Intel® Core™ Ultra Processor for Network and Edge Platforms

Datasheet Addendum

Rev. 001

February 2024

Doc. No.: 793432, Rev.: 001

Contents

Figures

Tables

Revision History

Table 1. Revision History

Date	Revision	Description
February 2024	001	• Initial release.

1.0 Introduction

This Intel® Core™ Ultra Processor for Network and Edge Platforms Datasheet is a supplementary document that covers Network and Edge-specific features and complements the Intel® Core™ Ultra Processor Datasheet Volume 1 of 2 (RDC Document #792044) and Intel® Core™ Ultra Processor Datasheet – Volume 2 of 2 (RDC Document #795249).

The Intel® Core™ processor is a 64-bit, multi-core processor built on Intel® 4 process technology.

Throughout this document, Intel® Core™ Ultra Processor for Network and Edge Platforms Datasheet is used as a general term to refer to the Intel® Core™ Ultra Processor Datasheet Volume 1 of 2 (RDC Document #792044) and Intel® Core™ Ultra Processor Datasheet – Volume 2 of 2 (RDC Document #795249) all SKUs, unless mentioned otherwise.

NOTE

This document serves as a supplementary documentation. Please ensure all required specifications from the Intel® Core™ Ultra Processor Datasheet are met prior to using this document (unless mentioned otherwise).

1.1 Terminology

Table 2. Terminology

Term	Description
BGA	Ball Grid Array
cTDP	Assured Power (previously known as configurable TDP)
TDP	Processor Base Power (a.k.a Thermal Design Power)

1.2 Reference Documents

Table 3. Reference Documents

Document	Document No./ Location
Intel® Core™ Ultra Processor Datasheet Volume 1 of 2	792044
Intel® Core™ Ultra Processor Datasheet Volume 2 of 2	795249

1.3 Supported Package and Die Plan

Table 4. Processor Lines Package and Die Plan

Processor Line Segment	Die Type	Package (mm)	TDP (cTDP)
H-Processor (6+8)	6P + 8E + 2	50 mm x 25 mm x 1.35mm	28 W (20W)
H-Processor (4+8)			
U-Processor (2+8)	2P + 8E + 2		15W (12W)

2.0 Thermal Management

2.1 Thermal Management Introduction

The Intel® Core™ Ultra processor is a 64-bit, multi-core processor built on the Intel® 4 fabrication process technology.

Table 5. Intel® Core™ Ultra Processor Line

Segment	Package	SKU Series	Processor Base Power (a.k.a TDP)	cTDP Down / Up	Processor IA P-Cores	Processor IA E-Cores	Graphics Configuration
H-Processor (6+8)	BGA2049	PC Client	28W	20W / 65W	6	8	8 X^e
H-Processor (4+8)			28W		4	8	7 X^e or 8 X^e
U-Processor (2+8)			15W	12W / 28W	2	8	3 X^e or 4 X^e

Notes:

1. Processor lines offer may change.
2. For more details, refer to Intel® Core™ Ultra Processor Datasheet Volume 1 of 2 (RDC #792044).
3. The Processor Base Power (a.k.a TDP) is the average power dissipation in junction temperature operating condition limit for the SKU Segment and Configuration. The processor is validated during manufacturing when executing an associated Intel-specified high-complexity workload at the processor IA core frequency corresponding to the configuration and SKU.
4. TDP workload may consist of a combination of intensive processor IA core and intensive graphics core applications.
5. P-Cores refer to Performance-cores and E-Cores refer to Efficient-cores.

2.2 Intel® Core™ Ultra Processor Line Thermal and Power Specifications

The notes below are additional for Thermal and Power Specifications aside from those mentioned in Thermal Management of the Intel® Core™ Ultra Processor Datasheet Volume 1 of 2 (RDC #792044)

Table 6. Processor Temperature Specifications

Segment	SKU Series	Symbol	Description	Temperature Range		TDP Specification Temperature Range		Notes
				Min	Max	Min	Max	
U/H-Processor Line BGA	PC Client	T_j	Junction Temperature	0°C	110°C	35°C	110°C	1,2,3,4

continued...

