



NVIDIA XID ERROR MESSAGES

R545 | February 2024

Application Note

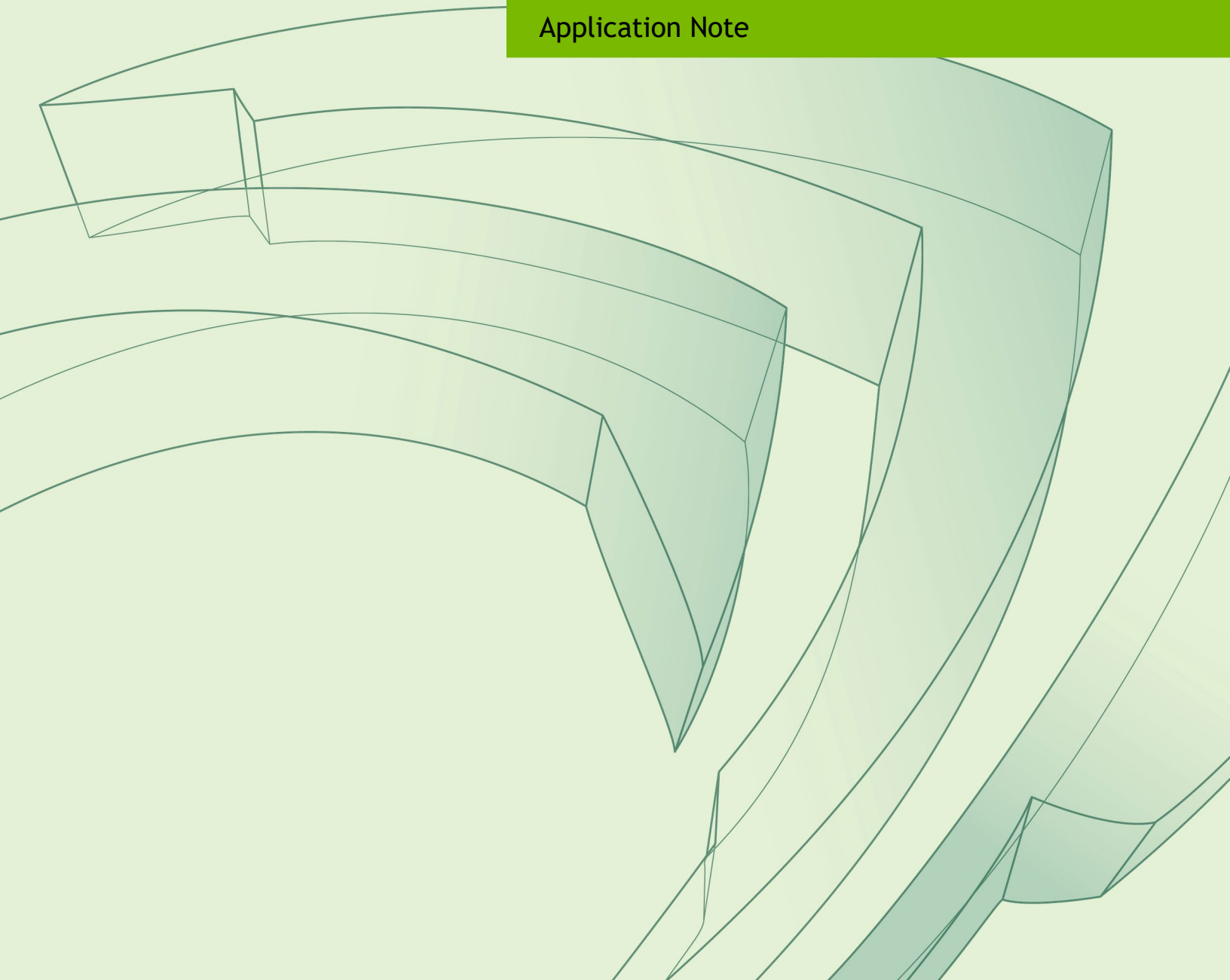


TABLE OF CONTENTS

Document Change History	2
Table of Contents	3
Introduction	4
What Is an Xid Message	4
What Is an SXid Message	4
How to Use Xid Messages	6
Working with Xid Errors	7
Viewing Xid Error Messages	7
Tools That Provide Additional Information About Xid Errors	8
Analyzing Xid Errors.....	9
Xid Error Listing	10
Common XID Errors	15
XID 13: GR: SW Notify Error.....	15
XID 31: Fifo: MMU Error	16
XID 32: PBDMA Error	16
XID 43: RESET CHANNEL VERIF ERROR	16
XID 45: OS: Preemptive Channel Removal	16
XID 48: DBE (Double Bit Error) ECC Error	17
XID 63, 64: ECC Page Retirement or ROW REMAPPING	17
XID 74: Nvlink ERROR	17
XID 79: GPU has fallen off the bus	18
XID 93: Non-fatal violation of provisioned inforom wear limit.....	18
XID 94, 95: CONTAINED/UNCONTAINED ECC ERRORS	18
XID 110 SECURITY FAULT ERROR.....	19
XID 119, 120: GSP RPC Timeout / GSP Error	19

INTRODUCTION

This document explains what Xid messages are, and is intended to assist system administrators, developers, and FAEs in understanding the meaning behind these messages as an aid in analyzing and resolving GPU-related problems.

WHAT IS AN XID MESSAGE

The Xid message is an error report from the NVIDIA driver that is printed to the operating system's kernel log or event log. Xid messages indicate that a general GPU error occurred, most often due to the driver programming the GPU incorrectly or to corruption of the commands sent to the GPU. The messages can be indicative of a hardware problem, an NVIDIA software problem, or a user application problem.

These messages provide diagnostic information that can be used by both users and NVIDIA to aid in debugging reported problems.

The meaning of each message is consistent across driver versions.

WHAT IS AN SXID MESSAGE

