**Architectural Uncore Frequency Scaling TPMI Interface**

**Introduction**

TPMI (Topology Aware Register and PM Capsule Interface), planned for future Intel® Xeon® processor generations, is an architectural, PCIe-standards based model, where PM feature support is provided cleanly as a driver and not as part of the base OS.

Uncore Frequency Scaling provides a set of algorithms that can provide dynamic change to fabric frequency based on the current workload requirement. These changes can enable power savings, increase performance, and reduce latency.

Today, the Uncore Frequency Scaling interface today uses Model Specific Registers (MSR) which would require complex enabling and high software maintenance to meet the needs of future SoCs. Uncore Frequency Scaling will transition to using a driver that utilizes the TPMI interface in the future.

This document provides the pertinent specifications for Uncore Frequency Scaling.

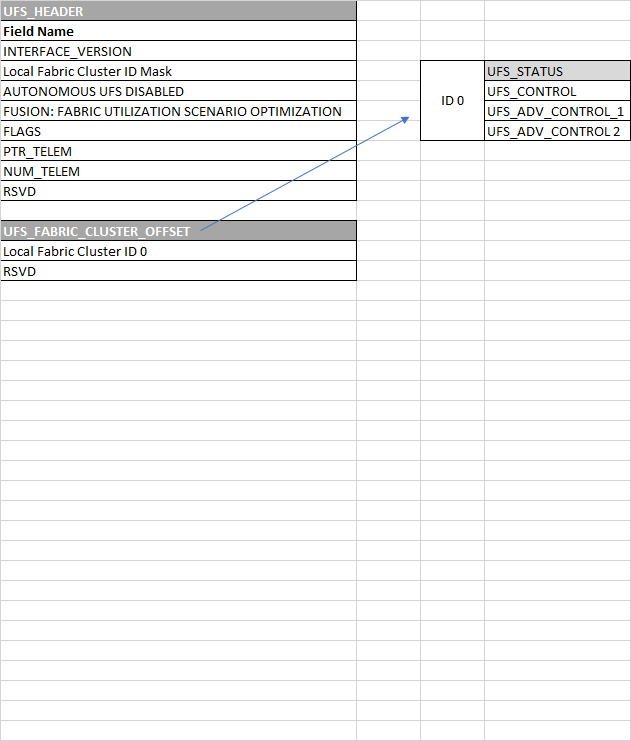
**Software Interface**

TPMI is the only SW interface for future Uncore Frequency Scaling features. The legacy MSR 0x620, 0x621 and BIOS\_SPARE2 registers are deprecated.

All TPMI Uncore Frequency Scaling registers are per-die scope. SW discovers the number of Uncore Frequency Scaling register instances through standard TPMI discovery mechanism.

**Uncore Frequency Scaling Register map**

This map is per die.

[](https://docs.intel.com/documents/pm_doc/src/server/GNR/Features/Hierarchical%20UFS/assets/UFS_sheets.xlsx)

**MAP**

**Throttle Modes**

Future Intel® Xeon® Processors will support a handful a Uncore Frequency Scaling throttle modes as listed below.

