for virtualization of the Zen6 IBS Extensions is supported when IBS Virtualization is enabled by setting bit 2 at offset B8h in the VMCB Control Area.

The following table shows the location of the IBS Fetch Control Register 2 and IBS Execution Control Register 2 in the VMCB State Save Area.

Offset	Size	Contents	Notes
7C8h	qword	IbsFetchCtl2	IBS Virtualization state (swap type C)
7D0h	qword	IbsOpCtl2	

Instruction Based Sampling Buffering

Instruction Based Sampling (IBS) Buffering is a feature which reduces the software overhead for processing IBS sample interrupts. With IBS Buffering the processor suppresses interrupts for Fetch and Execution IBS and instead writes the samples (IBS register values) into a software specified memory buffer. Support for IBS Buffering Version 1 is indicated by CPUID Fn8000_001B_EAX[IbsBufferV1] (bit 17) = 1. IBS Buffering is not supported for the IBS Memory Profiler.

Enabling and Using IBS Buffering

To use IBS Buffering, software configures the size of the IBS memory buffer in IbsBufferSize[IbsBufferSize], writes bits 63:12 of the IBS Buffer linear starting address into IbsBufferBase[IbsBufferLinAddr] and sets IbsBufferBase[IbsBufferEn]=1. The processor assumes that the buffer start address is aligned on a 4-Kbyte boundary with the low-order 12 address bits 11:0 assumed to be 0. For IBS buffering to be active, IbsBufferFail and IbsBufferFull bits in the IBS Buffer Tail register need to be cleared.

Once IBS Buffering is enabled and active, the processor can write Fetch and Execution IBS samples into the IBS Buffer. Enabling IBS Buffering does not automatically enable Fetch or Execution IBS. Software needs to configure and enable these features individually. When a new Fetch or Execution IBS sample is available and IBS Buffering is active the processor does not send an IBS interrupt but instead attempts to write the IBS sample (MSR values of the respective IBS facility) into the IBS Buffer. When IbsBufferBase[TscRecordEn]=1, the processor also writes a TSC record before it writes a Fetch or Execution IBS record or both. It is possible for the processor to write both Fetch and Execution IBS sample if they become ready sufficiently close enough in time. The format of Fetch IBS, Execution IBS and TSC records is described in section *"IBS Buffer Records"* below.

Once the buffer write succeeds, the processor advances IbsBufferTail by the number of quadwords written to the IBS Buffer. If the remaining buffer capacity (IbsBufferSize[IbsBufferSize] – IbsBufferTail[IbsBufferTail]) is insufficient to write the maximum number of records that could be written at any time, assuming both Fetch and Execution IBS are enabled, the processor sets IbsBufferTail[IbsBufferFull] = 1 and signals an IBS interrupt as specified by the local APIC. The processor may also set IbsFetchCtl[IbsFetchVal]=1 and IbsOpCtl[IbsIOpVal]=1 to temporarily halt IBS.

When servicing an IBS interrupt with IBS buffering enabled, software should not interpret IbsFetchCtl[IbsFetchVal] = 1 or IbsOpCtl[IbsOpVal] = 1 as valid samples in the Fetch and Execution IBS register but instead check IbsBufferTail[IbsBufferFull] and IbsBufferTail[IbsBufferFail].

When IbsBufferTail[IbsBufferFull] is set the processor successfully wrote all IBS sample information into the memory buffer and software should empty the IBS buffer followed by writing IbsBufferTail[IbsBufferTail] to zero and clearing IbsBufferTail[IbsBufferFull]. When IbsBufferTail[IbsBufferFull] is cleared, the processor clears IbsFetchCtl[IbsFetchVal] and IbsOpCtl[IbsIOpVal], if set, to resume temporarily halted IBS activity.

Failed IBS Buffer Writes

Due to a non-canonical buffer address, page fault, nested page fault and other reasons it is possible for an IBS buffer write to fail. When an IBS buffer write fails, the processor sets IbsBufferTail[IbsBufferFail] indicating that IBS sample data has not been written and is available in the Fetch IBS and/or Execution IBS registers as

indicated by their respective IbsVal bits. When setting IbsBufferTail[IbsBufferFail] the processor signals an interrupt as specified by the local APIC. On a failed buffer write, the processor does not set IbsBufferTail[IbsBufferFull]

With IBS Buffering enabled, software should only check IbsFetchCtl[IbsFetchVal] and IbsOpCtl[IbsIOpVal] when IbsBufferTail[IbsBufferFail] is set.