Segment	SKU Series	Symbol	Description	Temperature Range		TDP Specification Temperature Range		Notes
				Min	Max	Min	Max	
			Limit					
		DTR	DTR Temperature Limit	Tj Min	Tj Max	N/A	N/A	5

Notes:

1. The processor supports commercial ambient temperature range, and memory devices should operate in the same condition.
2. For other junction temperature specification, refer to the note mentioned in Processor Line Thermal and Power Specifications under Thermal Management in the Intel® Core™ Ultra Processor Datasheet (RDC Document #792044).
3. The T_j used to define U-Processor Base Power (TDP) is 100°C. Operating the part at Tj_max (110°C) is feasible but may result in slightly higher power than TDP.
4. The processor junction temperature is monitored by Digital Temperature Sensors (DTS). For DTS accuracy, refer to Digital Thermal Sensor under Intel® Core™ Ultra Processor Datasheet (RDC Document #792044).
5. DTR is the range of T_j covering from T_{j_min} to T_{j_max} and/or T_{j_max} to T_{j_min}. No re-boot required for any boot temperature within the T_j limit range.

3.0 Display

3.1 Pipelock

3.1.1 Overview

Pipelock is defined as the synchronization between ports driven by a single GPU. Genlock is defined as the synchronization between ports driven from multiple GPUs. For timing synchronization, the display hardware supports port sync, CMTG and Genlock features. For a single GPU use case, it is recommended to use (legacy) port sync for tiled DP display and Genlock for synchronization of multiple external displays

NOTE

Pipelock is a sub-feature of Genlock that vsync timing of all secondary pipes under Genlock is synced on the primary pipe's timing. Pipelock is activated when enabling the Genlock as the primary pipe.

Figure 1. High Level Diagram of a Single Host iGPU Pipelock

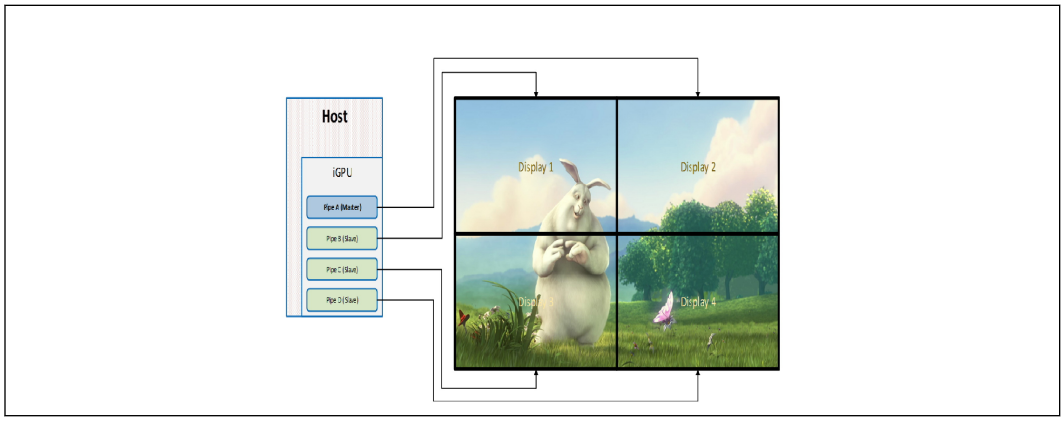


Figure 1 illustrates the single host iGPU Pipelock where one of the pipes acts as a primary display, and its vsync timings are synchronized with the secondary displays.

3.2 Bezel Correction

This feature allows users to adjust the image to correct the physical monitor bezel so images that span across multiple monitors can be visually adjusted. Current versions of the Intel® Graphics Command Center now support this capability. The following figures show illustrations of how bezel correction can be used. The top illustration shows how the image will appear without enabling bezel correction. The bottom illustration shows how the image will appear after enabling bezel correction and setting the value to a desired number.