NVIDIA drivers for NVSwitch report error conditions relating to NVSwitch hardware in kernel logs through a similar mechanism to Xids. These "Switch Xids", or SXids and guidelines for their usage are documented separately in the following user guide:

<https://docs.nvidia.com/datacenter/tesla/pdf/fabric-manager-user-guide.pdf>

HOW TO USE XID MESSAGES

Xid messages are intended to be used as debugging guides. Because many problems can have multiple possible root causes it's not always feasible to understand each issue from the Xid value alone.

For example, an Xid error might indicate that a user program tried to access invalid memory. But, in theory, memory corruption due to PCIE or frame buffer ("FB") problems could corrupt any command and thus cause almost any error. Generally, the Xid classifications listed below should be used as a starting point for further investigation of each problem.

The following manual provides additional guidance for debugging GPU problems, including advice for interpreting Xids and provides guidance for next steps to handle common Xids.

<https://docs.nvidia.com/deploy/gpu-debug-guidelines/index.html>

WORKING WITH XID ERRORS

VIEWING XID ERROR MESSAGES

Under Linux, the Xid error messages are placed in the location `/var/log/messages`.

Grep for “NVRM: Xid” to find all the Xid messages.

The following is an example of a Xid string:

```
[...] NVRM: GPU at 0000:03:00: GPU-b850f46d-d5ea-c752-ddf3-  
c4453e44d3f7  
[...] NVRM: Xid (0000:03:00): 14, Channel 00000001
```

- ▶ The first Xid in the log file is preceded by a line that contains the GPU GUID and device IDs.

In the above example,

- “GPU-b850f46d-d5ea-c752-ddf3-c4453e44d3f7” is the GUID.
The GUID is a globally unique, immutable identifier for each GPU.
- “0000:03:00” is the device ID.

- ▶ Each subsequent Xid line contains the device ID, the Xid error, and information about the Xid.

In the above example,

- “0000:03:00” is the device ID.
- “14” is the Xid error identifier .
- “Channel 00000001” is data specific to that Xid error.