Recommended IBS Interrupt Handling Pseudo-Code with IBS Buffering

The following pseudo-code shows the recommended software flow when handling IBS interrupts with IBS Buffering enabled.

```
IF IbsBufferTail[IbsBufferFull] == 1
       process IBS buffer
       write IbsBufferTail[IbsBufferTail] = 0
       write IbsBufferTail[IbsBufferFull] = 0
ELSIF IbsBufferTail[IbsBufferFail] == 1
       Disable Fetch IBS and Execution IBS
       IF IbsBufferTail[IbsBufferTail] > 0
          process IBS buffer
          write IbsBufferTail[IbsBufferTail] = 0
       IF IbsFetchCtl[IbsFetchVal] == 1
          Process Fetch IBS sample
          Write IbsFetchCtl[IbsFetchVal] = 0
       IF IbsOpCtl[IbsOpVal] == 1
          Process Execution IBS sample
          Write IbsOpCtl[IbsOpVal] = 0
       write IbsBufferTail[IbsBufferFail] = 0
       Reenable Fetch IBS and Execution IBS
ELSE
       // No Fetch or Execution IBS data available for consumption
```

IBS Buffering Registers

The IBS Buffering Version 1 registers consists of the IBS Buffer Base register (IbsBufferBase) which specifies the location of the IBS Buffer in memory and hosts IBS Buffering control bits, the IBS Buffer Size (IbsBufferSize) register which specifies the size of the IBS Buffer and the IBS Buffer Tail register which contains the current IBS Buffer tail pointer and IBS Buffering status bits. These registers can be accessed with WRMSR and RDMSR instructions.

IBS Buffer Base (IbsBufferBase) (MSR C001_0390h)

Bits	Mnemonic	Description	Access Type
63:12	IbsBufferLinAddr	IBS Buffer linear starting address	R/W
11:2	Reserved		MBZ
1	TscRecordEn	TSC Record Enable	R/W
0	IbsBufferEn	IBS Buffering Enable	R/W

The fields of IbsBufferBase are further described below:

- *IbsBufferLinAddr (IBS Buffer linear starting address*) – Bits 63:12, read/write. This field contains bits 63:12 of the IBS Buffer linear starting address, The processor assumes that the IBS Buffer is aligned at a 4K boundary and that bits 11:0 of its linear starting address are zero. The address must be canonical relative to the processor's maximum linear address width otherwise the write result in a general-protection exception (#GP).
- *TscRecordEn (TSC Record Enable)* – Bit 1, read/write. When set, the processor writes a TSC (Time Stamp Counter) record into the IBS buffer before it writes a Fetch and/or Execution IBS sample.
- *IbsBufferEn (IBS Buffering Enable)* – Bit 0, read/write. When set, IBS Buffering is enabled. See section *"Enabling and Using IBS Buffering"* for a description of how to configure and use IBS Buffering.

IBS Buffer Size (IbsBufferSize) (MSR C001_0391h)

Bits	Mnemonic	Description	Access Type
63:20	Reserved		MBZ
19:3	IbsBufferSize	IBS Buffer Size	R/W
2:0	Reserved		MBZ

The fields of IbsBufferSize are described below.

- *IbsBufferSize (IBS Buffer Size)* – bits 19:3, read/write. This field specifies the size of the IBS memory buffer in quadwords.

IBS Buffer Tail (IbsBufferTail) (MSR C001_0392h)

Bits	Mnemonic	Description	Access Type
63:20	Reserved		MBZ
19:3	IbsBufferTail	IBS Buffer tail pointer	R/W
2	Reserved		MBZ
1	IbsBufferFail	IBS Buffer write fail indication	R/W
0	IbsBufferFull	IBS Buffer full indication	R/W

The fields of IbsBufferTail are described below:

- *IbsBufferTail (IBS Buffer tail pointer)* – Bits 19:3, read/write. This field returns the current value of the IBS Buffer tail pointer which points to the quadword within the IBS Buffer that would be written next by the processor. After processing a full IBS Buffer, software needs to write this field to zero to empty the IBS buffer and restart IBS Buffering.
- *IbsBufferFail (Ibs Buffer write fail indication)* – Bit 1. The processor sets this bit when the write of IBS sample data into the IBS Buffer failed. See section *"Failed IBS Buffer Writes"* for more detail.
- *IbsBufferFull (IBS Buffer full indication)* – Bit 0. The processor sets this bit after it has written one or multiple IBS samples and determined that the remaining IBS buffer capacity

(IbsBufferSize[IbsBufferSize] – IbsBufferTail[IbsBufferTail]) is not sufficient for the most IBS sample information that might be written next.

