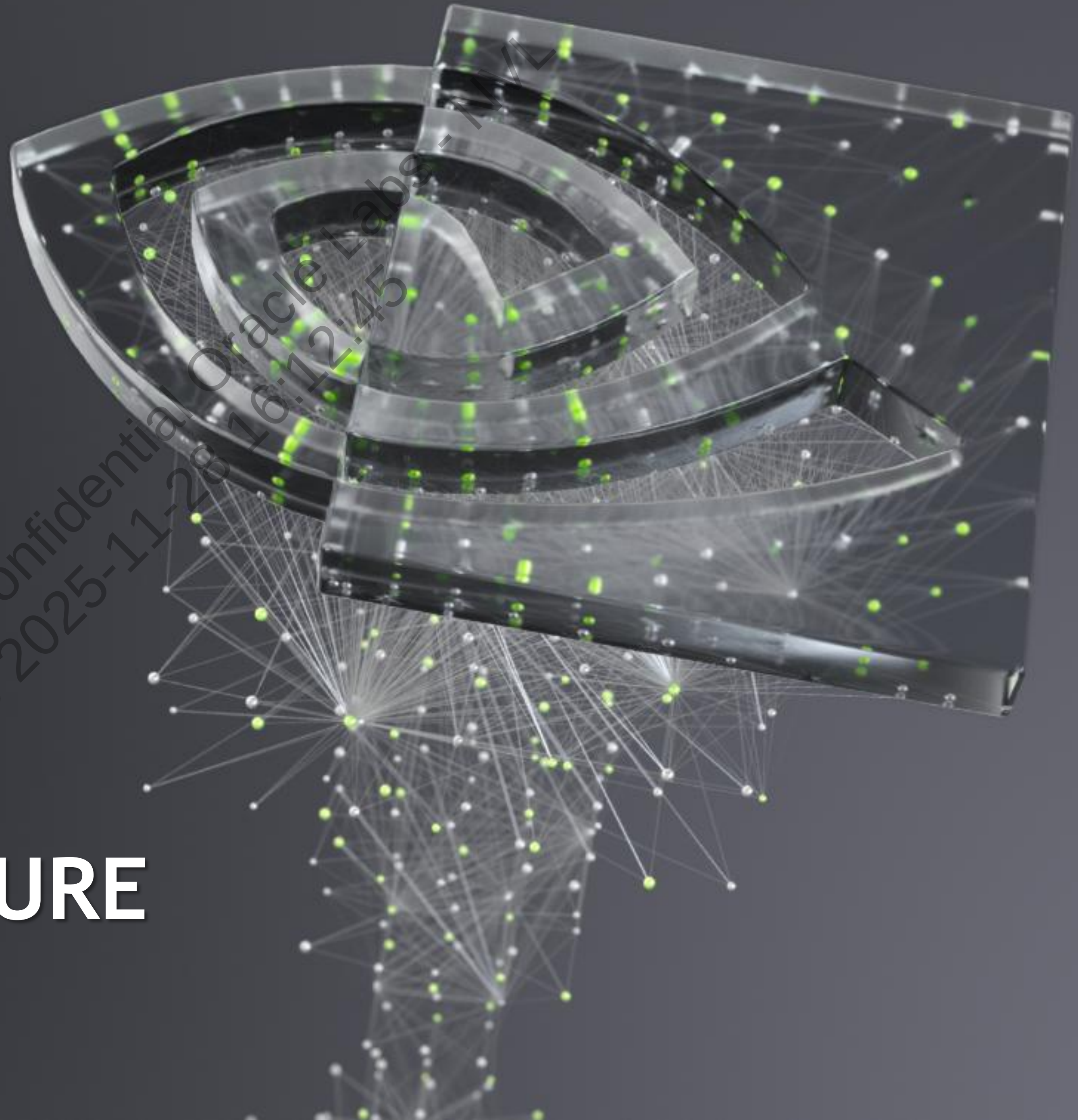


GPU RAS ARCHITECTURE

GPU HW & SW Resiliency Teams

NVIDIA CONFIDENTIAL Presentation

- All information in the following presentation is NVIDIA confidential, including codenames, future products, and performance projections
- No information in this presentation is allowed to be revealed or published without NVIDIA consent
- Sharing or distributing copies of this presentation to anyone is strictly prohibited. Use of cameras to capture information is strictly prohibited
- THIS INFORMATION IS INTENDED TO OUTLINE OUR GENERAL PRODUCT DIRECTION. MANY OF THE PRODUCTS AND FEATURES DESCRIBED HEREIN REMAIN IN VARIOUS STAGES AND WILL BE OFFERED ON A WHEN-AND-IF-AVAILABLE BASIS. THIS ROADMAP DOES NOT CONSTITUTE A COMMITMENT, PROMISE, OR LEGAL OBLIGATION AND IS SUBJECT TO CHANGE AT THE SOLE DISCRETION OF NVIDIA.
- THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITIES DESCRIBED FOR OUR PRODUCTS REMAINS AT THE SOLE DISCRETION OF NVIDIA. NVIDIA WILL HAVE NO LIABILITY FOR FAILURE TO DELIVER OR DELAY IN THE DELIVERY OF ANY OF THE PRODUCTS, FEATURES, OR FUNCTIONS SET FORTH IN THIS DOCUMENT.

OVERVIEW

A summary of GPU RAS Architecture

- RAM protections
- Error Containment
- HW availability features
- Interfaces: PCIe, NVLink, and Chip-to-Chip (C2C) error handling

Error Containment Definitions

- A hardware fault is contained at the origin if error indication (poison) is attached to all erroneous data leaving the source.
- A hardware fault is contained at an intermediate point or destination if the error indication is recognized in all possible fault paths and the destination halts to prevent the poison from spreading further.

GPU RAS FEATURE ROADMAP

RAM Protections: Ampere -> Hopper -> Blackwell -> Rubin

Unit Name	Ampere (A100)		Hopper (H100, H200)		Blackwell (B200, B300, GB200)		Rubin (R100)	
	Protection	Error Contain	Protection	Error Contain	Protection	Error Contain	Protection	Error Contain
HBM	ECC	✓	ECC*	✓	ECC*	✓	ECC**	✓
GCC (L 1.5)	Parity		Parity	✓	Parity	✓	Parity	✓
MMU/TLB	Parity		Parity	✓	Parity-Retry	✓	Parity-Retry	✓
PCIe	Parity/SEC-DED		Parity/SEC-DED	✓	Parity/SEC/SEC-DED	✓	Parity/SEC/SEC-DED	✓
NVLink	Parity/SEC-DED	✓	Parity/SEC-DED	✓	Parity/SEC-DED	✓	Parity/SEC-DED	✓
Register File	SEC-DED		SEC-DED	✓	SEC-DED	✓	SEC-DED	✓
L2 Request Coalescer					Parity		SEC-DED	
L2 Cache (Data)	SEC-DED	✓	SEC-DED	✓	SEC-DED	✓	SEC-DED	✓
L2 Cache (Tag)	Parity		Parity	✓	Parity	✓	SEC-DED	✓
L1 Cache (Instr)	Parity		Parity	✓	Parity	✓	Parity	✓
L1 Cache (Data)	SEC-DED		SEC-DED	✓	SEC-DED	✓	SEC-DED	✓
Copy engine, CXL, FB					SEC/SEC-DED	✓	SEC/SEC-DED	✓
L2 state					Parity	✓	Parity/SEC-DED	✓
Microcontrollers	Some Parity/SEC-DED		Some Parity/SEC-DED		Parity	✓	Parity	✓

SRAMs that can generate Uncorrectable Errors (UCE) designate affected data with poison as shown in Error Contain column

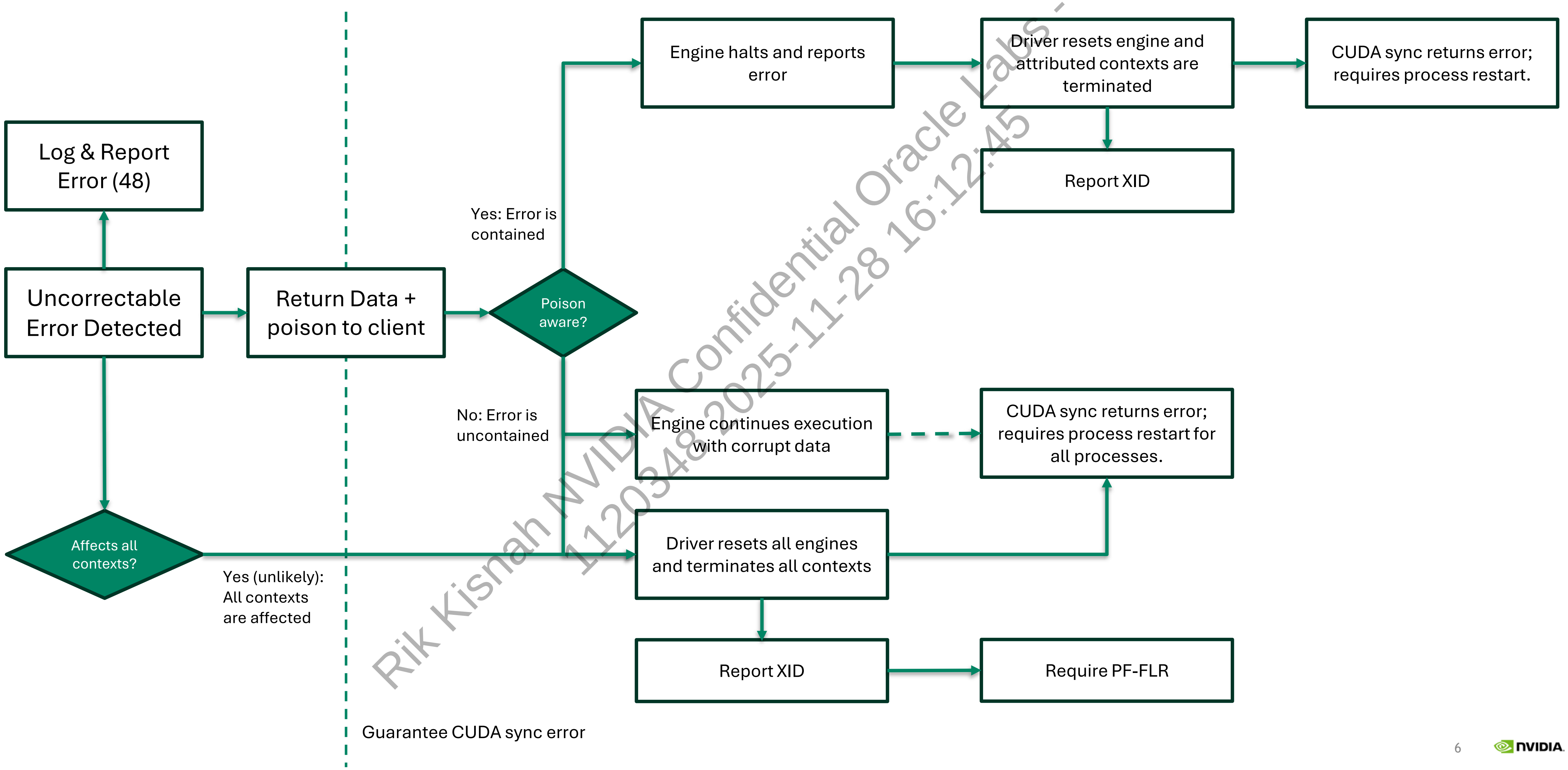
SEC: Single-bit Error Correction; DED: Double-bit Error Detection
**: HBM3/3E - DRAM On-Die ECC + Sideband ECC; **: HBM4 - adds TBD RAS support*

