NVIDIA Data Center GPU Driver version 525.125.06 (Linux)/ 529.11 (Windows)

Release Notes

RN-08625-525 _v4.0 | July 2023

Table of Contents

Chapter 1. Version Highlights

This section provides highlights of the NVIDIA Data Center GPU R535 Driver (version 525.125.06 Linux and 529.11 Windows).

For changes related to the 535 release of the NVIDIA display driver, review the file "NVIDIA_Changelog" available in the .run installer packages.

- Linux driver release date: 06/26/2023
- Windows driver release date: 06/26/2023

1.1. Software Versions

For this release, the software versions are as follows:

- CUDA Toolkit 12: 12.0 Update 1

  Note that starting with CUDA 11, individual components of the toolkit are versioned independently. For a full list of the individual versioned components (for example, nvcc, CUDA libraries, and so on), see the CUDA Toolkit Release Notes.
- NVIDIA Data Center GPU Driver: 525.125.06 (Linux) / 529.11 (Windows)
- Fabric Manager: 525.125.06 (Use `nv-fabricmanager -v`)
- NVFlash: 5.791

For more information on getting started with the NVIDIA Fabric Manager on NVSwitch-based systems (for example, NVIDIA HGX A100), refer to the Fabric Manager User Guide.

1.2. Fixed Issues

- 4093278 - `GpuFabricProbe` info was accessed without validity checks. This is now validated before using its attributes to send fabric probe.
- 3954952 - Redfish Row Remapping Failed event was not logged when DRAM uncontained was injected at the same address after resetting GPU.
- 4046386 - On Viking systems, CUDA waits for 5s for link training to complete (if it is not complete already). It was observed on some systems that the link training may take up to 10s. The fabric probe timeout in CUDA is now increased from 5s to 30s, which should be ample for link training to complete.

- 4079365 - On H100, atomicops capability was not available. PCIE atomic capabilities are now enabled for Genoa and SPR family CPUs.
- 4087329 - A Linux kernel device driver API used for timer management in the Linux kernel interface of the NVIDIA GPU driver is susceptible to a race condition under multi-GPU configurations. The Linux kernel interface of the NVIDIA GPU driver has been updated to use a different Linux kernel device driver API for timer management, which properly addresses the race condition inherent in the previously-used API.
- 4026611 - Fixed an issue specific to GSP-RM that could lead to GSP RPC timeout errors (Xid 119). The issue was most likely to happen after installing a new driver without resetting the GPU.
- 4100483 - Updated the communication request handling to PMU to update RPC stats only when PMU should be ready to process request. Previously we always updated this statistics, even when RPC-s were issues too early / too late.
- 4101679 - On GPU resets, SXIDs were seen in kernel logs which disabled interrupts used for setting up fabric state, causing customers to report unsuccessful fabric state initialization. Fix here was to squash the SXIDs on GPU reset when NvLink links are re-trained to active between LS10 and GH100 (this prevents interrupts used for fabric state setup from being disabled).
- 4102371 - For security, certain routing error handling state can only be modified by secure clients. Kernel driver is not secure enough, so handling of some routing error handling state is offloaded to SOE (Secure Offload Engine).
- 4101829 - Corrected an issue where the Fabric Manager service fails to initialize when the underlying operating system has an excessively high open file handle limit.
- 4032382 - Resolved an issue where the Fabric Manager service would unexpectedly terminate on earlier production sample NVIDIA HGX H800 systems due to its failure in reading necessary board type information.
- 4025007 - Updated the NVSwitch regression testing to include link fault testing.
- 4109287 - By default, when `CUDA_ENABLE_COREDUMP_ON_EXCEPTION=1` is set in the user environment and a GPU fault occurs, a GPU coredump is generated followed by a CPU coredump. There is an environment variable that users can set to cause the CPU coredump to not be generated (`CUDA_ENABLE_CPU_COREDUMP_ON_EXCEPTION=0`). This bugfix addresses an issue where the `CUDA_ENABLE_CPU_COREDUMP_ON_EXCEPTION` environment variable wasn't being honored and a CPU coredump would be generated in all cases.