IBS Buffer Records

IBS Buffer records are always multiple of 8 bytes (quadword). The first quadword contains a header which specifies the type of IBS record and its size in bytes. Subsequent bytes contain IBS sample information as read from the respective IBS registers.

Processors that set CPUID_Fn8000001B_EAX[IbsBufferV1] (bit 17) = 1 support the following three types of IBS Buffer records shown below. Future versions of IBS buffering may extend these records by adding additional bytes. Software should always use the size read from the IBS Buffer record to advance to the next IBS Buffer entry. When IBS Buffering is supported the IbsBrTarget and IbsFetchCtlExt MSRs are also supported.

Fetch IBS Record

Offset in bytes	Size	Description
0	byte	Type = 1, Fetch IBS record
1	byte	Size = 40, Fetch IBS record size in bytes
2 .. 7	byte	Reserved
8	quadword	IbsFetchCtl (MSR C001_1030h)
16	quadword	IbsFetchLinAd (MSR C001_1031h)
24	quadword	IbsFetchPhysAd (MSR C001_1032h)
32	quadword	IbsFetchCtlExt (MSR C001_103Ch)

Execution IBS Record

Offset in bytes	Size	Description
0	byte	Type = 2, Execution IBS record
1	byte	Size = 64, Execution IBS Buffer Record size in bytes
2 .. 7	byte	Reserved
8	quadword	IbsOpRip (MSR C001_1034h)
16	quadword	IbsOpData1 (MSR C001_1035h)
24	quadword	IbsOpData2 (MSR C001_1036h)
32	quadword	IbsOpData3 (MSR C001_1037h)
40	quadword	IbsDcLinAd (MSR C001_1038h)
48	quadword	IbsDcPhysAd (MSR C001_1039h)
56	quadword	IbsBrTarget (MSR C001_103Bh)

Timestamp Counter (TSC) Record

Offset in bytes	Size	Description
0	byte	Type = 3, TSC record
1	byte	Size = 16, TSC Record size in bytes
2 .. 7	byte	Reserved
8	quadword	Time Stamp Counter (MSR 10h)

IBS Buffering Virtualization

IBS Buffering is virtualized when IBS Virtualization is enabled. IBS virtualization is enabled when bit 2 at offset B8h in the VMCB is set to 1 for non SEV-ES guests and when bit 12 in SEV_FEATURES in VMSA is set to 1 for SEV-ES guests.

When IBS Virtualization is enabled, the following fields in the VMCB State Save Area or VMSA hold the guest IBS Buffering register values.

Offset	Size	Contents	Notes
7D8h	quadword	IbsBufferBase	IBS Virtualization state (swap type C)
7E0h	doubleword	IbsBufferSize[31:0]	
7D4h	doubleword	IbsBufferTail[31:0]	

IBS Memory Profiler

This section describes the Instruction Based Sampling (IBS) Memory Profiler facility. The IBS Memory Profiler functions like IBS Execution Sampling but omits sample information which is less relevant to profiling data memory operations. Support for IBS Memory Profiler Version 1 is indicated by CPUID Fn8000_0001B_EAX[IbsMemProfilerV1] (bit 18) = 1.

The IBS Memory Profiler does not collect information on memory access for instruction fetch. Hardware virtualization and IBS Buffering are not supported for IBS Memory Profiler Version 1.