HARDWARE ERROR CONTAINMENT

Error Containment by Consumer of Poisoned Data

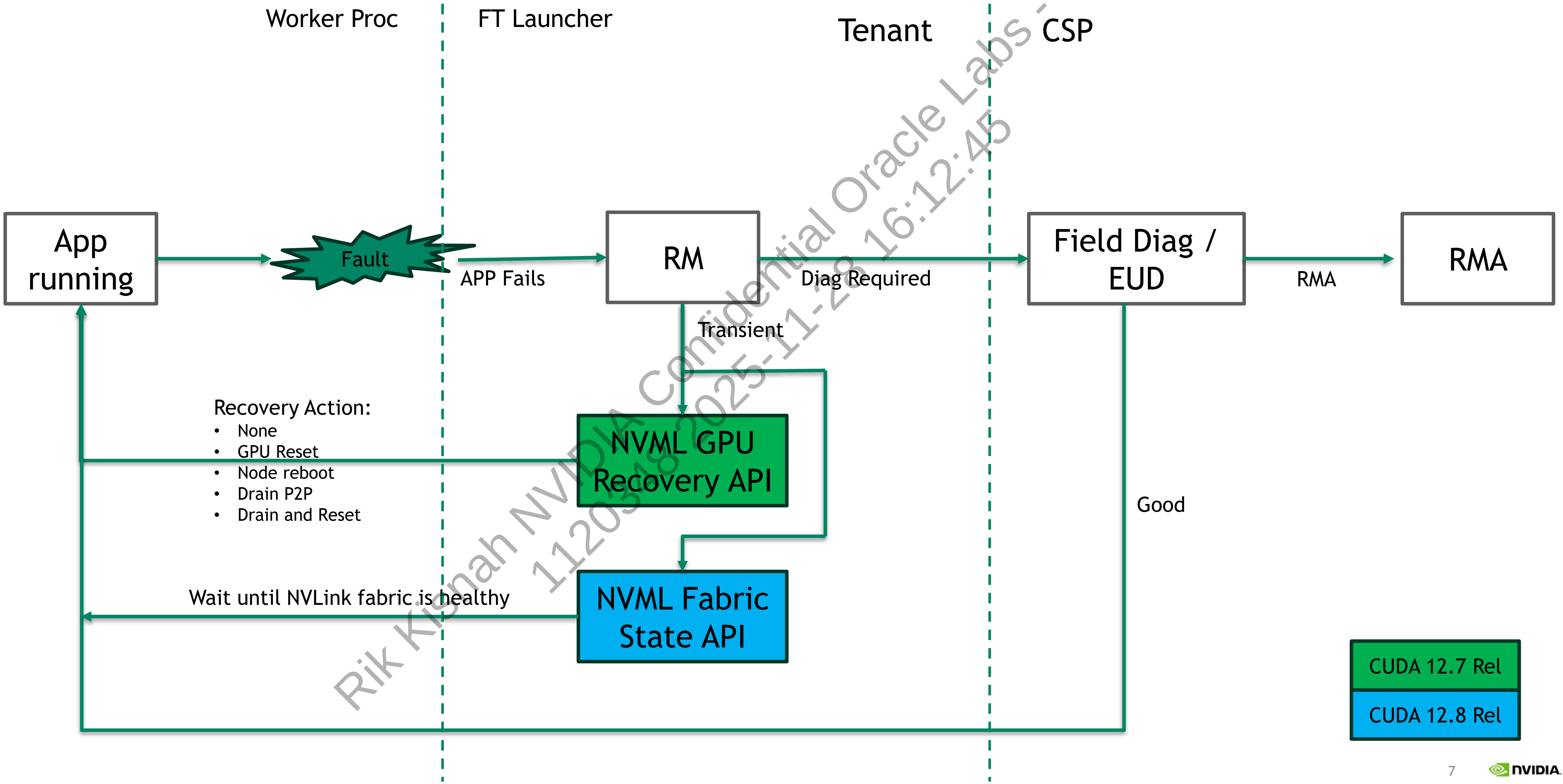
	Ampere (A100)	Hopper (Hx00)	Blackwell& Rubin (Bx00/Rx00)
Streaming Multiprocessor	✓	✓	✓
Copy Engine	✓	✓	✓
Memory Mgmt. Unit Fill		✓	✓
Frame Buffer	✓	✓	✓
Microcontrollers (OFA/SEC/GSP/FSP/PMU)			✓
Context Switch Engine (FECS)			✓
Video Engines (Decoder/JPEG)			✓

SOFTWARE ERROR CONTAINMENT



Log & Report Error (48)
Uncorrectable Error Detected
Return Data + poison to client
Poison aware?
Yes: Error is contained
Engine halts and reports error
Driver resets engine and attributed contexts are terminated
CUDA sync returns error; requires process restart.
Report XID
No: Error is uncontained
Engine continues execution with corrupt data
CUDA sync returns error; requires process restart for all processes.
Affects all contexts?
Yes (unlikely): All contexts are affected
Driver resets all engines and terminates all contexts
Report XID
Require PF-FLR
Guarantee CUDA sync error

SOFTWARE ERROR RECOVERY



Worker Proc

FT Launcher

Tenant

CSP

App running

Fault

APP Fails

RM

Diag Required

Field Diag / EUD

RMA

RMA

Transient

Recovery Action:

- None
- GPU Reset
- Node reboot
- Drain P2P
- Drain and Reset

NVML GPU Recovery API

Wait until NVLink fabric is healthy

NVML Fabric State API

Good

CUDA 12.7 Rel

CUDA 12.8 Rel

HW AVAILABILITY FEATURES

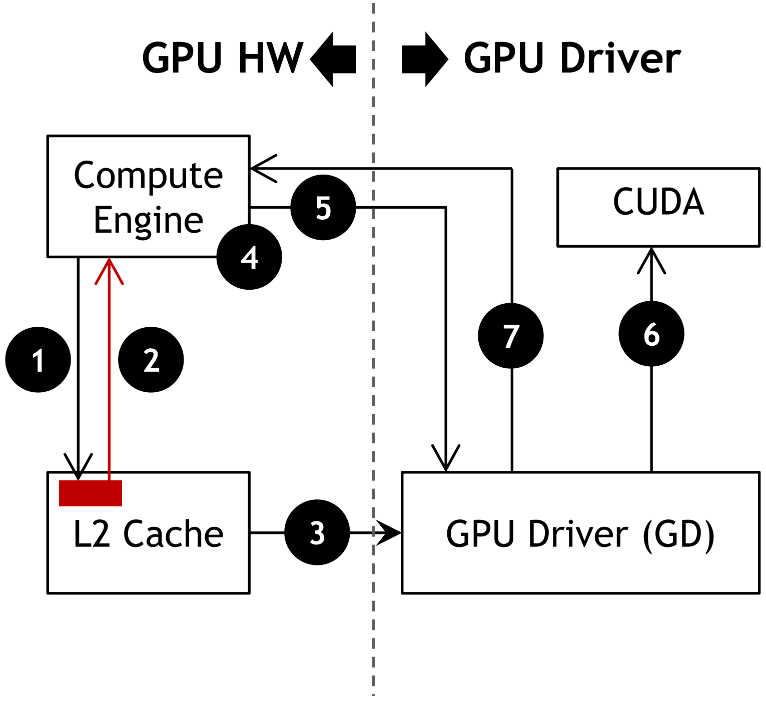
AVAILABILITY FEATURES

- Goal: Improve up-time by minimizing fatal errors
  - Non-fatal errors terminate impacted application while avoiding the need for GPU HW reset

Technique	Summary
Error Containment	▪ Data poisoning for error attribution to the consuming process ▪ HW executing the process halts to prevent error leak ▪ An error code is returned to caller CUDA application (on CPU) ▪ HW engine reset without requiring GPU reset ▪ HW support for fine-grain local checkpointing
Graceful Degradation	▪ Continue HW operation with degraded capacity with full HW functionality ▪ Example: HBM dynamic page retirement on Detectable and Uncorrectable Errors (DUE)
Self-recovery	▪ Retry transactions that can recover from transient failures ▪ Example: Invalidate TLB entry on parity error and do a page walk

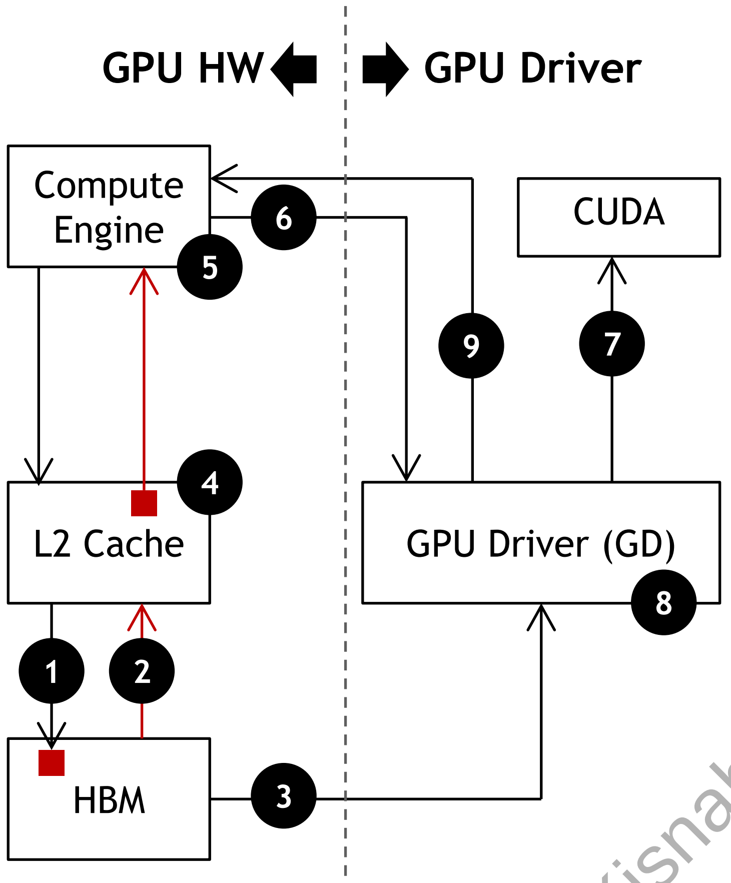
AVAILABILITY: ERROR CONTAINMENT

ERROR CONTAINMENT: L2 CACHE READ EXAMPLE



Step	Actions
1	A memory read accesses hits in L2 cache, data has DUE
2	▪ L2 cache returns the data as poisoned ▪ L2 cache tags the data as poisoned for internal storage ▪ Future L2 hits return data with poison ▪ L2 eviction will write poisoned data to HBM
3	L2 Cache notifies the GD about the DUE. GD reports XID-172 (w/ 48)
4	Execution halts on receiving poison. No new memory accesses are issued
5	Engine notifies GD after all outstanding memory responses are drained. GD reports XID-94
6	GD halts further execution and notifies an error to CUDA, stalled engine is reset and GD reports XID-94.
7	Cuda sets sticky error state and enforces a process restart