Figure 2. Without Bezel Correction

Figure 3. With Bezel Correction

4.0 Graphics

4.1 SRIOV Overview

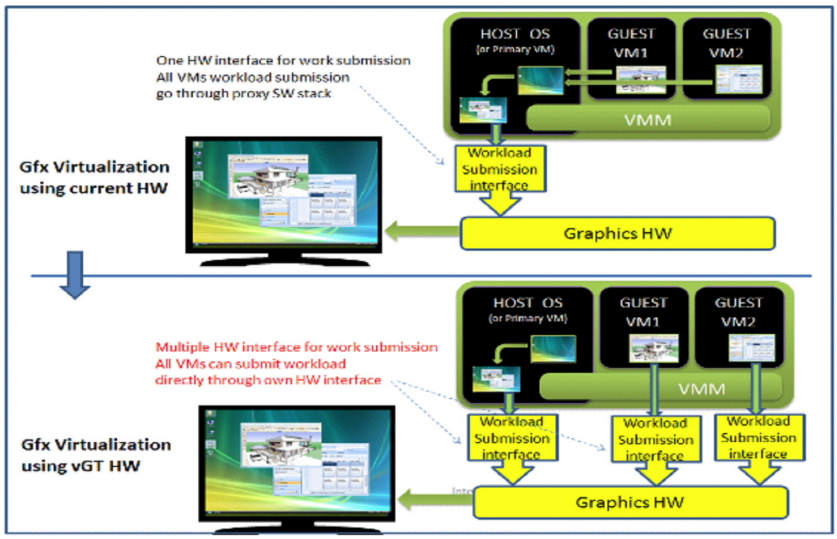
Graphics virtualization allows multiple Virtual Machines (VMs) to access high-quality, high-performance graphics, with minimal software overhead. The graphics virtualization feature adds hardware and firmware to improve performance and enable VMMs to support Intel® HD Graphics, Intel® Iris® graphics, and Intel® Iris® Pro graphics in a standard way. This eliminates special requirements that could be barriers to adoption.

4.2 Introduction

Intel® platforms have supported the Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d). The original Intel® Graphics Virtualization Technology (Intel® GVT) model using (Intel® VT-d)technologies is called GVT-d, which assigns the whole graphics to a single VM. It brings close-to-native GPU performance to the VM which owns the graphics. However, it prevents other VMs and the host sharing the capabilities of the graphics. By contrast, Intel® Graphics SR-IOV Enablement Toolkit provides graphics sharing capabilities across multiple VMs and host without compromise in graphics performance.

Figure 4 shows the new concept of Intel® Graphics Virtualization Technology (Intel® GVT), where each VM can access fully accelerated graphics capabilities.

Figure 4. Fully Virtualized Graphics - Conceptual View

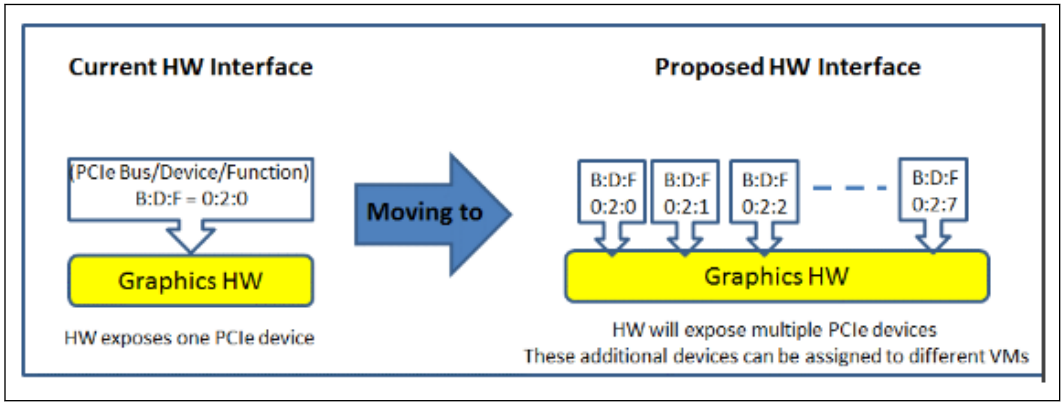


The capability depicted above requires the following infrastructure:

- Graphics hardware that exposes multiple hardware interfaces for assignment to different VMs
- Resources to submit the graphics workload:
  - Memory (aperture) allowing each VM to submit workload and associated data surfaces
  - A graphics translation table (GTT) to manage aperture pages
  - Some MMIO registers exposed to each VM
- A signaling mechanism to invoke the hardware to execute the workload
- A signaling mechanism for the hardware to convey relevant information by sending interrupts
- A mechanism to display the output produced by the execution of the workload

The figure below shows the hardware interface used to allow multiple software stacks to each get their own "graphics device."