1.3. Known Issues

General

- When polling the H100 GPU via SMBPBI using GPU Performance Monitoring metrics, driver reloads or GPU resets can result in driver errors that manifest as PID (X62) errors on Linux. NVIDIA is investigating this issue and more information will be updated here soon.

- On NVIDIA H800, monitoring software such as DCGM or NVML might report lower double-precision (FP64) utilization metrics. This is expected as per the NVIDIA H800 product configuration. Refer to the NVIDIA H800 product brief for more details.
- For some SKUs of GH100 the MIG profile name reported by `cuDeviceGetName`, particularly the number of compute instances, might be incorrect. Use `nvidia-smi` to query the actual loaded MIG profile names. Only `cuDeviceGetName` is affected; developers are recommended to query the precise SM information for precise configuration. This will be fixed in a subsequent driver release.
- "Change ECC State" and "Enable Error Correction Code" do not change synchronously when ECC state changes.
- The GPU driver build system might not pick the `Module.symvers` file, produced when building the `ofa_kernel` module from MLNX_OFED, from the right subdirectory. Because of that, `nvidia_peermem.ko` does not have the right kernel symbol versions for the APIs exported by the IB core driver, and therefore it does not load correctly. That happens when using MLNX_OFED 5.5 or newer on a Linux Arm64 or ppc64le platform.

  To work around this issue, perform the following:

  1. Verify that nvidia_peermem.ko does not load correctly.
  2. Uninstall old MLNX_OFED if one was installed.
  3. Manually remove `/usr/src/ofa_kernel/default` if one exists.
  4. Install MLNX_OFED 5.5 or newer.
  5. Manually create a soft link:
     ```
     /usr/src/ofa_kernel/default -> /usr/src/ofa_kernel/$(uname -m)/$(uname -r)
     ```
  6. Reinstall the GPU driver.
- If you encounter an error on RHEL7 when installing with `cuda-drivers-fabricmanager` packages, use the following alternate instructions. For example:

  If you are upgrading from a different branch, for example to driver 515.65.01:

  ```
  new_version=515.65.01
  sudo yum swap nvidia-driver-latest-dkms nvidia-driver-latest-dkms-${new_version}
  sudo yum install nvidia-fabric-manager-${new_version}
  ```
- When installing a driver on SLES15 or openSUSE15 that previously had an R515 driver installed, users need to run the following command afterwards to finalize the installation:

  ```
  sudo zypper install --force nvidia-gfxG05-kmp-default
  ```

  Without doing this, users may see the kernel objects as missing.
- `nvidia-release-upgrade` may report that not all updates have been installed and exit.

  When running the

  ```
  nvidia-release-upgrade
  ```

command on DGX systems running DGX OS 4.99.x, it may exit and tell users: "Please install all available updates for your release before upgrading" even though all upgrades have been installed.

Users who see this can run the following command:

```
sudo apt install -y nvidia-fabricmanager-450/bionic-updates --
allow-downgrades
```

After running this, proceed with the regular upgrade steps:

```
sudo apt update
sudo apt full-upgrade -y
sudo apt install -y nvidia-release-upgrade
sudo nvidia-release-upgrade
```

- By default, Fabric Manager runs as a `systemd` service. If using

  `DAEMONIZE=0`

  in the Fabric Manager configuration file, then the following steps may be required.

  1. Disable FM service from auto starting.

     `systemctl disable nvidia-fabricmanager`

  2. Once the system is booted, manually start FM process.

     `/usr/bin/nv-fabricmanager -c /usr/share/nvidia/nvswitch/fabricmanager.cfg`

     Note, since the process is not a daemon, the SSH/Shell prompt will not be returned (use another SSH shell for other activities or run FM as a background task).

- Important correctness fix for H100 GPU instructions used by cuBLAS, other CUDA libraries, and user CUDA code

  An issue was discovered recently with H100 GPUs (H100 PCIe and HGX H100) where certain operations put the GPU in an invalid state that allowed some GPU instructions to operate at unsupported frequency that can result in incorrect computation results and faster than expected performance. The affected GPU instructions are used by cuBLAS, other CUDA libraries, and can also be used for user CUDA code.