Enabling and Using IBS Memory Profiler

The Memory Profiler IBS facility utilizes the same programming model as IBS Execution Sampling which is described in *"AMD64 Architecture Programmer's Manual"*, Volume 2, Section 13.3.3, *"IBS Execution Sampling"*. The following mapping between IBS Execution Sampling and IBS Memory Profiler registers exists:

IBS Execution Sampling Register	IBS Memory Profiler Register	Description
IbsOpCtl	IbsMemCtl	Control register 1
IbsOpCtl2	IbsMemCtl2	Control register 2
IbsOpRip	IbsMemRip	Instruction Linear Address Register
IbsOpData1	–	Not supported for IBS Memory Profiler
IbsOpData2	IbsMemData2	Northbridge and L2 miss related performance data
IbsOpData3	IbsMemData3	Data Cache (first-level) cache performance data
IbsDcLinAd	IbsMemLinAd	Linear address of the data access
IbsDcPhysAd	IbsMemPhysAd	Physical address of the data access
IbsBrTarget	–	Not supported for IBS Memory Profiler

All IBS Filtering capabilities supported for IBS Execution Sampling are also supported for IBS Memory Profiler. Due to the IBS Memory Profiler's focus on memory operations, it is recommended to use the IBS Memory Profiler with at least one of these IBS Filtering controls:

- *IbsMemCtl1[IbsMemL3MissOnly] = 1* – L3 Miss Filtering
- *IbsMemCtl[IbsMemLatFltEn] = 1* – Load Latency Filtering
- *IbsMemCtl2[IbsMemStrmStFilter] = 1* – Streaming Store Filtering

When a sample has been collected and the IBS Memory Profiler sets IbsMemCtl[IbsMemVal] = 1 an interrupt is signaled to the local APIC. The interrupt source differs from the interrupt source used from for Fetch and Execution IBS and can be discovered in IbsCtl[IbsMemElvtOffset] when IbsCtl[IbsMemElvtOffset] = 1.

IBS Memory Profiler Registers

Unless stated otherwise, support for these registers, bits and fields is indicated by CPUID Fn8000_0001B_EAX[IbsMemProfilerV1] (bit 18) = 1.

IBS Memory Profiler Control Register (IbsMemCtl) (MSR C001_0380h)

Bits	Mnemonic	Description	Access Type
63	IbsMemLatFltEn	IBS Mem Load Latency Filter Enable	R/W
62:59	IbsMemLatThrsh	IBS Mem Load Latency Threshold	R/W
58:32	IbsMemCurCnt[26:0]	IBS Mem Current Op Count, bits 26:0	R/W
31:27	Reserved		RAZ
26:20	IbsMemMaxCnt[26:20]	IBS Mem Maximum Op Count, bits 26:20	R/W
19	IbsMemCntCtl	IBS Mem Op Counter Control	R/W
18	IbsMemVal	IBS Mem Sample Valid	R/W
17	IbsMemEn	IBS Mem Sampling Enable	R/W
16	IbsMemL3MissOnly	IBS Mem L3 Miss Filtering	R/W
15:0	IbsMemMaxCnt[19:4]	IBS Mem Maximum Op Count, bits 19:4	R/W

The fields of IbsMemCtl are further described below:

- *IbsMemLatFltEn (IBS Mem Load Latency Filter Enable)* – Bit 63, read/write. When set, Load Latency Filtering for the IBS Memory Profiler is enabled.
- *IbsMemLatThrsh (IBS Mem Load Latency Threshold)* – Bits 62:59, read/write. When IbsMemLatFltEn = 1, this field specifies the load latency threshold. Memory Profiler IBS samples meet the load latency filter criterium if they are a load op (IbsMemData3[IbsMemLdOp] = 1), not a software prefetch op (IbsMemData3[IbsMemSwPf] = 0) and their load latency (IbsMemData3[IbsDcMissLat]) is greater than (IbsMemLatThrsh + 1) * 128. See section *"IBS Filtering Extensions"* of this document for IBS Filtering capabilities and behavior that might be supported on some Zen6 products.
- *IbsMemCurCnt[26:0] (IBS Mem Op Current Count)* – Bits 58:32, read/write. This field returns the current value of the op counter, and provides the starting value of the counter when software writes this register to clear the IbsOpVal bit and start another sampling interval.
- *IbsMemMaxCnt[26:20] (IBS Mem Op Maximum Count[26:20])* – Bits 26:20, read/write. This field is used to specify the most significant 7 bits of IbsMemMaxCnt.
- *IbsMemCntCtl (IBS Mem Op Counter Control)* – Bit 19, read/write. This bit controls op tagging. When this bit is zero, IBS Memory Profiler counts core clock cycles to select an op for tagging. When this bit is one, IBS counts dispatched ops to select an op for tagging.
- *IbsMemVal (IBS Mem Sample Valid)* – Bit 18, read/write. This bit is set when a tagged op retires and passes the enabled IBS filter criteria. It indicates that new IBS Memory Profiler sample data is available. The op counter stops counting. An interrupt is generated as specified by the local APIC. The software interrupt handler captures the performance data before clearing the bit to enable the hardware to take another sample.