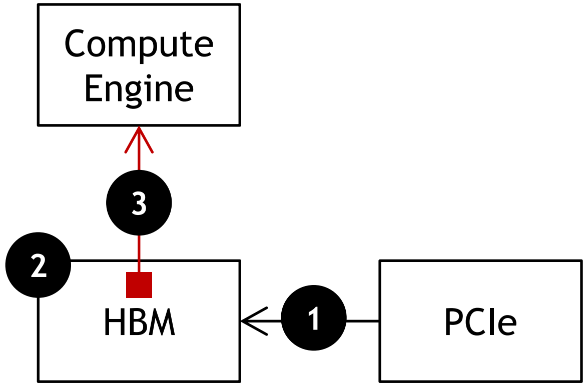
ERROR CONTAINMENT: MEMORY READ EXAMPLE



Step	Actions
1	A memory read accesses data with DUE
2	Data is returned as-is with inline poison indicator
3	HBM reports DUE to GD. GD reports XID-63
4	▪ L2 cache stores the data as poisoned and forwards poisoned data ▪ Future L2 hits return data with poison ▪ L2 eviction will write poisoned data to HBM
5	Execution halts on receiving poison. No new memory accesses are issued
6	Engine notifies GD after all outstanding memory responses are drained.
7	GD halts further execution and notifies an error to CUDA, stalled engine is reset and GD reports XID-94
8	GD removes the faulty page from future allocation
9	Cuda sets sticky error state and enforces a process restart

End Result*: CUDA aborts with an error and faulty page is removed from allocation until next GPU reset. Faulty page is row-remapped during next GPU reset. HW does not have to be offlined immediately*

ERROR CONTAINMENT: MEMORY WRITE EXAMPLE

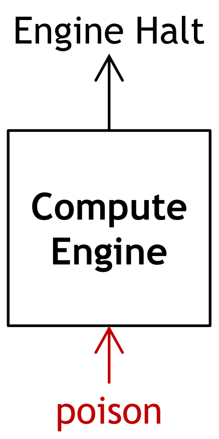


Step	Actions
1	▪ GPU receives a PCIe downstream write request with EP bit set to 1 ▪ GPU can optionally report poison in downstream data packet
2	▪ HBM stores the line as poisoned. No error is reported to RM
3	▪ A future reader of poisoned data will find a DUE. ▪ Memory read example has details on poisoned data response

Rules for Handling Poisoned Writes

- A poisoned write request will store the data as poisoned (in L2 cache and HBM)
- A partial write request will be stored as poisoned data if either the written data or stored data is poisoned
- A fully covered write request with no poison will clear poisoned state of the stored data
- An atomic request will store the result as poisoned data if either the request payload or stored data is poisoned. Poisoned data will be returned, if applicable (depending on the atomic instruction)

UNIT SPECIFIC POISON HANDLING

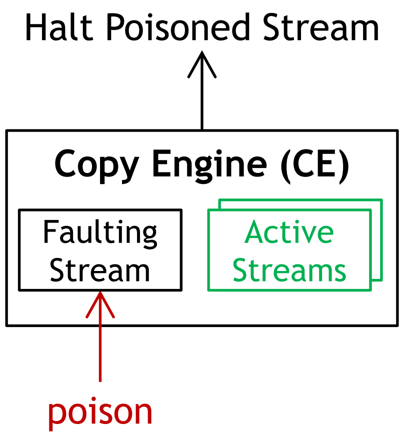


Engine Halt

Compute Engine

poison

- See memory read example



Halt Poisoned Stream

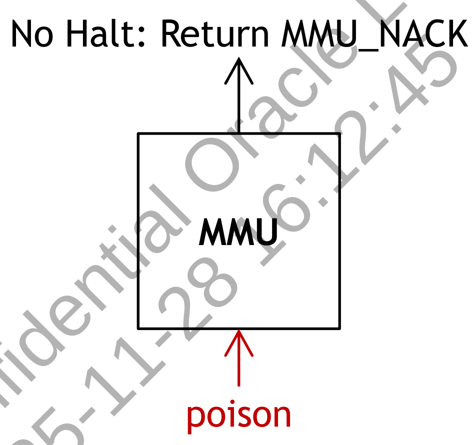
Copy Engine (CE)

Faulting Stream

Active Streams

poison

- CE has multiple DMA streams
- Only the stream(s) receiving poison halt
- Poisoned data sourced by CE is not copied to the destination

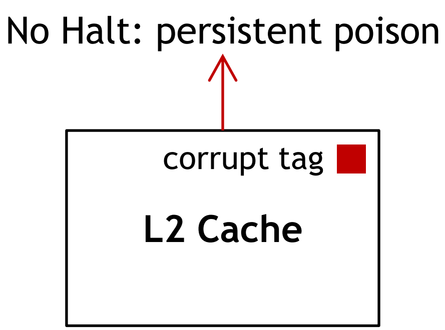


No Halt: Return MMU_NACK

MMU

poison

- MMU may see poisoned page tables during page walk
- MMU_NACK is returned to the unit requesting page translation
- Treated like any translation error inside the requesting unit



No Halt: persistent poison

corrupt tag 🟥

L2 Cache

- Address information is lost when L2 tag is corrupt
- Memory state of GPU is no longer reliable
- L2 returns persistent poison for all future read requests
- GPU reset is required for recovery

XID INDICATIONS

XID	Likely cause
48	Indicates UCE in SRAM/DRAM. Details in XID string.
63	Row remap pending due to HBM UCE
64	Row remap either already pending or failed due to HBM UCE
94	Poison consumed & hardware contained
95	Hardware uncontained due poison unaware consumption. Containment provided via SW mechanism.
154 (new)	GPU recovery action changed: Specifies action needed to clear the effects from a previous fault and get GPU back to operational state
156 (new)	Resource retirement event (TPC repair)
157 (new)	Resource retirement event failure (TPC repair)
160 (new)	Resource retirement event (HBM memory channel repair)
161 (new)	Resource retirement event failure (no HBM memory channel repair available) – generally observed in conjunction with XID-64
171 (new)	Indicates UCE in DRAM. Observed in conjunction with a corresponding XID-48.
172 (new)	Indicates UCE in SRAM. Observed in conjunction with a corresponding XID-48.

HBM RAS

HBM MEMORY RESILIENCE

RAS Features for DRAM

	Ampere	Hopper	Blackwell	Rubin
Memory cell/ row failures	Page Retirement	Row Remapping	Row Remapping	Row Remapping
Large scale failures		Channel sparing (H200)	Channel sparing	Channel sparing; Bank Sparing

HBM MEMORY RESILIENCE

Error Correction and Detection capabilities

	HBM2E	HBM3	HBM3E	HBM4
GPU generation(s)	Ampere, Hopper	Hopper	Blackwell, Hopper	Rubin
DRAM (OD-ECC)	None	272/32, Reed-Solomon SSC	272/32, Reed-Solomon SSC	272/32, Reed-Solomon SSC
E2E (Sideband)	64/8, SEC-DED	256/16, SEC-DED or CRC	256/16, SEC-DED or CRC	256/16, SEC-DED or CRC

IN FIELD REPAIR

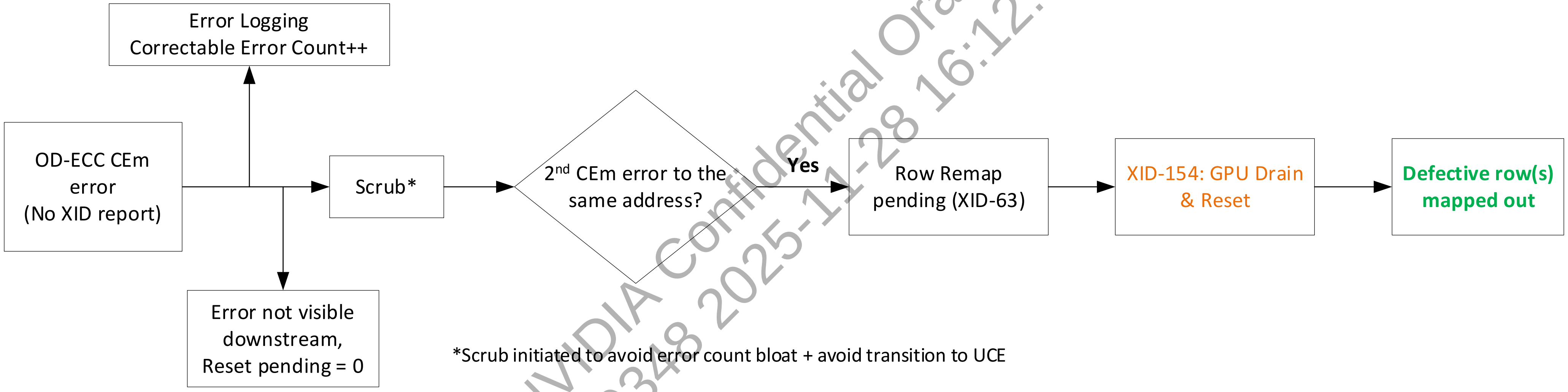
Runtime Memory (HBM and GDDR) repair

- https://docs.nvidia.com/deploy/a100-gpu-mem-error-mgmt/index.html
- Policies that exercise these repair mechanisms are explained in the following slides

Repair Mechanism	Environment	Condition and Symptom	Root Cause	Applicability
Row remapping (driver)	**Runtime**, reported via driver Memory ECC: Correctable/Uncorrectable errors	Correctable errors (No XID) or XID-171 (w/ 48) Uncorrectable ECC Errors, and then either: - XID 63 (Page Retirement Event) - XID 64 (Page Retirement Failure)	HBM/GDDR Errors	HBM: Hopper, Blackwell, Rubin GDDR: Ampere, Ada, Blackwell (and later)
Channel repair (driver)	**Runtime**, reported via driver Memory ECC: Uncorrectable errors	XID-171 (w/ 48) Uncorrectable ECC Errors, and then either: - XID 160 (Resource Retirement Event) - XID 161 (Resource Retirement Failure)	HBM Errors	Blackwell (Initial Support): 97.00.7D.00.00 CUDA 12.9 (R575) / CUDA 12.8 (R570 TRD3)*

* FW >= 97.00.D9.00.29 (B200)/97.10.4C.00.06 (B300)/91.00B9.00.2C (GB200)/ 97.10.4A.00.05 (GB300) critical for channel repair proper functionality