  The operations that allow the GPU to enter an invalid state are the following:

  - Enabling MIG
  - Deinitialize and reinitialize the GPU (for example, turn off persistence mode and turn it back on or reload the nvidia.ko driver)
  - Any Compute Engine error (for example, MMU fault, Out of Range warp error, and so on)

  Once the GPU enters the invalid state, the performance for some GPU instructions is increased by 7-10%, but the computation results may be incorrect.

  The current release fixes this issue, and it is no longer possible to enter the invalid GPU state. This issue has been present in all drivers since the H100 launch, and we recommend that you upgrade to the current release as soon as possible. If upgrading

is not immediately possible, a GPU reset can restore the GPU back to the correct operational state, except for when MIG is being used. For MIG, the new driver is required, and there is no workaround available.

- Uninstalling the driver fails, and the system reboots automatically.

  On Windows 2019 and 2022 servers, uninstalling the driver causes the system to restart automatically before the uninstallation is completed. The issue also occurs when you upgrade the driver from an older version to a new version, even after selecting the Perform Clean Installation option in the installer UI.

  Note: This issue does not occur in Linux.

  Workaround

  We strongly recommend that you always install, uninstall, and upgrade drivers from Safe mode.

- In Shared Switch virtualization mode, the guest VM GPU driver load and unload stress test fails after certain iteration

  In the Shared Switch virtualization mode, the stress test to load and unload the GPU driver on Guest VM in every 30 second interval runs into issues approximately after three hours of the test.

  Workaround

  Do not run the stress reload driver cycle at this time.

- A few Async SMBPBI commands do not function as intended when the driver is unloaded.

  When the driver is unloaded, the following Async SMBPBI commands do not operate as specified:

  - Arg1 0x00: Reads total GPU power limit control data.
  - Arg1 0x01: Sets the total GPU power limit.
  - Arg1 0x02: Reads the total GPU power limit policy information.

  Due to this issue, some properties of the following Redfish URIs are impacted:

  - PowerLimitWatts.SetPoint:

    `/redfish/v1/Systems/HGX_Baseboard_0/Processors/GPU_SXM_[1-8]/EnvironmentMetrics`

  - SpeedLimitMHz, SpeedLocked:

    `/redfish/v1/Systems/HGX_Baseboard_0/Processors/GPU_SXM_[1-8]`

  The Patch operation of the following URIs are impacted:

  - PowerLimitWatts.SetPoint:

`/redfish/v1/Systems/HGX_Baseboard_0/Processors/GPU_SXM_[1-8]/EnvironmentMetrics`

- Oem.Nvidia.PowerMode "MaxP" or "MaxQ":

  `/redfish/v1/Chassis/HGX_Chassis_0/EnvironmentMetrics`

- SpeedLimitMHz, SpeedLocked:

  `/redfish/v1/Systems/HGX_Baseboard_0/Processors/GPU_SXM_[1-8]`

Workaround

Load the driver for these URIs to work properly.

- Fabric Manager state is not reported accurately on NVSwitch OOB query

  The NVSwitch SMPBI query that reports Fabric Manager state (Manager State) is not reporting the actual FM state.

- Instructions to reset all GPUs Using the nvidia-smi -r Command

  When resetting all GPUs using the `nvidia-smi` command with the `-r` option instead of a resetting specific GPU using the `-i <gpu_index>` option, all the NVSwitches will also be reset. This process wipes out the NVSwitch routing entries, and subsequent CUDA application launches will fail. The Fabric Manager service will also show interaction errors with the NVSwitch device via the switch driver.

  Workaround

  1. Stop the Fabric Manager service.
  2. To reset all GPUs, run `nvidia-smi -r`.
  3. After the reset is finished, start the Fabric Manager service.