- *IbsMemEn (IBS Mem Sample Enable)* – Bit 17, read/write. Software sets this bit to enable IBS Memory Profiler sampling. Clearing this bit disables IBS Memory Profiler sampling.
- *IbsMemL3MissOnly (IBS Mem L3 Miss Filtering)* – Bit 16, read/write. This bit controls L3 Miss Filtering. When set, IBS Memory Profiler only sends interrupts for samples with an L3 miss. See *"AMD64 Architecture Programmer's Manual"*, Volume 2, Section 13.3.5, *"IBS Filtering"* for a description of Execution IBS Filtering which also applies to Memory Profiler IBS Filtering. Also, see section "*IBS Filtering Extensions*" of this document for IBS Filtering capabilities and behavior that might be supported on some Zen6 products.
- *IbsMemMaxCnt[19:4] (IBS Mem Op Maximum Count[19:4])* – Bits 15:0, read/write. This field specifies the maximum count value for bits 19:4 of the op interval counter. When the value in bits 26:4 of the op interval counter equals the value specified by the concatenation of the IbsOpMaxCnt[26:20] field with this field, the next op is tagged for profiling.

IBS Memory Profiler Control Register 2 (IbsMemCtl2) (MSRC001_0381h)

Bits	Mnemonic	Description	Access Type
63:4	Reserved		RAZ
3	IbsMemStrmStFilter	IBS Mem Streaming Store Filter Control	R/W
2	IbsMemExclRip63Eq1	Exclude IBS Mem samples with RIP bit 63 set	R/W
1	IbsMemExclRip63Eq0	Exclude IBS Mem samples with RIP bit 63 cleared	R/W
0	IbsMemDis	IBS Mem Sampling disable	R/W

The fields of IbsMemCtl2 are further described below:

- *IbsMemStrmStFilter (IBS Mem Streaming Store Filter Control)* – Bit 3, read/write. This bit controls Memory Profiler IBS Filtering. When set, samples for streaming stores (IbsMemData2[StrmSt] = 1) are reported. For more detail on Zen6 IBS Filtering see section *"IBS Filtering Extensions*".
- *IbsMemExclRip63Eq1 (Exclude IBS Mem samples with RIP bit 63 set)* – Bit 2, read/write. This bit controls Memory Profiler IBS Filtering. When set, Memory Profiler IBS excludes samples when the logical instruction address (IbsMemRip) has bit 63 set.
- *IbsMemExclRip63Eq0 (Exclude IBS Mem samples with RIP bit 63 cleared)* – Bit 1, read/write. This bit controls Memory Profiler IBS Filtering. When set, Memory Profiler IBS excludes samples when the logical instruction address (IbsMemRip) has bit 63 cleared.
- *IbsMemDis (IBS Mem Sampling Disable)* – Bit 0, read/write. When set, Memory Profiler IBS is disabled. Memory Profiler IBS is only enabled if IbsMemDis = 0, IbsMemCtl[IbsMemEn] = 1, and IbsMemCtl[IbsMemVal] = 0.

IBS Memory Profiler RIP Register (IbsMemRip) (MSR C001_0386h)

Bits	Mnemonic	Description	Access Type
63:0	IbsMemRip	IBS Mem Op Linear Address	R/W

The fields of IbsMemRip are further described below:

- *IbsMemRip[63:0] (IBS Mem Op Linear Address)* – Bits 63:0, read/write. Specifies the linear address for the instruction from which the tagged op was issued. The address is valid only if the IbsMemCtl[IbsMemVal] bit is set. The address is in canonical form.

IBS Memory Profiler Data 2 Register (IbsMemData2) (MSR C001_0382h)

The IbsMemData2 Register captures Northbridge and system related performance data. The information captured is implementation-dependent. See the *Processor Programming Reference Manual (PPR)* applicable to your product for details.