HBM ERRORS – DRAM ECC - CORRECTABLE

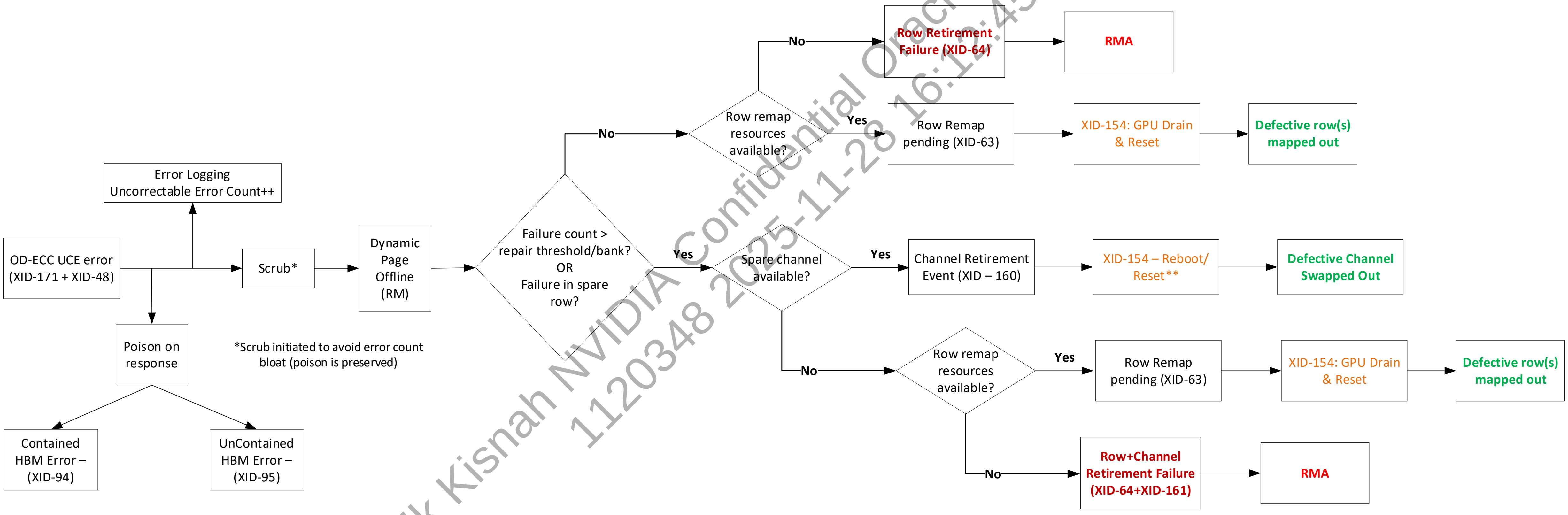

Error Logging
Correctable Error Count++
OD-ECC CEm error (No XID report)
Scrub*
2nd CEm error to the same address?
Yes
Row Remap pending (XID-63)
XID-154: GPU Drain & Reset
Defective row(s) mapped out
Error not visible downstream, Reset pending = 0

*Scrub initiated to avoid error count bloat + avoid transition to UCE

Interrupt Storm condition (XID-92 emitted): If we exceed a rate of interrupts in a short period of time, CE interrupt is disabled until the next driver load

HBM ERRORS – DRAM ECC - UNCORRECTABLE

**Present firmware requires Node Reboot for channel repair, but future firmware versions will be optimized to limit the blast radius with GPU Bus Reset



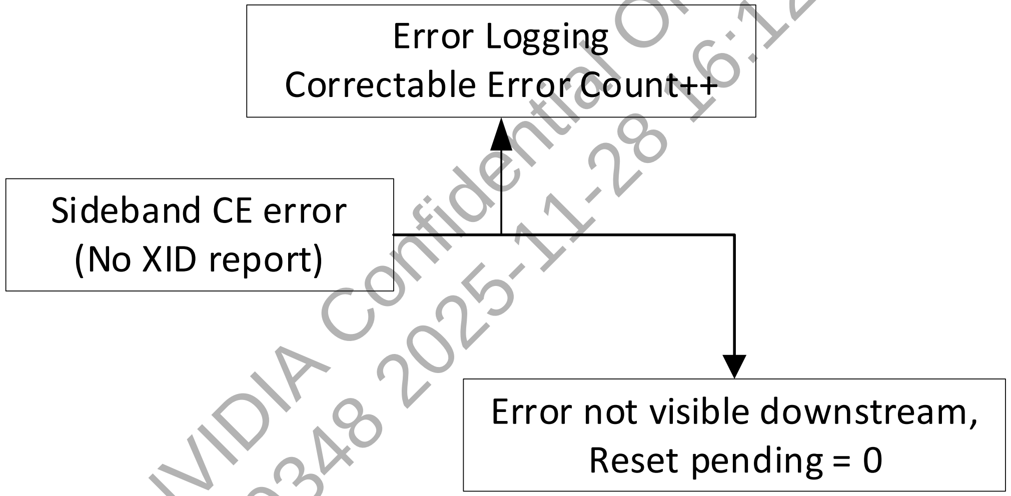
OD-ECC UCE error (XID-171 + XID-48)
Error Logging Uncorrectable Error Count++
Scrub*
Dynamic Page Offline (RM)
Poison on response
Contained HBM Error – (XID-94)
UnContained HBM Error – (XID-95)
Failure count > repair threshold/bank? OR Failure in spare row?
No
Row remap resources available?
No
Row Retirement Failure (XID-64)
RMA
Yes
Row Remap pending (XID-63)
XID-154: GPU Drain & Reset
Defective row(s) mapped out
Yes
Spare channel available?
Yes
Channel Retirement Event (XID – 160)
XID-154 – Reboot/ Reset**
Defective Channel Swapped Out
No
Row remap resources available?
Yes
Row Remap pending (XID-63)
XID-154: GPU Drain & Reset
Defective row(s) mapped out
No
Row+Channel Retirement Failure (XID-64+XID-161)
RMA

*Scrub initiated to avoid error count bloat (poison is preserved)

** XID-154 currently asks for GPU reset, but reboot is needed currently for channel repair to take effect

HBM ERRORS – SIDEBAND - CORRECTABLE

(Before CUDA 13.1 (RM 590), OD-ECC Correctable policy is used)



Error Logging
Correctable Error Count++

Sideband CE error
(No XID report)

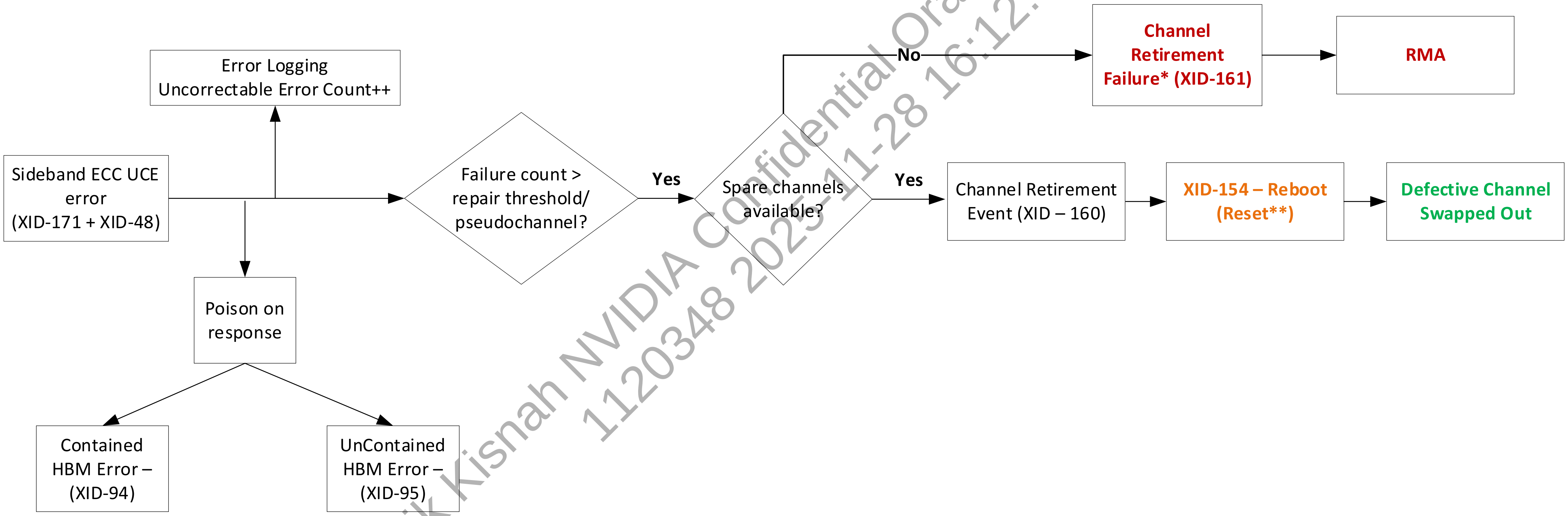
Error not visible downstream,
Reset pending = 0

Interrupt Storm condition (XID-92 emitted): If we exceed a rate of interrupts in a short period of time, CE interrupt is disabled until the next driver load

HBM ERRORS – SIDEBAND - UNCORRECTABLE

(Before CUDA 13.1 (RM 590), OD-ECC Uncorrectable policy is used)

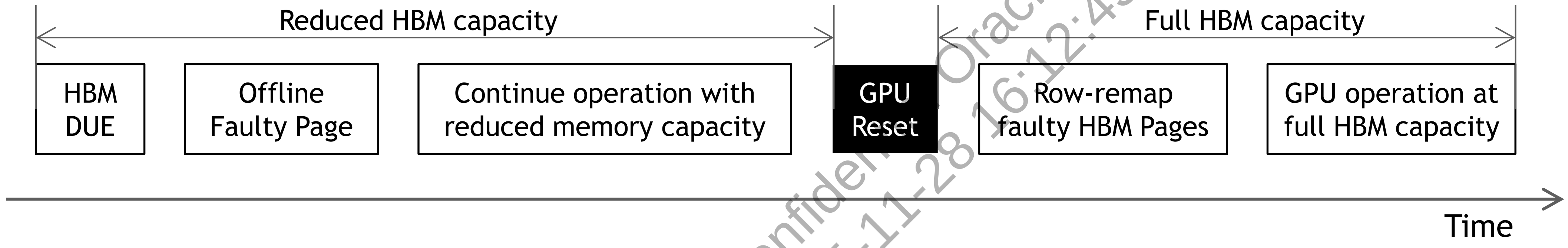
**Present firmware requires Node Reboot for channel repair, but future firmware versions will be optimized to limit the blast radius with GPU Bus Reset


Sideband ECC UCE error (XID-171 + XID-48)
Error Logging Uncorrectable Error Count++
Failure count > repair threshold/ pseudochannel?
Yes
Spare channels available?
No
Channel Retirement Failure* (XID-161)
RMA
Yes
Channel Retirement Event (XID – 160)
XID-154 – Reboot (Reset**)
Defective Channel Swapped Out
Poison on response
Contained HBM Error – (XID-94)
UnContained HBM Error – (XID-95)