GPU Performance Counters

The use of developer tools from NVIDIA that access various performance counters requires administrator privileges. See this note for more details. For example, reading NVLink utilization metrics from `nvidia-smi (nvidia-smi nvlink -g 0)` would require administrator privileges.

NoScanout Mode

NoScanout mode is no longer supported on NVIDIA Data Center GPU products. If NoScanout mode was previously used, then the following line in the “screen” section of /etc/X11/xorg.conf should be removed to ensure that X server starts on data center products:

```
Option         "UseDisplayDevice" "None"
```

NVIDIA Data Center GPU products now support one display of up to 4K resolution.

Unified Memory Support

CUDA and unified memory is not supported when used with Linux power management states S3/S4.

IMPU FRU for Volta GPUs

The driver does not support the IPMI FRU multi-record information structure for NVLink. See the Design Guide for Tesla P100 and Tesla V100-SXM2 for more information.

OpenCL 3.0 Known Issues

Device side enqueue

- Device-Side-Enqueue related queries may return 0 values, although corresponding built-ins can be safely used by kernel. This is in accordance with conformance requirements described at https://www.khronos.org/registry/OpenCL/specs/3.0-unified/html/OpenCL_API.html#opencl-3.0-backwardscompatibility
- Shared virtual memory - the current implementation of shared virtual memory is limited to 64-bit platforms only.

Chapter 2. Virtualization

To make use of GPU passthrough with virtual machines running Windows and Linux, the hardware platform must support the following features:

- A CPU with hardware-assisted instruction set virtualization: Intel VT-x or AMD-V.
- Platform support for I/O DMA remapping.
- On Intel platforms, the DMA remapper technology is called Intel VT-d.
- On AMD platforms, it is called AMD IOMMU.

Support for these features varies by processor family, product, and system, and should be verified at the manufacturer's website.

The following hypervisors are supported for virtualization:

Hypervisor	Notes
Citrix XenServer	Version 6.0 and later
VMware vSphere (ESX / ESXi)	Version 5.1 and later.
Red Hat KVM	Red Hat Enterprise Linux 7 with KVM
Microsoft Hyper-V	Windows Server 2016 Hyper-V Generation 2

Data Center products now support one display of up to 4K resolution.

The following GPUs are supported for device passthrough for virtualization:

GPU Family	Boards Supported
NVIDIA Ada Lovelace	NVIDIA L40, L4
NVIDIA Hopper	NVIDIA H100, NVIDIA H800
NVIDIA Ampere GPU Architecture	NVIDIA A800, A100, A40, A30, A16, A10, A10G, A2
NVIDIA Turing	NVIDIA T4, NVIDIA T4G
NVIDIA Volta	NVIDIA V100
NVIDIA Pascal	Quadro: P2000, P4000, P5000, P6000, GP100 Tesla: P100, P40, P4

GPU Family	Boards Supported
NVIDIA Maxwell	Quadro: K2200, M2000, M4000, M5000, M6000, M6000 24GB Tesla: M60, M40, M6, M4

Chapter 3. Hardware and Software Support

Support for these features varies by processor family, product, and system, and should be verified at the manufacturer's website.

Supported Operating Systems for NVIDIA Data Center GPUs

The Release 525 driver is supported on the following operating systems:

- Windows x86_64 operating systems:
  - Microsoft Windows® Server 2022
  - Microsoft Windows® Server 2019
  - Microsoft Windows® Server 2016

    Note: R525TeslaRD will be the last TRD to support Server 2016.
  - Microsoft Windows® 11 21H2
  - Microsoft Windows® 11 22H2 - SV2
  - Microsoft Windows® 10
- The following table summarizes the supported Linux 64-bit distributions. For a complete list of distributions, kernel versions supported, see the CUDA Linux System Requirements documentation.