IBS Memory Profiler Data 3 Register (IbsMemData3) (MSR C001_0383h)

Bits	Mnemonic	Description	Access Type
63:48	Reserved		RAZ
47:32	IbsMemDcMissLat[15:0]	IBS Mem Data Cache Miss Latency	R/W
31:26	IbsMemDcMissOpenReqs[5:0]	IBS Mem Outstanding Memory Requests on DC Fill	R/W
25:22	IbsMemWidth	IBS Mem Access Size	R/W
21	IbsMemSwPf	IBS Mem Software Prefetch	R/W
20	IbsMemL2Miss	IBS Mem L2 Cache Miss	R/W
19	Reserved		RAZ
18	IbsMemDcPhyAddrValid	IBS Mem Data Cache Physical Address Valid	R/W
17	IbsMemDcLinAddrValid	IBS Mem Data Cache Linear Address Valid	R/W
16	IbsMemMissNoMabAlloc	IBS Mem No Miss Buffer Allocation	R/W
15	IbsMemLockedOp	IBS Mem Locked Op	R/W
14	IbsMemUcMemAcc	IBS Mem UC Memory Access	R/W
13	IbsMemWcMemAcc	IBS Mem WC Memory Access	R/W
12:9	Reserved		RAZ
8	IbsMemMisAcc	IBS Mem Misaligned Access Penalty	R/W
7	IbsMemDcMiss	IBS Mem Data Cache Miss	R/W
6:2	Reserved		RAZ
1	IbsMemStOp	IBS Mem Store Op	R/W
0	IbsMemLdOp	IBS Mem Load Op	R/W

The fields of IbsMemData3 are further described below:

- *IbsMemDcMissLat[15:0] (IBS Mem Data Cache Miss Latency)* - Bits 47:32, read/write. This field indicates the number of core clock cycles from when a miss is detected in the data cache to when the data is delivered to the core. The value is not valid for data cache store operations.
- *IbsMemDcMissOpenReqs[5:0] (IBS Mem Outstanding Memory Requests on DC Fill)* – Bits 31:26, read/write. The number of outstanding data cache miss requests when the data cache fill data for the

tagged data cache miss op is received. The count Includes the fill request by the sampled op. A value of 0 means that no information is provided.

- *IbsMemWidth[3:0] (IBS Mem Access Size)* – Bits 25:22, read/write. The encoded number of bytes the tagged load or store is accessing.

Value	Memory Access Size
0h	No information provided
1h	byte
2h	word
3h	doubleword (4 bytes)
4h	quadword (8 bytes)
5h	16 bytes
6h	32 bytes
7h	64 bytes
8h-Fh	Reserved

- *IbsMemSwPf (IBS Mem Software Prefetch)* – Bit 21, read/write. This bit is set when the tagged op belongs to a data prefetch instruction.
- *IbsMemL2Miss (IBS Mem L2 Cache Miss)* – Bit 20, read/write. This bit is set if the cache line used by the tagged load or store operation was not present in the L2 cache.
- *IbsMemDcPhyAddrValid (IBS Mem Data Cache Physical Address Valid)* – Bit 18, read/write. This bit is set if the physical address in the IBS Memory Profiler DC Physical Address Register (IbsMemPhysAd) is valid for a load or store operation.
- *IbsMemDcLinAddrValid (IBS Mem Data Cache Linear Address Valid)* – Bit 17, read/write. This bit is set if the physical address in the IBS Memory Profiler DC Linear Address Register (IbsMemLinAd) is valid for a load or store operation.
- *IbsMemMissNoMabAlloc (IBS Mem No Miss Buffer Allocation)* – Bit 16, read/write. This bit is set if the tagged load or store operation matched an already outstanding cache miss and no new miss buffer entry was allocated.
- *IbsMemLockedOp (IBS Mem Locked Op)* – Bit 15, read/write. This bit is set if the tagged load or store operation was a locked operation.
- *IbsMemUcMemAcc (IBS Mem UC Memory Access)* – Bit 14, read/write. This bit is set if the tagged load or store operation accessed uncacheable memory.
- *IbsMemWcMemAcc (IBS Mem Cache WC Memory Access)* – Bit 13, read/write. This bit is set if the tagged load or store operation accessed Write-Combining memory.
- *IbsMemMisAcc (IBS Memory Misaligned Access Penalty)* – Bit 8, read/write. This bit is set if a tagged load or store operation incurred a performance penalty due to misaligned access.
- *IbsMemDcMiss (IBS Data Cache Miss)* – Bit 7, read/write. This bit is set if the cache line used by the tagged load or store operation was not present in the data cache.
- *IbsMemStOp (IBS Mem Store Op)* – Bit 1, read/write. This bit is set if the tagged op was a store.
- *IbsMemLdOp (IBS Mem Load Op)* – Bit 0, read/write. This bit is set if the tagged op was a load.