**Uncore Frequency Scaling Throttle Modes (**[**source**](https://docs.intel.com/documents/pm_doc/src/server/GNR/Features/Hierarchical%20UFS/assets/UFS_sheets.xlsx)**)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Throttle Mode bit position** | **Throttle Mode bit value** | **Throttle Mode Name** | **Frequency Bounds** | **Use Case** | **Pcode Algorithm(s)** | ***Uncore Frequency Scaling Heuristics*** | ***RAPL line*** |
| 0 | 0 | Power Limited Ordered Throttling | min=max | Latency sensitive customers (comms). Don't change mesh frequency at all. | Disable RAPL line  Disable Uncore Frequency Scaling heuristics | *No* | *No* |
| 0 | 0 | Power Limited Ordered Throttling | min<max | CPU manages mesh frequency autonomously within their bounds, but don't throttle mesh when power is limited. Another case is when boosting performance of core bound workload, i.e., core is not throttled when power is limited until mesh frequency has reached floor. | Disable RAPL line  Enable Uncore Frequency Scaling Heuristic  clip final freq within min/max bounds. | *Yes* | *No* |
| 0 | 1 | Power Limited Proportional Throttling | min<max | CPU manages mesh frequency autonomously within the customer provided min/max bounds | enable RAPL line  enable Uncore Frequency Scaling Heuristics  clip final freq within min/max bounds | *Yes* | *Yes* |
| 0 | 1 | Power Limited Proportional Throttling | min=max | Customer wants traffic agnostic mesh frequency, but when power is limited it is okay to throttle mesh frequency. | enable RAPL line  disable Uncore Frequency Scaling heuristics  clip final freq within min/max bounds | *No* | *Yes* |
| 1, 2, 3, 4, 5, 6 & 7 | Reserved | Reserved | Reserved | Reserved | Reserved | *Reserved* | *Reserved* |

Uncore Frequency Scaling registers are as defined in the sections below.

**Register Indexing (**[**source**](https://docs.intel.com/documents/pm_doc/src/server/GNR/Features/Hierarchical%20UFS/assets/UFS_sheets.xlsx)**)**

|  |  |
| --- | --- |
| **Uncore Frequency Scaling Registers** | |
| Name | Index |
| UFS\_HEADER | 0 |
| UFS\_FABRIC\_CLUSTER\_OFFSET | 1 |
| UFS\_STATUS | 0 |
| UFS\_CONTROL | 1 |
| UFS\_ADV\_CONTROL\_1 | 2 |
| UFS\_ADV\_CONTROL\_2 | 3 |

**UFS\_HEADER**

**Header Register (**[**source**](https://docs.intel.com/documents/pm_doc/src/server/GNR/Features/Hierarchical%20UFS/assets/UFS_sheets.xlsx)**)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UFS\_HEADER** |  |  |  |  |  |
| **Field Name** | **Bits** | **Width** | **Access Type** | **Description** | **Default** |
| INTERFACE\_VERSION | 7:0 | 8 | RO | Version number for this interface | 0x01 |
| LOCAL\_FABRIC\_CLUSTER\_ID\_MASK | 15:8 | 8 | RO | ID assigned for each Fabric V/F Domain. | 0x01 |
| FLAGS | 31:16 | 16 | RO | Bit mask of the supported domain register.  Pcode populates default setting.  SW use it to discover which register is valid in the Uncore Frequency Scaling register bank. | 0 |
| AUTONOMOUS\_UFS\_DISABLED | 32:32 | 1 | RO | 0=Autonomous Uncore Frequency Scaling algorithm is supported. 1=not supported | 0 |
| FUSION | 33:33 | 1 | RO | FUSION: FABRIC UTILIZATION SCENARIO OPTIMIZATION. 1=Mesh Boot Algorithm is supported. 0=not supported | 0 |
| RATIO\_UNIT | 35:34 | 2 | RO | Frequency ratio unit. 00: 100MHz. All others : Reserved. | 0 |
| RSVD | 63:36 | 28 | RO | reserved | 0 |

Please note that min and max supported values can be different for each dielet.

The P0, P1, Pm ratios are present in the SST\_PP\_INFO-11 register in the [SST HAS](https://docs.intel.com/documents/pm_doc/src/server/Wave3_common/SST/Intel_SST.html). It is the per-PP level mesh frequency info register.