** XID-154 currently asks for GPU reset, but reboot is needed currently for channel repair to take effect

HANDLING HBM ERRORS

Graceful degradation when HBM Detectable and Uncorrectable Errors (DUE) are encountered. Applies to transient and permanent errors


Reduced HBM capacity
Full HBM capacity
HBM DUE
Offline Faulty Page
Continue operation with reduced memory capacity
GPU Reset
Row-remap faulty HBM Pages
GPU operation at full HBM capacity
Time

	Page Offlining	Row-remapping
Trigger	HBM Detectable & Uncorrectable Error (DUE)	GPU reset with one or more faulty HBM pages
Actions	Remove faulty pages from memory allocation	Replace faulty pages with spare pages, accessible at the same physical address
Implications	▪ Continued operation at reduced memory capacity ▪ Holes in physical memory	▪ Full memory capacity ▪ Holes filled to provide contiguous physical memory
Spare limit	▪ Max 8 row-remaps per HBM bank or 512 row-remaps per GPU ▪ RMA when row-remapper capacity is reached	

SRAM ERROR HANDLING

SRAM & DRAM UCE

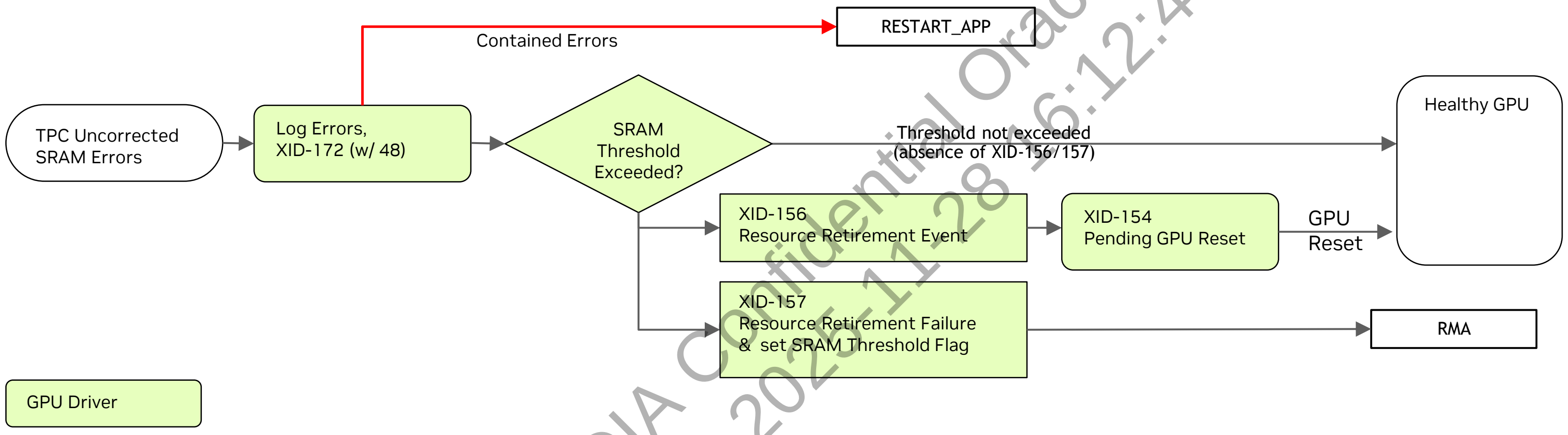
Recovery Action

For tenant/CSP, upon any DRAM UCE (XID 48 w/ 171) or SRAM UCE (XID 48 w/ 172):

1. Check the “SRAM Threshold Exceeded” flag from nvidia-smi. If that is set, RUN_FIELDDIAGS.
2. Check if XID-157 (TPC Retirement Failure), or XID-161 (Channel Retirement Failure), or XID-64 (DRAM Retirement Failure) are emitted. If any, RUN_FIELDDIAGS.
3. Otherwise, if XID-154 (GPU_RECOVERY_ACTION_CHANGED) is emitted, follow its suggestion.
4. Otherwise, RESTART_APP if that is killed.
5. Otherwise, IGNORE.

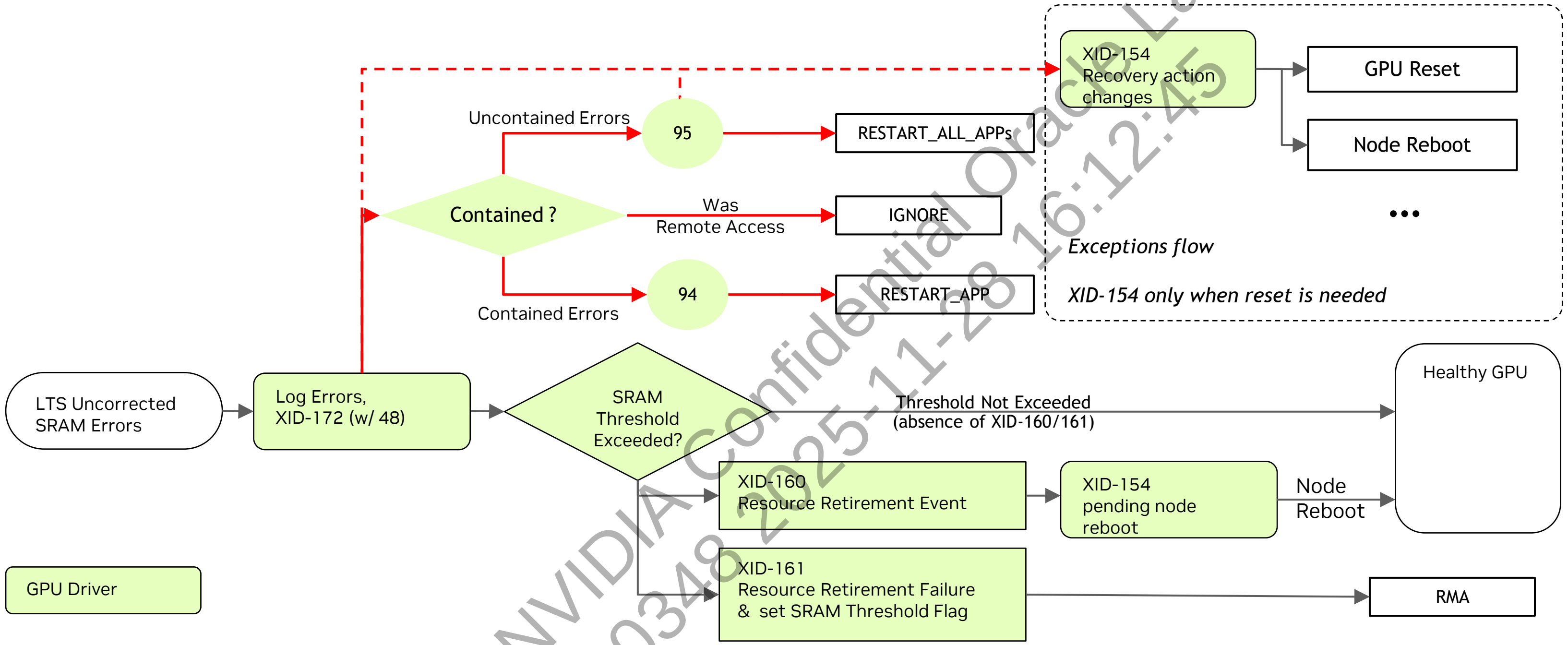
- *Notes on Item 4: When an SRAM UCE (XID 48 with 172) occurs outside the TPC and leads to an application crash, restarting the application resolves the issue in the majority of cases. Nonetheless, the tenant or CSP may choose to perform a GPU reset to ensure that all correlated or latent faults are fully cleared and to accelerate the increment of the SRAM threshold counter.*

SRAM UCE: TPC


Contained Errors
RESTART_APP
TPC Uncorrected SRAM Errors
Log Errors, XID-172 (w/ 48)
SRAM Threshold Exceeded?
Threshold not exceeded (absence of XID-156/157)
Healthy GPU
XID-156 Resource Retirement Event
XID-154 Pending GPU Reset
GPU Reset
XID-157 Resource Retirement Failure & set SRAM Threshold Flag
RMA
GPU Driver

Environment	Symptom	Root Cause	Repair	Applicability
Runtime, reported via Driver - TPC ECC UCEs	XID-172 (w/ 48) DBE ECC Errors, and then either: - XID 156 (Resource Retirement Event) - XID 157 (Resource Retirement Failure)	TPC SRAM Errors	TPC repair (driver)	TPC Ready: CUDA 12.8 (R570) Hopper: 96.00.8A.00.00 Blackwell: 97.00.40.00.00