Figure 5. Hardware Interface



4.3 Hardware Interface

4.3.1 Display Output from a Virtualized Environment

There are several possible ways to display content from a virtualized graphics environment:

- Rely on the virtual machine monitor (VMM) or PF driver to composite rendered surfaces from different VMs
- Assign a display engine exclusively to a virtual function (VF), preferably a trusted VF from the VMM perspective, and use the corresponding planes/pipes to output data
- Assign a display pipe or plane to a VF

The graphics virtualization infrastructure is exposed to the system software by the Single Root I/O virtualization standard (SR-IOV) of the PCIe standard. The exposure is accomplished using a PCIe device Physical function that includes the SR-IOV Extended Capability structure, within the PCIe Extended Capabilities list.

4.3.2 GuC

Graphics Micro Controller (GuC) is an embedded micro-controller in the graphics sub-system that is designed to perform graphics workload scheduling on the various graphics parallel engines. In this scheduling model, the host software submits work through one of the 256 graphics doorbells, and this invokes the micro-kernel running on the GuC core to perform the scheduling operation on the appropriate graphics engine.

Scheduling operations include determining which workload to run next, submitting a workload to a command streamer, pre-empting existing workloads running on an engine, monitoring progress, and notifying the host software when work is done. To perform these actions, the GuC requires access to a wide range of assets within the graphics subsystem. The GuC has access to the entire graphics device MMIO register space to allow it to schedule work on any graphics engine.

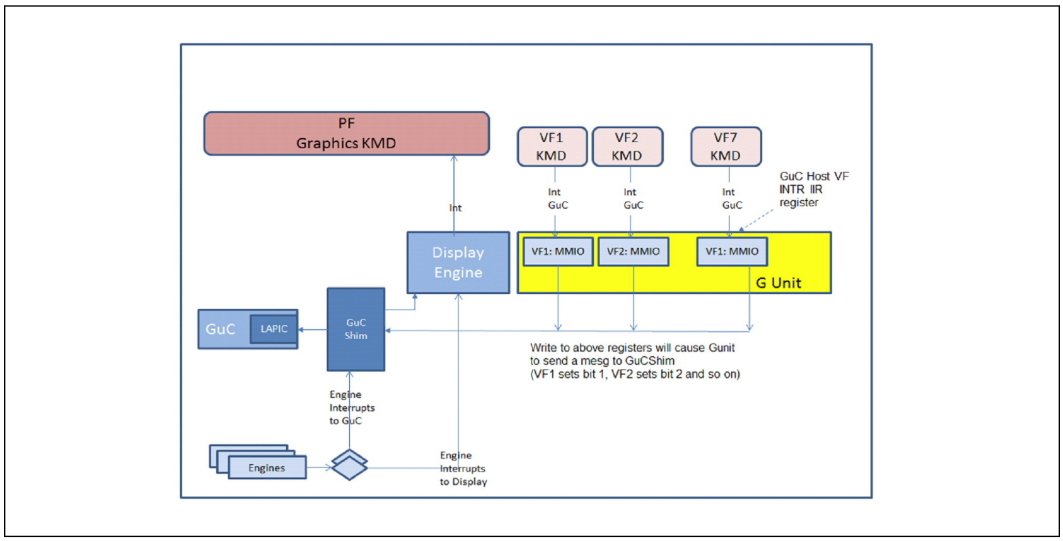
4.3.2.1 Virtual Function Communication with GuC

The Virtual Function (VF) software stack needs to be able to communicate with the GuC. For example, it needs to be able to communicate to execute the VM assigned task shown in Figure 6. This communication occurs via the VF's GuC interrupt path (i.e., Host2GuC path). Similarly, the VF software stack may require some services from the PF software stack.