**Global Fabric ID Computation in Software**

**Examples to compute Global Fabric ID (**[**source**](https://docs.intel.com/documents/pm_doc/src/server/GNR/Features/Hierarchical%20UFS/assets/UFS_sheets.xlsx)**)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Example 1: Two clusters per dielet** |  |  |  |
| **Global fabric ID** | **local fabric ID** | **Die ID** | **Cluster\_ID\_MASK** |
| 0 | 0 | 0 | 0000\_0011 |
| 1 | 1 | 0 |  |
| 2 | 0 | 1 | 0000\_0011 |
| 3 | 1 | 1 |  |
| 4 | 0 | 2 | 0000\_0011 |
| 5 | 1 | 2 |  |
| 6 | 0 | 3 | 0000\_0011 |
| 7 | 1 | 3 |  |
|  |  |  |  |
| **Example 2: one cluster per dielet** |  |  |  |
| **Global fabric ID** | **local fabric ID** | **Die ID** | **Cluster\_ID\_MASK** |
| 0 | 0 | 0 | 0000\_0001 |
| 1 | 0 | 1 | 0000\_0001 |
| 2 | 0 | 2 | 0000\_0001 |
| 3 | 0 | 3 | 0000\_0001 |

Num of local fabric clusters in one dielet = Count the number of 1s in CLUSTER\_ID\_MASK.

Global Fabric ID = (Die ID \* Num of local fabric clusters in one dielet) + Local fabric cluster ID.

Max num of local fabric clusters = 1 for future Intel® Xeon Processors.

Each fabric with unique ID will have different observability register and control registers.

Future Intel® Xeon® processors will only have one fabric cluster per die. Therefore, Local fabric ID cluster mask is 8'b0000\_0001 and only ID0 is used per die.

**UFS\_FABRIC\_CLUSTER\_OFFSET**

**UFS\_FABRIC\_CLUSTER\_OFFSET (**[**source**](https://docs.intel.com/documents/pm_doc/src/server/GNR/Features/Hierarchical%20UFS/assets/UFS_sheets.xlsx)**)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UFS\_FABRIC\_CLUSTER\_OFFSET** |  |  |  |  |  |
| **Field Name** | **Bits** | **Width** | **Access Type** | **Description** | **Default** |
| OFFSET\_0 | 7:0 | 8 | RO | Offset (Qword, 8 bytes) for status and control registers belonging to local cluster ID 0 | 0x02 |
| RSVD | 63:8 | 56 | RO | reserved | 0 |

**STATUS**

**Uncore Frequency Scaling Status Register (**[**source**](https://docs.intel.com/documents/pm_doc/src/server/GNR/Features/Hierarchical%20UFS/assets/UFS_sheets.xlsx)**)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UFS\_STATUS** |  |  |  |  |  |
| **Field Name** | **Bits** | **Width** | **Access Type** | **Description** | **Default** |
| CURRENT\_RATIO | 6:0 | 7 | RO | Instantaenous fabric frequency ratio | 0 |
| CURRENT\_VOLTAGE | 22:7 | 16 | RO | Indicates current Fabric voltage in U3.13 format | 0 |
| AGENT\_TYPE\_CORE | 23:23 | 1 | RO | 1: At least one core agent exists on the fabric cluster; 0: No core agent is present on the fabric cluster. | 0 |
| AGENT\_TYPE\_CACHE | 24:24 | 1 | RO | 1: At least one cache agent exists on the fabric cluster; 0: No cache agent is present on the fabric cluster. | 0 |
| AGENT\_TYPE\_MEMORY | 25:25 | 1 | RO | 1: At least one memory agent exists on the fabric cluster; 0: No memory agent is present on the fabric cluster. | 0 |
| AGENT\_TYPE\_IO | 26:26 | 1 | RO | 1: At least one IO agent exists on the fabric cluster; 0: No IO agent is present on the fabric cluster. | 0 |
| RSVD | 31:27 | 5 | RO | Reserved | 0 |
| THROTTLE\_COUNTER | 63:32 | 32 | RO | Count the number of 1ms intervals in which the fabric frequency violated the freq bound provided in Uncore Frequency Scaling control register. Increment counter only once within a 1ms interval if there is violation. | 0 |