SRAM UCE: LTS

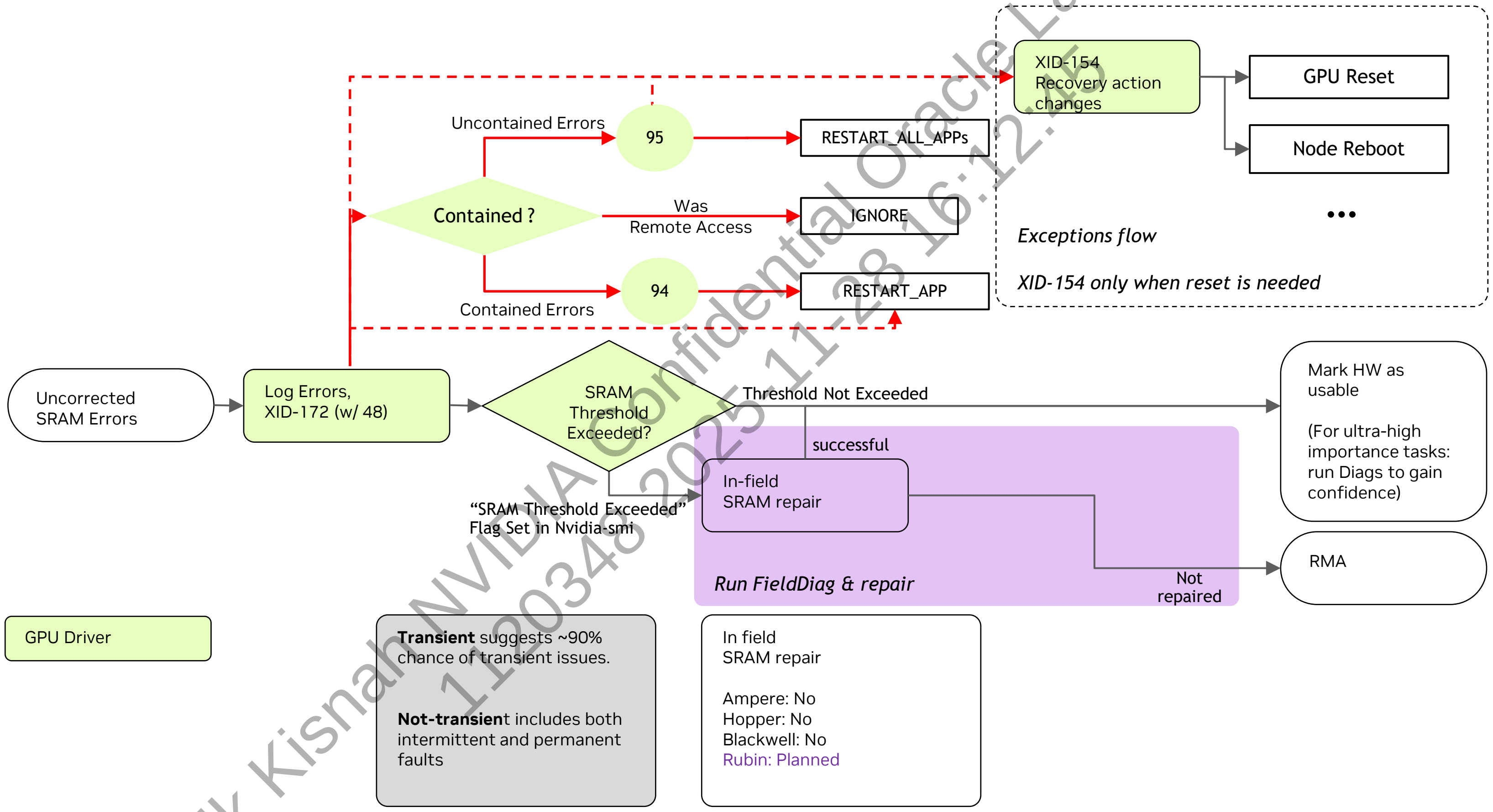


LTS Uncorrected SRAM Errors
Log Errors, XID-172 (w/ 48)
Contained ?
Uncontained Errors
95
RESTART_ALL_APPs
Was Remote Access
IGNORE
Contained Errors
94
RESTART_APP
XID-154 Recovery action changes
GPU Reset
Node Reboot
•••
Exceptions flow
XID-154 only when reset is needed
SRAM Threshold Exceeded?
Threshold Not Exceeded (absence of XID-160/161)
Healthy GPU
XID-160 Resource Retirement Event
XID-154 pending node reboot
Node Reboot
XID-161 Resource Retirement Failure & set SRAM Threshold Flag
RMA
GPU Driver

Environment	Symptom	Root Cause	Repair	Applicability
Runtime, reported via driver - WIP LTS ECC UCEs	XID-172 (w/ 48) DBE ECC Errors, and then either: - XID 160 (Resource Retirement Event) - XID 161 (Resource Retirement Failure)	LTS SRAM Errors	Channel repair (driver)	WIP LTS ECC UCEs: targets CUDA 13.1

- Node reboot for Blackwell after XID-160 is needed.
- Before 13.1, once threshold is met, the SRAM threshold flag will be set, w/o attempting to repair. (no XID 160 or 161)

SRAM UCE: OUTSIDE OF TPC/LTS



Uncorrected SRAM Errors

Log Errors, XID-172 (w/ 48)

Contained ?

Uncontained Errors

95

RESTART_ALL_APPs

Was Remote Access

IGNORE

Contained Errors

94

RESTART_APP

XID-154 Recovery action changes

GPU Reset

Node Reboot

•••

Exceptions flow

XID-154 only when reset is needed

SRAM Threshold Exceeded?

Threshold Not Exceeded

Mark HW as usable

(For ultra-high importance tasks: run Diags to gain confidence)

"SRAM Threshold Exceeded" Flag Set in Nvidia-smi

successful

In-field SRAM repair

Run FieldDiag & repair

Not repaired

RMA

GPU Driver

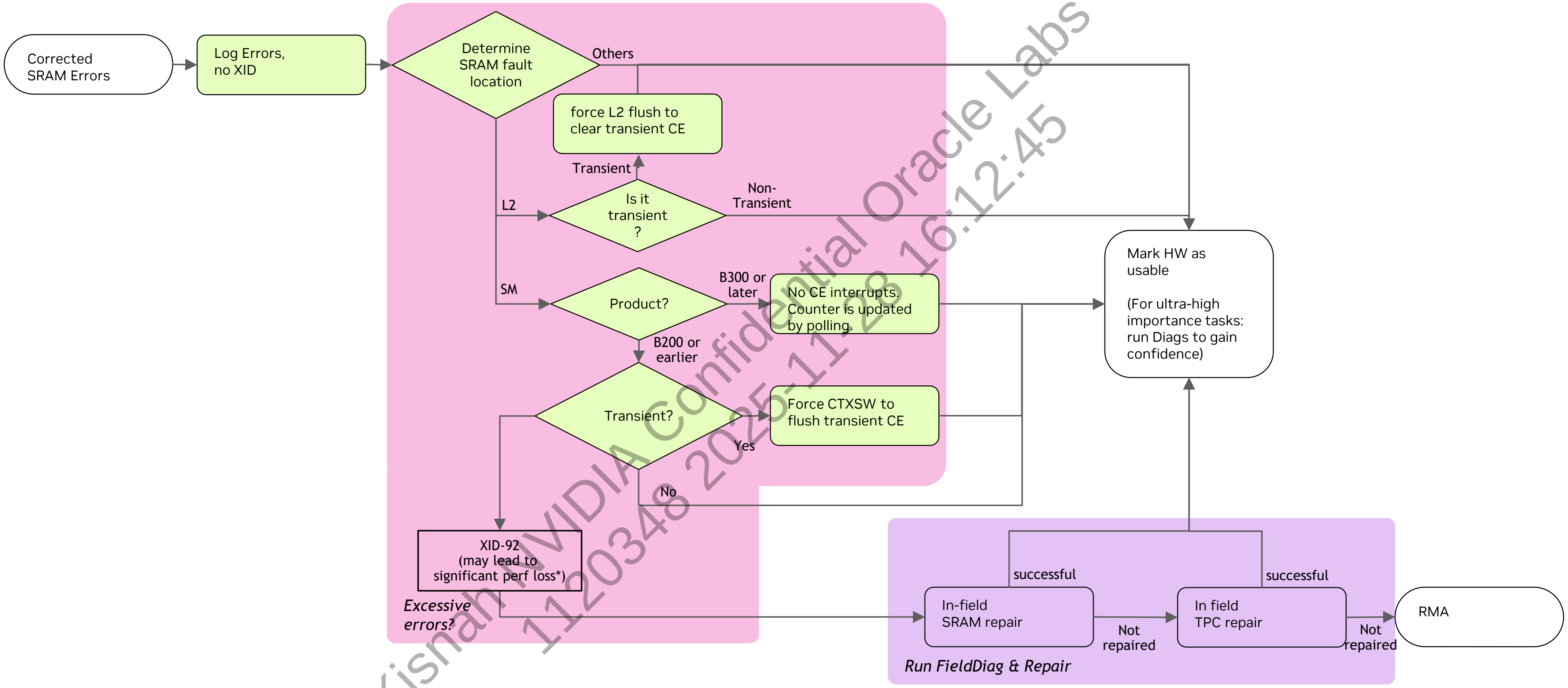
Transient suggests ~90% chance of transient issues.

Not-transient includes both intermittent and permanent faults

In field SRAM repair

Ampere: No
Hopper: No
Blackwell: No
Rubin: Planned

SRAM ERRORS - CORRECTED



Corrected SRAM Errors

Log Errors, no XID

Determine SRAM fault location

Others

force L2 flush to clear transient CE

Transient

L2

Is it transient ?

Non-Transient

SM

Product?

B300 or later

No CE interrupts, Counter is updated by polling.

B200 or earlier

Transient?

Yes

Force CTXSW to flush transient CE

No

XID-92 (may lead to significant perf loss*)

Excessive errors?

Mark HW as usable

(For ultra-high importance tasks: run Diags to gain confidence)

successful

successful

In-field SRAM repair

Not repaired

In field TPC repair

Not repaired

RMA

Run FieldDiag & Repair

GPU Driver

Significant perf loss*

Significant perf loss can only be possible if the error rate is >>100s / min.
So Nvidia suggest to ignore errors at a rate of <100 / min

Transient suggests ~90% chance of transient issues.

Non-transient includes both intermittent and permanent faults

In field
SRAM repair

Ampere: No
Hopper: No
Blackwell: No
Rubin: Planned

In field
TPC repair

Ampere: No
Hopper: Yes
Blackwell: Yes
Rubin: Planned

SRAM ERRORS - CORRECTED CHANGES IN B300

Intermittent or permanent faults caused SRAM ECC Corrected Errors (CE) from SM: NO functional impacts.

Repeated CEs cause interrupt storms, impacts performance:

- **Issue in B200 and prior generations:**
  - **GPU processing is stalled when CE interrupt are pending, need GPU driver to handle**
  - **Error rate of >100 counts / mins may cause performance degradation. Guidance to customers:** https://apps.nvidia.com/PID/ContentLibraries/Detail?id=1116581
- **Fix in B300 and later: No SM CE interrupts. No performance impact.**
  - Achieved by permanently masking those CE interrupts. GPU driver will still poll hardware error counters registers