An example communication sequence is as follows:

- VF Kernel Mode Driver (KMD) requires services from the PF stack.
- VF Kernel Mode Driver (KMD) writes to the VF-GuC-Interrupt MMIO register with a pre-defined message.
- The MMIO write triggers an interrupt to the GuC.
- The GuC firmware interrupt routine executes and inspects the MMIO register.
- GuC firmware uses message content to determine whether GuC or PF is the intended message target:
  - If GuC is the target, then GuC firmware consumes the message and provides the appropriate service; or
  - If PF software is the target, then GuC generates an interrupt to the PF. PF provides the appropriate service and communicates back to GuC upon completion.
- GuC interrupts the VF to indicate that the service is completed.

Figure 6. Virtual Function to Physical Function Communication through GuC



5.0 Enhanced Serial Peripheral Interface (eSPI)

Note: For any capability not mentioned in this section, refer to Enhanced Serial Peripheral Interface (eSPI) of Datasheet Volume 1 (RDC# 792044).

1. Functional Overview

The eSPI controller supports up to 4 devices. The eSPI controller supports 20 MHz, 25 MHz, 33 MHz, and 50 MHz. An eSPI device can support frequencies lower than the recommended maximum frequency (50 MHz). In addition, the eSPI device must support a minimum frequency of 20 MHz for default (reset) communication between the host and device.

Second Target Device Presence

NOTE

The eSPI controller does not perform discovery to confirm the presence of the target connection. If a second device is enabled by soft strap, it must be physically present on the platform. If the second device is enabled by soft strap but is not physically present, the platform will fail to boot.

Table 7. Signal Description

Signal Name	Type	Description	Availability
GPP_A13/ESPI_CS1#/USB-C_GPP_A13	O	eSPI Chip Select1: Driving CS# signallow to select eSPI device for the transaction.	H/U Processor Line
GPP_A14/ESPI_CS2#/USB-C_GPP_A14	O	eSPI Chip Select 2: Driving CS# signal low to selecteSPI device for the transaction.	H/U Processor Line
GPP_A15/ESPI_CS3#/USB-C_GPP_A15	O	eSPI Chip Select 3: Driving CS# signal low to selecteSPI device for the transaction.	H/U Processor Line
GPP_A16/ESPI_ALERT0#/ USB-C_GPP_A16	I	eSPI Alert 0: Alert signal from eSPI deviceto the Processor. Note: If only a single device is connected, the eSPI Compatibility Specification requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.	H/U Processor Line
GPP_A17/ESPI_ALERT1#/ USB-C_GPP_A17	I	eSPI Alert 1: Alert signal from eSPI device to the Processor. Note: If only a single device is connected, the eSPI Compatibility Spec requires that the device must operate with in-band	H/U Processor Line

continued...

		Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.	
GPP_A18/ESPI_ALERT2#/ USB-C_GPP_A18	I	eSPI Alert 2: Alert signal from eSPI device to the Processor. Note: If only a single device is connected, the eSPI Compatibility Spec requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.	H/U Processor Line
GPP_A19/ESPI_ALERT3#/ USB-C_GPP_A19	I	eSPI Alert 3: Alert signal from eSPI device to the Processor. Note: If only a single device is connected, the eSPI Compatibility Spec requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.	H/U Processor Line

Table 8. Integrated Pull-Ups and Pull-Downs

Signal	**Resistor Type**	**Value**	**Notes**
ESPI_CS [3:0] #	Pull-up	20 kohm +/- 30%	
ESPI_ALERT [3:0] #	Pull-up	20 kohm +/- 30%	

Table 9. I/O Signal Planes and States

Signal Name	**Power Plane**	**During Reset**	**Immediately after Reset[1]**	**S4/S5**
ESPI_CS [3:0] #	Primary	Internal Pull-up	Driven High	Driven High
ESPI_ALERT [3:0] #	Primary	Internal Pull-up	Driven High	Driven High
Note: 1. Reset reference for primary well pins is RSMRST#.				