IBS Memory Profiler DC Linear Address Register (IbsMemLinAd) (MSR C001_0384h)

Bits	Mnemonic	Description	Access Type
63:0	IbsMemLinAd	IBS Mem Linear Address	R/W

The fields of IbsMemLinAd are further described below:

- *IbsMemLinAd[63:0] (IBS Mem Linear Address)* – Bits 63:0, read/write. Specifies the linear address of the tagged op's memory operand. The address is valid only if IbsMemData3[IbsMemLinAdVal] is set. The address is in canonical form.

IBS Memory Profiler DC Physical Address Register (IbsMemPhysAd) (MSR C001_0385h)

Bits	Mnemonic	Description	Access Type
63:0	IbsMemPhysAd	IBS Mem Physical Address	R/W

The fields of IbsMemPhysAd are further described below:

- *IbsMemPhysAd[63:0] (IBS Mem Physical Address)* – Bits 63:0, read/write. Specifies the physical address of the tagged op's memory operand. The address is valid only if IbsMemData3[IbsMemPhysAdVal] is set.

General IBS Registers

The following register contains information on Fetch, Execution and Memory Profiler IBS.

IBS Control (IbsCtl) (MSR C001_103Ah)

Bits	Mnemonic	Description	Access Type
63:25	Reserved		RAZ
24	IbsMemElvtOffsetVal	Memory Profiler IBS Extended LVT Offset Valid	R
23:20	Reserved		RAZ
19:16	IbsMemElvtOffset	Memory Profiler IBS Extended LVT Offset	R
15:9	Reserved		RAZ
8	IbsElvtOffsetVal	Fetch and Execution IBS Extended LVT Offset Valid	R
7:4	Reserved		RAZ
3:0	IbsElvtOffset	Fetch and Execution IBS Extended LVT Offset	R

The bits and fields of IbsCtl are described below.

- *IbsMemElvtOffsetVal (Memory Profiler IBS Extended LVT Offset Valid)* – Bit 24, read/error-on-write. When set, IbsMemElvtOffset (bits 19:16) reports a valid value. This bit is supported when CPUID Fn8000_0001B_EAX[IbsMemProfilerV1] (bit 18) = 1.
- *IbsMemElvtOffset (Memory Profiler IBS Extended LVT Offset)* – Bits 19:16, read/error-on-write. This field is valid when IbsMemElvtOffsetVal (bit 24) is set and reports the Extended Local Vector Table

offset used for IBS Memory Profiler interrupts. This bit is supported when CPUID Fn8000_0001B_EAX[IbsMemProfilerV1] (bit 18) = 1.

- *IbsElvtOffsetVal (Fetch and Execution IBS Extended LVT Offset Valid)* – Bit 8, read/error-on-write. When set, IbsElvtOffset (bits 3:0) reports a valid value.
- *IbsElvtOffset (Fetch and Execution IBS Extended LVT Offset)* – Bits 3:0, read/error-on-write. This field is valid when IbsElvtOffsetVal (bit 8) is set and reports the Extended Local Vector Table offset used for IBS Fetch and Execution Sampling interrupts.

APIC Register Extension

Zen6 extends the Extended Interrupt Local Vector Table from 4 to 8 entries. The additional Extended Interrupt [7:4] Local Vector Table Registers are located at offsets 540h though 570h in the APIC Register space and MSR addresses 854h through 857h in x2APIC mode.

The Guest vAPIC register access behavior for these additional Extended Interrupt [7:4] Local Vector Table Registers are:

- CPUID Fn8000_000A_EDX[27]=1: Read: Allowed, Write #VMEXIT(trap)
- CPUID Fn8000_000A_EDX[27]=0: Read: VMEXIT(fault), Write: #VMEXIT(fault)