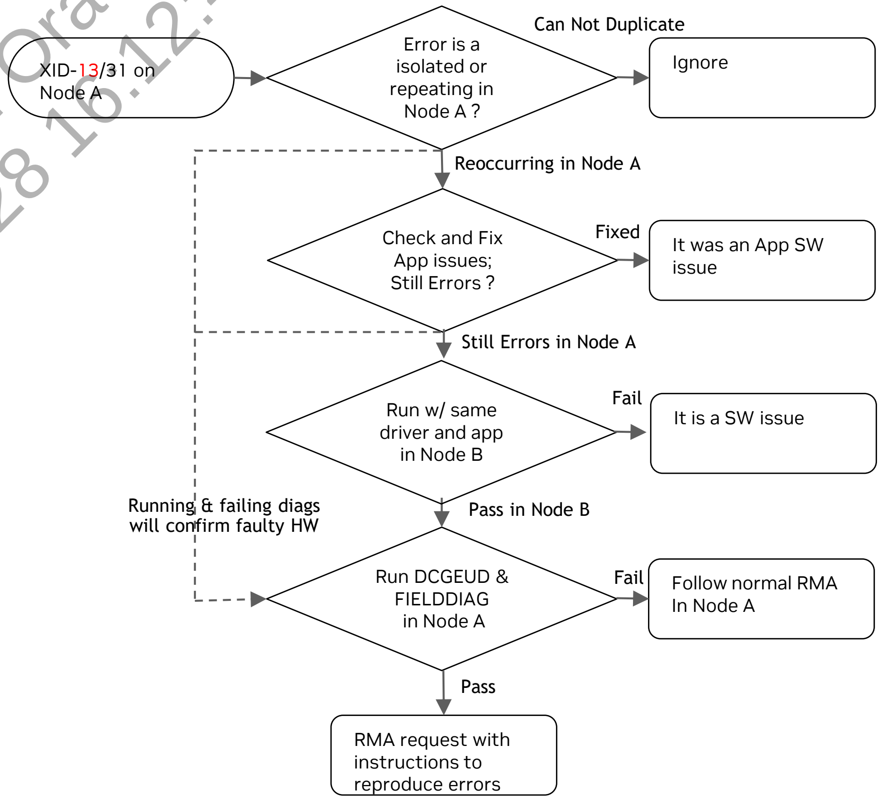
GPU MMU ERROR HANDLING

GPU MMU FAULTS

Error type	Reported Errors	Policies
MMU ECC UnCorr Error - HW-recovered - (HW issue in MMU)	No Xid	No recovery action from Driver/SW - GPU MMU HW automatically retried, the faulty location is added to the denylist (feature added since GH100)
MMU ECC UnCorr Error - HW-unrecovered - (HW issue in MMU)	Xid-172 (w/ 48) Uncorrectable DRAM ECC error	- MMU retry could not resolve (since unique error counter has exceeded the limit)
MMU received poison - (HW issue outside of MMU)	Xid-94 Contained ECC error FAULT_TYPE_POISONED	- Follow the playbook of XID-172 from other units to root cause the issue.
MMU page faults - HW/SW caused - (HW/SW issues in/outside of MMU)	Xid-31 GPU Memory Page Fault FAULT_TYPE_PDE FAULT_TYPE_PTE FAULT_TYPE_VA_LIMIT_VIOLATION FAULT_TYPE_PRIV_VIOLATION FAULT_TYPE_RO_VIOLATION FAULT_TYPE_WO_VIOLATION FAULT_TYPE_PITCH_MASK_VIOLATION FAULT_TYPE_WORK_CREATION FAULT_TYPE_UNSUPPORTED_APERTURE FAULT_TYPE_CC_VIOLATION FAULT_TYPE_UNSUPPORTED_KIND FAULT_TYPE_REGION_VIOLATION FAULT_TYPE_ATOMIC_VIOLATION FAULT_TYPE_UNBOUND_INST_BLOCK (no chance for app to cause this issue, but driver may)	The error may stem from hardware or software.

MMU WORKFLOWS

- WORKFLOW_XID_13
  - Repeat TPC and GPC, diff SMs: RUN_DCGMEUD (possible HW issue); if pass RUN_FIELDDIAGS
  - Repeat TPC and GPC, single SM: RUN_DCGMEUD (possible HW issue); if pass RUN_FIELDDIAGS
  - Solo, no burst: INVESTIGATE_APP
  - Not Repeat TPC and GPC: INVESTIGATE_APP
  - Non-prod environment: INVESTIGATE_APP
  - If known good APP and Solo: REPORT_ISSUE
- WORKFLOW_XID_31
  - ○ Multiple runs needed to establish pattern
  - ○ Repeat MMU faults to same GPU (via PCI-ID): RUN_DCGMEUD (possible HW issue); if pass RUN_FIELDDIAGS
  - ○ Repeat MMU faults to diff GPU (via PCI-ID): INVESTIGATE_APP
  - ○ Solo, no burst: INVESTIGATE_APP
  - ○ If known good APP: REPORT_ISSUE


XID-13/31 on Node A
Error is a isolated or repeating in Node A ?
Can Not Duplicate
Ignore
Reoccurring in Node A
Check and Fix App issues; Still Errors ?
Fixed
It was an App SW issue
Still Errors in Node A
Run w/ same driver and app in Node B
Fail
It is a SW issue
Pass in Node B
Running & failing diags will confirm faulty HW
Run DCGEUD & FIELDDIAG in Node A
Fail
Follow normal RMA In Node A
Pass
RMA request with instructions to reproduce errors

AVAILABILITY: SELF-RECOVERY

SELF-RECOVERY

L1 caches and TLBs self-invalidate an entry with parity error and re-fetch from memory

	L1 Data Cache	TLB
Trigger	ECC or parity error, in either data or tag arrays	Parity error in either looked-up or translated address
Actions	Invalidate the entry, and treat it as miss	
Recovery	▪ Persistent retries at the same structure are indicative of permanent faults ▪ SW should mark the part as faulty under this condition, HW continues to retry	

PCIE GEN6 (BLACKWELL+)

SUMMARY OF PCIE RAS

- PCIe Gen6 (Blackwell+) with Advanced Error Reporting (AER)
  - Supports FEC, LCRC and ECRC
  - Poison forwarding in upstream and downstream transactions (EP bit)
- Internal SRAM protections (ECC or parity)
- Real and fake error injection at physical layer
- DPC capability at system level (supported at root port)
- Proprietary and spec-defined correctable errors (CE) handling

PCIE POISON SUPPORT

- PCIe forwards ingress/egress poison as defined by the Spec.
- Poison egress from GPU
  - Forwarded from internal traffic sources (e.g. L2 cache and DRAM)
  - Generated on uncorrectable SRAM ECC errors for portion of the PCIe controller
- Poison ingress to GPU
  - Forwarded to internal requesters from external sources
  - AER reporting performed per PCIe spec
  - Controller optionally converts Completions with Status=UR/CA to poison on ingress

PCIE INTERNAL SRAM PROTECTION

- Like SRAMs in other IPs in GPU, internal SRAM parity/ECC errors in PCIe controller are considered transient until the threshold has been reached.
  - (please see SRAM correctable and uncorrectable error handling section)
- Parity and uncorrected ECC errors are reported as interrupt.
  - They are considered fatal leading to GPU hot-reset.
- Errors are logged in InfoROM via other GPU interfaces and they do not go through PCIe.

PCIE FAILURES AND RECOVERY

- Poison and ECC/parity uncorrectable error (UCE) recovery procedure:
  - Uncontained poison consumption -> kill all app contexts.
  - Contained poison consumption -> kill affected app context.
  - Uncontained UCE error in PCIE controller internal buffers -> kill all app contexts and reset GPU.
  - Errors affecting UVM -> Reboot OS.
- With SRIOV, errors can be contained to VFs, use VF AER, etc.
- NVIDIA GPU driver handles all GPU interrupts but does not handle AER (left to kernel/aerdrv)
- Posted deadlock timeout interrupt is fatal leading to GPU reset.

PCIE LINE ERROR MECHANISMS

- FEC in Gen6 (Blackwell+) and CRC along with replay will act as correction method below a certain threshold.
  - All mandatory counters, interrupts, rate measurements are implemented to PCIe specification
  - Proprietary per-lane FBER counters with extended widths are preferable to small spec-defined counters
  - Proprietary, flexible FBER rate threshold interrupt is preferable to fixed-rate spec-defined mechanism
- Error rates above a certain rate can lead to PCIE Recovery State which causes performance degradation.
- Error rates higher than the recovery threshold may lead to link-down, requiring GPU reset.
- NVidia-SMI (in-band command line) and SMBPBI (OOB) can be used to read the error correction and recovery counts.
- Field Diag and NVQual also measure correction and recovery counters and check against NVIDIA-defined thresholds, no less stringent than PCIe spec.

PCIE LINE ERROR POLICIES

- Transient line errors result in a single recovery event. Persistent recovery events are sign of required service.
- Line errors can’t be isolated to an application.
- Service and recovery actions to be taken:
  - In case of fatal errors GPU will be unavailable/reset and OOB is necessary to retrieve telemetry.
  - In case of non-fatal but above single recovery threshold, service routine is recommended.
  - In case of non-fatal but above spec BER threshold, service routine is recommended.
  - If not fatal but below threshold it can be safely ignored.
    - We expect a low but non-zero quantity of FBER and CRC errors on all PCIe links.
- Errors above spec defined BER could be associated with higher risk of SDC.
- Example service/failure-analysis routine*:
  - Take the GPU offline, run field diag, narrow down to failing physical link.
  - For failing link, service the hardware as required (e.g. cleaning connectors, swap SXM module)

***** This is a procedural suggestion; official customer RMA instructions still need to be followed*

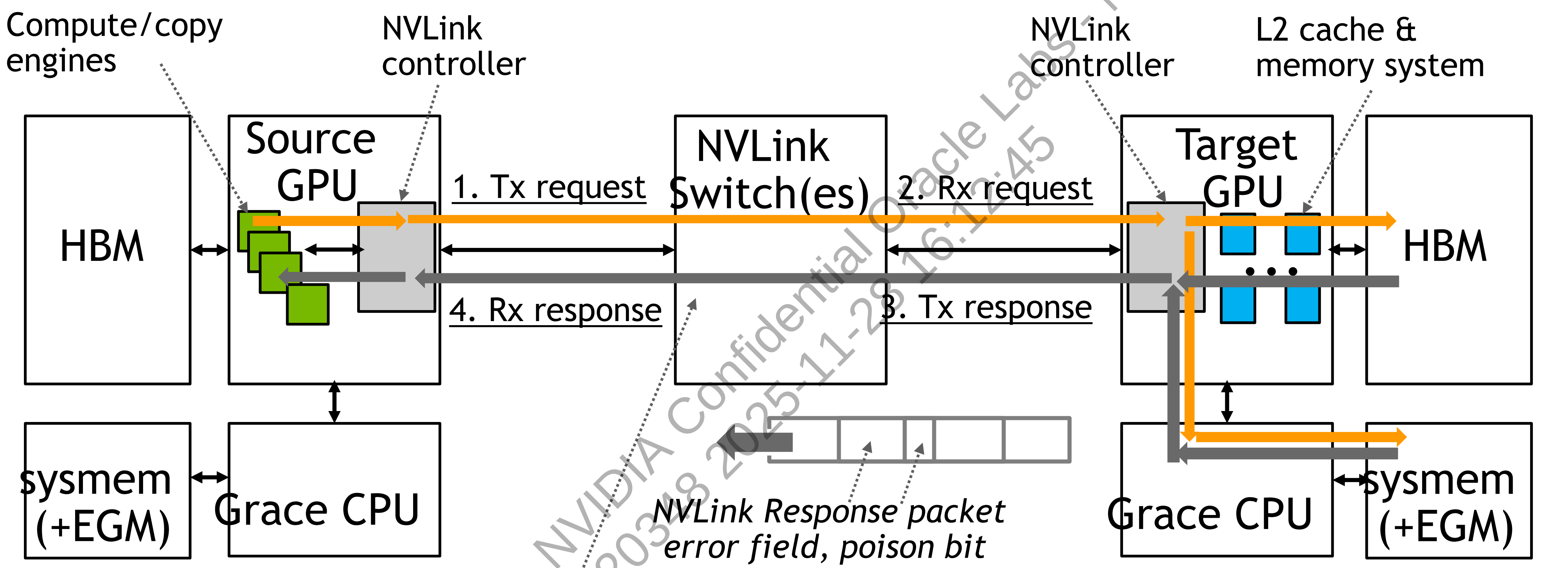
PCIE-RELATED ERROR LOGGING

- 3 ways: XIDs in sys logs on the node, OOB messages through SMBPBI, Running “nvidia-smi” on the node