Distribution	x86_64	POWER	Arm64 Server
Debian 11.x (where x <= 7)	Yes	No	No
Debian 10. x (where x <= 13)	Yes	No	No
OpenSUSE Leap 15.x (where y <= 4)	Yes	No	No
Fedora 37	Yes	No	No
Red Hat Enterprise Linux 9.y (where y <= 2)	Yes	No	Yes

Distribution	x86_64	POWER	Arm64 Server
Rocky Linux 9.y (where y <= 2)	Yes	No	No
Red Hat Enterprise Linux 8.y (where y <= 8)	Yes	Yes	Yes
Rocky Linux 8.y (where y <= 8)	Yes	No	No
Red Hat Enterprise Linux / CentOS 7.y (where y <= 9)	Yes	No	No
SUSE Linux Enterprise Server 15.y (where y <= 4)	Yes	No	Yes
Ubuntu 22.04.z LTS (where z <= 2)	Yes	No	Yes
Ubuntu 20.04.z LTS (where z <= 6)	Yes	No	Yes
Ubuntu 18.04.z LTS (where z <= 6)	Yes	No	No
KylinOS V10 SP2	Yes	No	No
CBL-Mariner 2.0*	Yes	No	No

* CBL-Mariner will be supported by TRD via runfile. CUDA Toolkit will not support this OS as this is a deployment OS.

Supported Operating Systems and CPU Configurations for NVIDIA HGX H100/H800

The Release 525 driver is validated with NVIDIA HGX H100 on the following operating systems and CPU configurations:

- Linux 64-bit distributions:
  - Red Hat Enterprise Linux 8.7 (in 4/8/16-GPU configurations)
  - Red Hat Enterprise Linux 9.2 (in 4/8/16-GPU configurations)
  - Ubuntu 22.04.2 LTS (in 4/8/16-GPU configurations)
- Windows 64-bit distributions:
  - Windows Server 2022
  - Windows Server 2019 (in 1/2/4/8-GPU configurations; 16-GPU configurations are currently not supported)

Windows is supported only in shared NVSwitch virtualization configurations.

Supported Operating Systems and CPU Configurations for NVIDIA HGX A100/A800

The Release 525 driver is validated with NVIDIA HGX A100 on the following operating systems and CPU configurations:

- Linux 64-bit distributions:
  - Debian 11.7
  - Debian 10.13
  - Red Hat Enterprise Linux 8.7 (in 4/8/16-GPU configurations)
  - Red Hat Enterprise Linux 7.9 (in 4/8/16-GPU configurations)
  - Rocky Linux 8.7 (in 4/8/16-GPU configurations)
  - Red Hat Enterprise Linux 9.2 (in 4/8/16-GPU configurations)
  - CentOS Linux 7.9 (in 4/8/16-GPU configurations)
  - Ubuntu 22.04.2 LTS (in 4/8/16-GPU configurations)
  - Ubuntu 20.04.6 LTS (in 4/8/16-GPU configurations)
  - Ubuntu 18.04.6 LTS (in 4/8/16-GPU configurations)
  - SUSE SLES 15.4 (in 4/8/16-GPU configurations)
  - KylinOS V10 SP2
- Windows 64-bit distributions:
  - Windows Server 2022
  - Windows Server 2019 (in 1/2/4/8-GPU configurations; 16-GPU configurations are currently not supported)

    Windows is supported only in shared NVSwitch virtualization configurations.
- CPU Configurations:
  - AMD Rome in PCIe Gen4 mode
  - Intel Skylake/Cascade Lake (4-socket) in PCIe Gen3 mode

Supported Virtualization Configurations

The Release 525 driver is validated with NVIDIA HGX A100, HGX A800, H100, and H800 on the following configurations:

- Passthrough (full visibility of GPUs and NVSwitches to guest VMs):
  - 8-GPU configurations with Ubuntu 18.04.6 LTS, 20.4.6, and 22.4.2
- Shared NVSwitch (guest VMs only have visibility of GPUs and full NVLink bandwidth between GPUs in the same guest VM):

- 1/2/4/8/16-GPU configurations with Ubuntu 18.04.6 LTS

API Support

This release supports the following APIs:

- NVIDIA® CUDA® 12.0 for NVIDIA® Maxwell™, Pascal™, Volta™, Turing™, Hopper™, NVIDIA Ampere architecture, and NVIDIA Ada Lovelace GPU architecture GPUs
- OpenGL® 4.6
- Vulkan® 1.3
- DirectX 11
- DirectX 12 (Windows 10)
- Open Computing Language (OpenCL™ software) 3.0

Note that for using graphics APIs on Windows (such as OpenGL, Vulkan, DirectX 11, and DirectX 12) or any WDDM 2.0+ based functionality on Data Center GPUs, vGPU is required. See the vGPU documentation for more information.