TOOLS THAT PROVIDE ADDITIONAL INFORMATION ABOUT XID ERRORS

NVIDIA provides two additional tools that may be helpful when dealing with Xid errors.

nvidia-smi is a command-line program that installs with the NVIDIA driver. It reports basic monitoring and configuration data about each GPU in the system. nvidia-smi can list ECC error counts (Xid 48) and indicate if a power cable is unplugged (Xid 54), among other things. Please see the nvidia-smi man page for more info. Run 'nvidia-smi -q' for basic output.

NVIDIA Data Center GPU Manager (DCGM) is a suite of tools for managing and monitoring NVIDIA datacenter GPUs in cluster environments. It includes active health monitoring, comprehensive diagnostics, system alerts and governance policies including power and clock management. DCGM diagnostics is a health checking tool that can check for basic GPU health, including the presence of ECC errors, PCIe problems, bandwidth issues, and general problems with running CUDA programs.

DCGM is documented and downloadable at <https://developer.nvidia.com/dcgm>

nvidia-bug-report.sh is a script that installs with the NVIDIA driver. It collects debug logs and command outputs from the system, including kernel logs and logs collected by the NVIDIA driver itself. The command should be run as root:

```
sudo nvidia-bug-report.sh
```

The output of this tool is a single compressed text file, nvidia-bug-report.log.gz, that can be included when reporting problems to NVIDIA.

nvidia-bug-report.sh will typically run quickly, but in rare cases may run slowly. Allow up to one hour for it complete. If the command remains hung, run the command with additional arguments as:

```
nvidia-bug-report.sh --safe-mode --extra-system-data
```

This will collect alternative logs, in such a way that it should avoid common causes of hangs during debug collection.

ANALYZING XID ERRORS

The following table lists the recommended actions to take for various issues encountered.

Issue	Recommended Action
Suspected User Programming Issues	Run the debugger tools. See the Compute Sanitizer “memcheck” tool, and cuda-gdb documentation at: https://docs.nvidia.com/compute-sanitizer/ComputeSanitizer/index.html#memcheck-tool https://docs.nvidia.com/cuda/cuda-gdb/index.html
Suspected Hardware Problems	Contact the hardware vendor. They can run through their hardware diagnostic process.
Suspected Driver Problems	File a bug with NVIDIA, including output of the command <code>nvidia-bug-report.sh</code> . See the document http://docs.nvidia.com/deploy/gpu-debug-guidelines/index.html for guidance on gathering additional information to provide to NVIDIA and troubleshooting common Xid causes.

XID ERROR LISTING

The following table lists the Xid errors along with the potential causes for each.

XID	Failure	Causes						
		HW Error	Driver Error	User App Error	System Memory Corruption	Bus Error	Thermal Issue	FB Corruption
1	Invalid or corrupted push buffer stream		X		X	X		X
2	Invalid or corrupted push buffer stream		X		X	X		X
3	Invalid or corrupted push buffer stream		X		X	X		X
4	Invalid or corrupted push buffer stream		X		X	X		X
	GPU semaphore timeout		X	X	X	X		X
5	Unused							
6	Invalid or corrupted push buffer stream		X		X	X		X
7	Invalid or corrupted push buffer address		X			X		X
8	GPU stopped processing		X	X		X	X	
9	Driver error programming GPU		X					
10	Unused							
11	Invalid or corrupted push buffer stream		X		X	X		X
12	Driver error handling GPU exception		X					
13	Graphics Engine Exception	X	X	X	X	X	X	X
14	Unused							
15	Unused							
16	Display engine hung		X					
17	Unused							