**CONTROL**

**Uncore Frequency Scaling Control Registers (**[**source**](https://docs.intel.com/documents/pm_doc/src/server/GNR/Features/Hierarchical%20UFS/assets/UFS_sheets.xlsx)**)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UFS\_CONTROL** |  |  |  |  |  |
| **Field Name** | **Bits** | **Width** | **Access Type** | **Description** | **Default** |
| UFS\_THROTTLE\_MODE | 1:0 | 2 | RW | Select one of the Uncore Frequency Scaling throttle modes | 1 |
| RSVD1 | 7:2 | 6 | RW | Reserved | 0 |
| MAX\_RATIO | 14:8 | 7 | RW | Max fabric domain frequency ratio | 0x7F |
| MIN\_RATIO | 21:15 | 7 | RW | Min fabric domain frequency ratio | 0 |
| EFFICIENCY\_LATENCY\_CTRL\_RATIO | 28:22 | 7 | RW | Fabric domain frequency ratio floor while in the low power activity region determined by Efficiency\_Latency\_Ctrl. | 0 |
| RSVD2 | 31:29 | 3 | RW | Reserved | 0 |
| EFFICIENCY\_LATENCY\_CTRL\_THRESHOLD | 38:32 | 7 | RW | This field provides the flexibility to alter the region of low power activity. It deteremines the region of Mesh utilization points to which the Efficiency\_Latency\_Ctrl mode will be applied. | 0 |
| RSVD3 | 63:39 | 25 | RW | Reserved | 0 |
|  |  |  |  |  |  |
| **UFS\_ADV\_CONTROL\_1** | | | | |  |
| **Field Name** | **Bits** | **Width** | **Access Type** | **Description** | **Default** |
| SLOPE\_1 | 7:0 | 8 | RW | Slope that controls how fast the mesh frequency is brought down with core frequency when the socket is power limited. In 1/16 ratio bins (S4.3 format). | 0 |
| BASE\_1 | 15:8 | 8 | RW | The core frequency below which mesh frequency is brought down when socket is power limited. (S7.0 format) | 0 |
| RSVD | 63:16 | 48 | RW | Reserved | 0 |
|  |  |  |  |  |  |
| **UFS\_ADV\_CONTROL\_2** | | | | |  |
| **Field Name** | **Bits** | **Width** | **Access Type** | **Description** | **Default** |
| SLOPE\_2 | 7:0 | 8 | RW | Slope that controls how fast the mesh frequency is brought down with core frequency when the socket is power limited. In 1/16 ratio bins (S4.3 format). | 0 |
| BASE\_2 | 15:8 | 8 | RW | The core frequency below which mesh frequency is brought down when socket is power limited. (S7.0 format) | 0 |
| UTILIZATION\_THRESHOLD | 23:16 | 8 | RW | Mesh Utilization Threshold. 255 = 100% utilization, 0 = 0% utilization. | 0xFF |
| HBM\_BW\_THRESHOLD | 31:24 | 8 | RW | HBM BW Threshold | 0xFF |
| RSVD | 63:32 | 32 | RW | Reserved | 0 |

The slope and offset default settings will be tuned by Intel for pre-defined workloads. When SST PP level changes then its software responsibility to change slope and intercept as needed. By default, if software writes to this interface, no matter the SST PP level, the SoC will choose that overwritten slope and intercept.

Customers should be able to use the UFS\_CONTROL register to select a throttle mode that suits their workload needs.

UFS\_ADV\_CONTROL exposes knobs for the more sophisticated customers who have desires to further fine-tune Uncore Frequency Scaling behaviors.

Slope1/base1 are for the legacy Uncore Frequency Scaling RAPL line while slope2/base2 for the Mesh Boost RAPL line.

**Acceptable Values from Software:**

We expect that software entity writing to this interface will make sure the following conditions are met for the feature to work correctly:

1. Min <= Max frequency (never write min value large than max value).
2. All dielets must have the same UFS\_CONTROL.ufs\_throttle\_mode settings. If the ufs\_throttle\_mode is different, the behavior is undefined.