Supported NVIDIA Data Center GPUs

The NVIDIA Data Center GPU driver package is designed for systems that have one or more Data Center GPU products installed. This release of the driver supports CUDA C/C++ applications and libraries that rely on the CUDA C Runtime and/or CUDA Driver API.

Attention: Release 470 was the last driver branch to support Data Center GPUs based on the NVIDIA Kepler architecture. This includes discontinued support for the following compute capabilities:

- sm_30 (NVIDIA Kepler)
- sm_32 (NVIDIA Kepler)
- sm_35 (NVIDIA Kepler)
- sm_37 (NVIDIA Kepler)

For more information on GPU products and compute capability, see https://developer.nvidia.com/cuda-gpus.

NVIDIA Server Platforms	
Product	**Architecture**
NVIDIA HGX H100	H100 and NVSwitch
NVIDIA HGX H800	H800 and NVSwitch
NVIDIA HGX A800	A800 and NVSwitch
NVIDIA HGX A100	A100 and NVSwitch
NVIDIA HGX-2	V100 and NVSwitch

Data Center L-Series Products	
Product	**GPU Architecture**
NVIDIA L40	NVIDIA Ada Lovelace
NVIDIA L4	NVIDIA Ada Lovelace

Data Center H-Series Products	
Product	**GPU Architecture**
NVIDIA H100 PCIe	NVIDIA Hopper
NVIDIA H800 PCIe	NVIDIA Hopper

RTX-Series / T-Series Products	
Product	**GPU Architecture**
NVIDIA RTX A6000	NVIDIA Ampere architecture
NVIDIA RTX A5000	NVIDIA Ampere architecture
NVIDIA RTX A4000	NVIDIA Ampere architecture
Quadro RTX 8000	NVIDIA Turing
Quadro RTX 6000	NVIDIA Turing
Quadro RTX 4000	NVIDIA Turing
NVIDIA T1000	NVIDIA Turing
NVIDIA T600	NVIDIA Turing
NVIDIA T400	NVIDIA Turing

Data Center A-Series Products	
Product	**GPU Architecture**
NVIDIA A2	NVIDIA Ampere architecture
NVIDIA A800	NVIDIA Ampere architecture
NVIDIA A100X	NVIDIA Ampere architecture
NVIDIA A100 NVIDIA A100 80 GB PCIe	NVIDIA Ampere architecture
NVIDIA A40	NVIDIA Ampere architecture
NVIDIA A30. A30X	NVIDIA Ampere architecture
NVIDIA A16	NVIDIA Ampere architecture
NVIDIA A10, A10M, A10G	NVIDIA Ampere architecture

Data Center T-Series Products	
Product	**GPU Architecture**
NVIDIA T4, T4G	NVIDIA Turing

Data Center V-Series Products	
Product	**GPU Architecture**
NVIDIA V100	Volta

Data Center P-Series Products	
Product	**GPU Architecture**
NVIDIA Tesla P100	NVIDIA Pascal
NVIDIA Tesla P40	NVIDIA Pascal
NVIDIA Tesla P4	NVIDIA Pascal

Data Center M-Class Products	
Product	**GPU Architecture**
NVIDIA Tesla M60	Maxwell
NVIDIA Tesla M40 24 GB	Maxwell
NVIDIA Tesla M40	Maxwell
NVIDIA Tesla M6	Maxwell
NVIDIA Tesla M4	Maxwell

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation (“NVIDIA”) makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer (“Terms of Sale”). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the Unites States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
https://www.nvidia.com