XID	Failure	Causes						
		HW Error	Driver Error	User App Error	System Memory Corruption	Bus Error	Thermal Issue	FB Corruption
18	Bus mastering disabled in PCI Config Space		X					
19	Display Engine error		X					
20	Invalid or corrupted Mpeg push buffer		X		X	X		X
21	Invalid or corrupted Motion Estimation push buffer		X		X	X		X
22	Invalid or corrupted Video Processor push buffer		X		X	X		X
23	Unused							
24	GPU semaphore timeout		X	X	X	X	X	X
25	Invalid or illegal push buffer stream		X	X	X	X		X
26	Framebuffer timeout		X					
27	Video processor exception		X					
28	Video processor exception		X					
29	Video processor exception		X					
30	GPU semaphore access error		X					
31	GPU memory page fault	X	X	X				
32	Invalid or corrupted push buffer stream		X		X	X	X	X
33	Internal micro-controller error		X					
34	Video processor exception		X					
35	Video processor exception		X					
36	Video processor exception		X					
37	Driver firmware error		X		X	X		
38	Driver firmware error		X					
39	Unused							
40	Unused							
41	Unused							
42	Video processor exception		X					
43	GPU stopped processing		X	X				
44	Graphics Engine fault during context switch		X					
45	Preemptive cleanup, due to previous errors -- Most likely to see when running multiple cuda applications and hitting a DBE		X					
46	GPU stopped processing		X					
47	Video processor exception		X					
48	Double Bit ECC Error	X						
49	Unused							
50	Unused							
51	Unused							
52	Unused							

XID	Failure	Causes						
		HW Error	Driver Error	User App Error	System Memory Corruption	Bus Error	Thermal Issue	FB Corruption
53	Unused							
54	Auxiliary power is not connected to the GPU board							
55	Unused							
56	Display Engine error	X	X					
57	Error programming video memory interface	X	X					X
58	Unstable video memory interface detected	X	X					
	EDC error - clarified in printout	X						
59	Internal micro-controller error (older drivers)		X					
60	Video processor exception		X					
61	Internal micro-controller breakpoint/warning (newer drivers)							
62	Internal micro-controller halt (newer drivers)	X	X				X	
63	ECC page retirement or row remapping recording event	X	X					X
64	ECC page retirement or row remapper recording failure	X	X					
65	Video processor exception	X	X					
66	Illegal access by driver		X	X				
67	Illegal access by driver		X	X				
68	NVDEC0 Exception	X	X					
69	Graphics Engine class error	X	X					
70	CE3: Unknown Error	X	X					
71	CE4: Unknown Error	X	X					
72	CE5: Unknown Error	X	X					
73	NVENC2 Error	X	X					
74	NVLINK Error	X	X			X		
75	CE6: Unknown Error	X	X					
76	CE7: Unknown Error	X	X					
77	CE8: Unknown Error	X	X					
78	vGPU Start Error		X					
79	GPU has fallen off the bus	X	X		X	X	X	
80	Corrupted data sent to GPU	X	X		X	X		X
81	VGA Subsystem Error	X						
82	NVJPG0 Error	X	X					
83	NVDEC1 Error	X	X					
84	NVDEC2 Error	X	X					
85	CE9: Unknown Error	X	X					
86	OFA Exception	X	X					

XID	Failure	Causes						
		HW Error	Driver Error	User App Error	System Memory Corruption	Bus Error	Thermal Issue	FB Corruption
87	Reserved							
88	NVDEC3 Error	X	X					
89	NVDEC4 Error	X	X					
90	Reserved							
91	Reserved							
92	High single-bit ECC error rate	X	X					
93	Non-fatal violation of provisioned InfoROM wear limit		X	X				
94	Contained ECC error	X	X					X
95	Uncontained ECC error	X	X					X
96	NVDEC5 Error	X	X					
97	NVDEC6 Error	X	X					
98	NVDEC7 Error	X	X					
99	NVJPG1 Error	X	X					
100	NVJPG2 Error	X	X					
101	NVJPG3 Error	X	X					
102	NVJPG4 Error	X	X					
103	NVJPG5 Error	X	X					
104	NVJPG6 Error	X	X					
105	NVJPG7 Error	X	X					
106	SMBPBI Test Message			X				
107	SMBPBI Test Message Silent			X				
108	Reserved							
109	Context Switch Timeout Error	X	X	X	X	X	X	X
110	Security Fault Error	X						
111	Display Bundle Error Event	X	X			X		
112	Display Supervisor Error	X	X					
113	DP Link Training Error	X	X					
114	Display Pipeline Underflow Error	X	X					X
115	Display Core Channel Error	X	X					
116	Display Window Channel Error	X	X					
117	Display Cursor Channel Error	X	X					
118	Display Pixel Pipeline Error	X	X					
119	GSP RPC Timeout	X	X		X	X	X	X
120	GSP Error	X	X		X	X	X	X
121	C2C Link Error	X				X		
122	SPI PMU RPC Read Failure	X	X					
123	SPI PMU RPC Write Failure	X	X					
124	SPI PMU RPC Erase Failure	X	X					
125	Infom FS Failure	X	X					
126-139	Reserved							
140	Unrecovered ECC Error	X	X					X