5.1 Functional Overview

Enter a short description of your topic here (optional).

The eSPI controller supports up to 4 devices. The eSPI controller supports 20 MHz, 25 MHz, 33 MHz, and 50 MHz. An eSPI device can support frequencies lower than the recommended maximum frequency (50 MHz). In addition, the eSPI device must support a minimum frequency of 20 MHz for default (reset) communication between the host and device.

NOTE

Second Target Device Presence

NOTE

The eSPI controller does not perform discovery to confirm the presence of the target connection. If a second device is enabled by soft strap, it must be physically present on the platform. If the second device is enabled by soft strap but is not physically present, the platform will fail to boot.

Table 10. Signal Description

Signal Name	Type	Description	Availability
GPP_A13/ESPI_CS1#/USB-C_GPP_A13	O	eSPI Chip Select1: Driving CS# signallow to select eSPI device for the transaction.	H/U Processor Line
GPP_A14/ESPI_CS2#/USB-C_GPP_A14	O	eSPI Chip Select 2: Driving CS# signal low to selecteSPI device for the transaction.	H/U Processor Line
GPP_A15/ESPI_CS3#/USB-C_GPP_A15	O	eSPI Chip Select 3: Driving CS# signal low to selecteSPI device for the transaction.	H/U Processor Line
GPP_A16/ESPI_ALERT0#/ USB-C_GPP_A16	I	eSPI Alert 0: Alert signal from eSPI deviceto the Processor. Note: If only a single device is connected, the eSPI Compatibility Specification requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.	H/U Processor Line
GPP_A17/ESPI_ALERT1#/ USB-C_GPP_A17	I	eSPI Alert 1: Alert signal from eSPI device to the Processor. Note: If only a single device is connected, the eSPI Compatibility Spec requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.	H/U Processor Line
GPP_A18/ESPI_ALERT2#/ USB-C_GPP_A18	I	eSPI Alert 2: Alert signal from eSPI device to the Processor. Note: If only a single device is connected, the eSPI Compatibility Spec requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.	H/U Processor Line
GPP_A19/ESPI_ALERT3#/ USB-C_GPP_A19	I	eSPI Alert 3: Alert signal from eSPI device to the Processor. Note: If only a single device is connected, the eSPI Compatibility Spec requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.	H/U Processor Line

Table 11. Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
ESPI_CS [3:0] #	Pull-up	20 kohm +/- 30%	
ESPI_ALERT [3:0] #	Pull-up	20 kohm +/- 30%	

Table 12. I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset[1]	S4/S5
ESPI_CS [3:0] #	Primary	Internal Pull-up	Driven High	Driven High
ESPI_ALERT [3:0] #	Primary	Internal Pull-up	Driven High	Driven High
Note: 1. Reset reference for primary well pins is RSMRST#.				

5.2 Pinlist

Table 13. Pinlist for Enhanced Serial Peripheral Interface (eSPI) 4-loads

Signal Name/ Interface	Package Name	GPIO Native Function	GPIO Native Direction	Package	Pin Map
ESPI_CS1#	GPP_A13/ ESPI_CS1#/USB-C_GPP_A13	1	Out	U/H	EE48
ESPI_CS2#	GPP_A14/ESPI_CS2#/USB-C_GPP_A14	1	Out	U/H	EE43
ESPI_CS3#	GPP_A15/ ESPI_CS3#/USB-C_GPP_A15	1	Out	U/H	EH48
ESPI_ALERT0#	GPP_A16/ESPI_ALERT0#/ USB-C_GPP_A16	1	In	U/H	EL48
ESPI_ALERT1#	GPP_A17/ESPI_ALERT1#/ USB-C_GPP_A17	1	In	U/H	EK48
ESPI_ALERT2#	GPP_A18/ESPI_ALERT2#/ USB-C_GPP_A18	1	In	U/H	EE46
ESPI_ALERT3#	GPP_A19/ESPI_ALERT3#/ USB-C_GPP_A19	1	In	U/H	EC43