Error ID	Related PCIe mechanism	
Xid 172 (w/ 48)	Internal hardware ECC/Parity	"PCIe" in Xid Text
Xid 79	PCIe fatal link errors	Conflated with ALL GPU fatal issues and link partner PCIe errors
Xid 167	PCIe fatal timeout error	
Xid TBD	Corrected rate > threshold	
SMBPBI opcode 0x21	Corrected and Fatal events	OOB, retrievable after link-down
SMBPBI opcode TBD	"Sticky" corrected rate>threshold	OOB, retrievable after link-down
nvidia-smi pci --getErrorCounters	Corrected errors – raw counts	Interpretation needed for rate

NVLINK

BASIC NVLINK TOPOLOGY AND TRANSACTION FLOW



Target memory can be HBM or EGM

- *Larger systems have multiple switch hops*
- *Direct-connect (GPU-to-GPU) systems have no NVLink switches*

- *NVLink response errors*
  - *STO_ERR (detected at source GPU, triggers cross-contain, brings down all active peer-enabled contexts)*
  - *PRIV_ERR (detected at source or target GPU and returned to source GPU; some cases trigger cross-contain)*

SUMMARY OF NVLINK RAS

NVLink

- Generations of NVLink
  - Ampere: NVL3 (50 Gb/s NRZ copper / 53.125 Gb/s PAM4 optical), Hopper: NVL4 (106.25 Gb/s)
  - Blackwell: NVL5 (212.5 Gb/s), Rubin: NVL6, R150: NVL6.2 (introduces x1 links)
- STO_ERR (Source Timeout error).
  - Transaction Layer timeout counter detects packet loss. Non-fatal error; local port enters contain mode and triggers "cross-contain" mode:
    - All NVLinks immediately enter contain mode.
    - Drop new Tx requests and new Rx responses to prevent data corruption.
  - Non-fatal from HW standpoint. All active peer-enabled contexts on the GPU - both usermode and kernel - are impacted. Context owners must be notified in order to ensure program correctness and complete recovery.
- Datapath RAMs protected with ECC.
  - SBE correction. DBE is contained and is fatal to GPU.
    - DBE forces link down.
    - DBE has extremely low (practically zero) FIT. Requires PF FLR on local GPU to recover. Other GPUs not impacted directly.
  - HW engine for periodic scrub of Routing RAMs storing static data.

SUMMARY OF NVLINK RAS (CONT)

NVLink

- Microcontroller RAMs protected with parity (NVL3/NVL4) and ECC (NVL5/NVL6).
- Link forced down due to DBE in pipeline or fatal error in Link/Physical layers. Requires PF FLR to recover.
- PRIV_ERR (Privilege error)
  - Non-fatal. Signaled for source GPU Remap table programming error and for target GPU MMU (FLA TLB) programming error.
  - For source PRIV_ERR: In NVL5, port enters contain mode, triggers "cross-contain" (see previous slide); in NVL6, contain/cross-contain is programmable: contain/cross-contain enabled only for PRIV_ERR due to invalid Remap Table entries.
- Poison support
  - Poison bit in NVLink packet set if there is DBE in payload data.
  - Propagates through NVLink fabric and GPU datapath to GPU memory destination where data is poisoned.
- CRC protection on link (for NVL3 and NVL4; dropped in NVL5 in favor of stronger FEC algorithm)
  - Physical layer retry on CRC error.
  - No ECRC.

SUMMARY OF NVLINK RAS (CONT)

NVLink

- All NVLink transactions are non-posted, so can attribute network errors to the source GPU
  - Reads, writes, Atomics all have response packet sent from target GPU to source GPU.
  - Response packet contains error status field.
  - Transaction timeout error (STO_ERR) unique - signaled locally at **source** GPU; response does not flow on NVLink fabric
    - In contrast, PRIV_ERR can be signaled at **source** GPU **or** **target** GPU.
  - Applies to all single and multi-node GPU configurations.
- More details on NVL5 customer-visible handling and recovering flows for each NVLink error available in NVIDIA Service RAS Catalog: 1116117 NVIDIA Server RAS Catalog.
  - Recovery actions are evolving with each product to minimize GPU fatal errors and minimize blast radius.
  - Link failure diagnosability features are evolving with each product, to help pinpoint location of fabric failures causing packet drops and STO_ERR.

NVLINK XID INDICATIONS

Only selected XIDs are shown. Refer to Server RAS catalog for full list: 1116117 NVIDIA Server RAS Catalog.

- Note: **XIDs associated with RLW unit** in NVL5 (Blackwell) and NVL6 (Rubin) replace equivalent **SXIDs** for NVSwitch NPORT in NVL3 (A100) and NVL4 (H100) because Remap/routing layer moved from NVSwitch to GPU starting with NVL5 (Blackwell).

XID	Subcode	Likely cause
74 (NVLINK_ERROR)	n/a	Marginal link integrity or mechanical connection. Fatal, requires GPU reset. [1]
92 (EXCESSIVE_SBE_INTERRUPTS)	n/a	Internal RAM marginal or faulty. Non-fatal but may require replacing GPU.
137 (NVLINK_FLA_PRIV_ERR)	n/a	Signaled at source Node, when fault reported at target GPU MMU PRIV_ERR, likely due to illegal NVLink peer-to-peer access made by application. Non-fatal.
145 (NVLINK_RLW_ERROR)	00100 (RLW_REMAP)	Source GPU detects PRIV_ERR due to system or application software error (invalid addr or request type, or inconsistent MMU vs. Remap Table entry). Non-fatal, triggers contain for NVL5 (contain can be disabled for NVL6).
	00110 (RLW_RXPIPE)	Source or target GPU detects DLID mismatch, likely due to system software programming error (source GPU Remap table or NVSwitch routing table). Non-fatal; packet is dropped; leads to STO_ERR at source GPU. Or Signaled at source Node, when fault reported at target GPU MMU PRIV_ERR, likely due to illegal NVLink peer-to-peer access made by application. (This case is reported together with XID-137) Non-fatal.
	00111 (RLW_SRC_TRACK)	Source GPU detects response timeout (STO_ERR) due to packet drop in fabric or at target GPU. Non-fatal, triggers contain.
149 (NVLINK_NETIR_ERROR)	010001 (NETIR_LNK_EVT/NETIR_LINK_DOWN)	Fault in link controller, connector, PCB trace, or cable. Fatal, requires GPU reset. [1]
	00000 (NETIR_BER_EVENT)	Potential signal integrity issues detected. Informational only but may lead to STO_ERR at source if errors rates persist at a high rate. Possible leading indicator of faulty link.

1. Per-link reset not available in Blackwell, Rubin; recovery requires GPU PF FLR.

NVLINK XID INDICATIONS (CONT)

Additional, less common XIDs shown here Only selected XIDs are shown. Refer to Server RAS catalog for full list: 1116117 NVIDIA Server RAS Catalog.

XID	Subcode	Likely cause
144 (SAW_MVB)	(Multiple Subcodes)	Packet buffer RAM DBE due to HW fault. Fatal, requires GPU reset.[1]
146 (TLW_TX or TLW_RX)	(Multiple Subcodes)	Packet buffer RAM DBE due to HW fault. Fatal, requires GPU reset.[1]
(n/a) 147 (TREX)		TREX not enabled in production (RTT telemetry function provided by TREX is not in NVL5 POR).
148 (NVLPW_CTRL)		PRI error caused by bug in system software.
150 (MSE Watchdog)		Likely due to HW fault in RISC-V CPU or memory. Fatal, requires GPU reset[1]

[1] Per-link reset not available in Blackwell, Rubin; recovery requires GPU PF FLR.

C2C

SUMMARY OF CHIP-2-CHIP RAS

NVLINK-C2C

- Bi-directional poison propagation support.
- Physical layer error -
  - Link CRC with replay for link error detection and recovery.
  - Link errors are correctable and not fatal.
- Excessive errors may lead to GPU hang or timeout
- XID-121 -
  - GPU driver raises XID-121 when excessive errors are detected by the link.
  - XID-121 is informational with no corrective action.
- C2C errors recovery action is FLR (on Blackwell) to GPU (Hopper needs ACPI_RST).
  - It will reset the GPU, which will reset and retrain C2C.
  - It does not crash CPU
- Error injection capability for HW validation and SW verification (it will trigger CRC retry).

SILENT DATA ERROR MITIGATION

SILENT DATA ERRORS MITIGATION

Silent Data Error is due to Undetected Errors by Built-In Detection Mechanisms

These Undetected Errors, caused by Permanent or Transient Faults, Corrupt Application Output.

Permanent Faults

- **Enhancing Built-In Detection Mechanisms to Augment Runtime Detection Coverage**
- **Detection:**
  - Structural (IST) or Functional Diagnostics to Detect Faults after or during Mission Execution Time
  - Algorithmic Based Error Detection (ABED) to Detect Anomalies During Mission Runtime
- **In Field Repair (where possible)**

Transient Faults

- **All Critical SRAM Structures ECC or Parity Protected**
- **Column Interleaving to Eliminate Multi-Bit Errors in SRAM Logical Words**
- **ABED if needed**

IN FIELD REPAIR

IN FIELD REPAIR

FieldDiag repairs

Environment	Error Log / Sequence	Root Cause	Repair	Applicability
FieldDiag: - IST Errors (MBIST and LBIST)	Bad Nvidia chip OK - GPU tuning candidate OK - GPU is tuned successfully / GPU tuning failed	TPC Errors	TPC repair (fielddiag)	Ready: Blackwell (and later) WIP: Hopper (LBIST-only)
FieldDiag: - IST Errors (MBIST and LBIST)	Bad Nvidia chip OK - GPU tuning candidate OK - GPU is tuned successfully / GPU tuning failed	LTS Errors	LTS repair (fielddiag)	Ready: Blackwell (and later)
FieldDiag: - Functional Tests Errors (ECC UCEs or result mis-match)	CRC checksum mis-compare / ECC errors, etc. OK - GPU tuning candidate OK - GPU is tuned successfully / GPU tuning failed	TPC Errors	TPC repair (fielddiag)	WIP: Hopper, Blackwell (and later)
FieldDiag: - Functional Tests Errors (ECC UCEs or result mis-match)	Bad memory / ECC errors, etc. OK - GPU tuning candidate OK - GPU is tuned successfully / GPU tuning failed	LTS Errors	LTS repair (fielddiag)	Ready: Blackwell (and later) WIP: Hopper
FieldDiag: - Functional Tests Errors (HBM ECC UCEs)	MODS runs functional tests and finds faulty rows and perform row remapping based on policy enforced by MODS cmd line.	HBM Errors	Row remapping (fielddiag)	Ready: Hopper WIP: Blackwell (and later)

BACKUP

GLOSSARY

ABBREVIATION	DESCRIPTION
GCC	Global Constant Cache – used to store instructions and constant data
MMU	Memory Management Unit
TLB	Translation Lookahead Buffer
OFA	Optical Flow Accelerator. Dedicated hardware for Stereo / Optical Flow (i.e., estimate motion of object from one frame to the next). DL Application: Video Classification
GSP	GPU System Processor. Microcontroller that runs subset of GPU driver
FSP	Foundation Security Processor. Root of Trust for boot, firmware security, and attestation.
SEC	Security Engine. Used for Confidential Compute and Digital Rights Management (DRM)
PMU	Power Management Unit