XID	Failure	Causes						
		HW Error	Driver Error	User App Error	System Memory Corruption	Bus Error	Thermal Issue	FB Corruption
141-142	Reserved							
143	GPU Initialization Failure	X	X					X

For the comprehensive list of XIDs, please refer to <https://github.com/NVIDIA/open-gpu-kernel-modules/blob/main/src/common/sdk/nvidia/inc/nverror.h>.

COMMON XID ERRORS

This section provides more information on some common Xid errors.

XID 13: GR: SW NOTIFY ERROR

This event is logged for general user application faults. Typically this is an out-of-bounds error where the user has walked past the end of an array, but could also be an illegal instruction, illegal register, or other case.

In rare cases, it's possible for a hardware failure or system software bugs to materialize as XID 13.

When this event is logged, NVIDIA recommends the following:

1. Run the application in `cuda-gdb` or the Compute Sanitizer “memcheck” tool , or
2. Run the application with `CUDA_DEVICE_WAITS_ON_EXCEPTION=1` and then attach later with `cuda-gdb`, or
3. File a bug if the previous two come back inconclusive to eliminate potential NVIDIA driver or hardware bug.

Note: The Compute Sanitizer “memcheck” tool instruments the running application and reports which line of code performed the illegal read.

XID 31: FIFO: MMU ERROR

This event is logged when a fault is reported by the MMU, such as when an illegal address access is made by an applicable unit on the chip. Typically these are application-level bugs, but can also be driver bugs or hardware bugs.

When this event is logged, NVIDIA recommends the following:

1. Run the application in `cuda-gdb` or the Compute Sanitizer “memcheck” tool, or
4. Run the application with `CUDA_DEVICE_WAITS_ON_EXCEPTION=1` and then attach later with `cuda-gdb`, or
5. File a bug if the previous two come back inconclusive to eliminate potential NVIDIA driver or hardware bug.

Note: The Compute Sanitizer “memcheck” tool instruments the running application and reports which line of code performed the illegal read.

XID 32: PBDMA ERROR

This event is logged when a fault is reported by the DMA controller which manages the communication stream between the NVIDIA driver and the GPU over the PCI-E bus. These failures primarily involve quality issues on PCI, and are generally not caused by user application actions.

XID 43: RESET CHANNEL VERIF ERROR

This event is logged when a user application hits a software induced fault and must terminate. The GPU remains in a healthy state.

In most cases, this is not indicative of a driver bug but rather a user application error.

XID 45: OS: PREEMPTIVE CHANNEL REMOVAL

This event is logged when the user application aborts and the kernel driver tears down the GPU application running on the GPU. Control-C, GPU resets, sigkill are all examples where the application is aborted and this event is created.

In many cases, this is not indicative of a bug but rather a user or system action.

XID 48: DBE (DOUBLE BIT ERROR) ECC ERROR

This event is logged when the GPU detects that an uncorrectable error occurs on the GPU. This is also reported back to the user application. A GPU reset or node reboot is needed to clear this error.

The tool **nvidia-smi** can provide a summary of ECC errors. See “Tools That Provide Additional Information About Xid Errors”.

XID 63, 64: ECC PAGE RETIREMENT OR ROW REMAPPING

These events are logged when the GPU handles ECC memory errors on the GPU.

On GPUs that support row remapping, starting with [NVIDIA® Ampere architecture](#) GPUs, these events provide details on row remapper activity. For more information row remapper Xids, see:

<https://docs.nvidia.com/deploy/a100-gpu-mem-error-mgmt/index.html#row-mapping>

On earlier GPUs that support dynamic page retirement, these events provide details on dynamic page retirement activity. For more information on dynamic page retirement Xids, see:

<https://docs.nvidia.com/deploy/dynamic-page-retirement/index.html>

XID 74: NVLINK ERROR

This event is logged when the GPU detects that a problem with a connection from the GPU to another GPU or NVSwitch over NVLink. A GPU reset or node reboot is needed to clear this error.

This event may indicate a hardware failure with the link itself, or may indicate a problem with the device at the remote end of the link. For example, if a GPU fails, another GPU connected to it over NVLink may report an Xid 74 simply because the link went down as a result.

The **nvidia-smi nvlink** command can provide additional details on NVLink errors, and connection information on the links.

If this error is seen repeatedly and GPU reset or node reboot fails to clear the condition, contact your hardware vendor for support.

XID 79: GPU HAS FALLEN OFF THE BUS

This event is logged when the GPU driver attempts to access the GPU over its PCI Express connection and finds that the GPU is not accessible.

This event is often caused by hardware failures on the PCI Express link causing the GPU to be inaccessible due to the link being brought down. Reviewing system event logs and kernel PCI event logs may provide additional indications of the source of the link failures.

This event may also be caused by failing GPU hardware or other driver issues.

XID 93: NON-FATAL VIOLATION OF PROVISIONED INFOROM WEAR LIMIT

This event is logged when the GPU driver fails to update the infoROM due to violation of the provisioned InfoROM wear limit that was set for the GPU using NVFlash using `nvflash -elsessionstart`.

In most cases this is not indicative of a driver or flash failure, but rather the intentional use of the InfoROM wear protection feature as set by NVFlash.

XID 94, 95: CONTAINED/UNCONTAINED ECC ERRORS

These events are logged when GPU drivers handle ECC memory errors in GPUs that support ECC Error containment, starting with NVIDIA® A100 GPUs.

Detailed documentation for these Xids are covered in the A100 Memory error management manual:

<https://docs.nvidia.com/deploy/a100-gpu-mem-error-mgmt/index.html#row-mapping>

XID 110 SECURITY FAULT ERROR

This event should be uncommon unless there is a hardware failure. To recover, revert any recent system hardware modifications and cold reset the system. If this fails to correct the issue, contact your hardware vendor for assistance.

XID 119, 120: GSP RPC TIMEOUT / GSP ERROR

One or both of these events may be logged when an error occurs in code running on the GSP core of the GPU and/or a timeout occurs while waiting for the GSP core of the GPU to respond to an RPC message. A GPU reset or node power cycle may be needed if the error persists. If this problem reoccurs after a power cycle, follow the NVIDIA GPU Debug Guidelines document for additional debugging steps:
<https://docs.nvidia.com/deploy/gpu-debug-guidelines/index.html>

XID 121: C2C LINK CORRECTED ERROR

This event may occur when the GPU driver has observed corrected errors on the C2C NVLink connection to a Grace CPU. These errors are corrected by the system and have no operational impact. Resetting the GPU at an available service window will allow the GPU to retrain the link.

XID 140: ECC UNRECOVERED ERROR

This event may occur when the GPU driver has observed uncorrectable errors in GPU memory, in such a way as to interrupt the GPU driver's ability to mark the pages for dynamic page offlining or row remapping. Reset the GPU, and if the problem persists, contact your hardware vendor for support.

Notice

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.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

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

Copyright

© 2013-2024 NVIDIA Corporation and affiliates. All rights reserved.

