

Re-enabling Fused-Off Compute Units on the AMD BC-250 APU via Register-Level Modification

Scott Duggan
Independent Research
duggasco@gmail.com

May 2026

Abstract

The AMD BC-250 is a standalone single-board computer built around a salvaged PlayStation 5 APU (“Ariel”), shipping with 24 of 40 physical RDNA 2 GPU compute units (CUs) active. Industry consensus holds that CU harvesting on modern AMD silicon is enforced by irreversible hardware fuses, rendering disabled CUs permanently inaccessible. We demonstrate that this assumption is incorrect for the BC-250: the CU disable mask resides in a software-writable configuration register (`CC_GC_SHADER_ARRAY_CONFIG`), and a second power-gating register (`SPI_PG_ENABLE_STATIC_WGP_MASK`) gates hardware dispatch. By writing both registers from within the kernel GPU driver’s initialization path, we re-enabled all 40 CUs, achieving a $1.54\times$ improvement in LLM inference prefill throughput (302 to 466 tok/s) with zero compute errors across extensive stress testing. The technique has been verified on three separate boards, and the patch is publicly available [1]. To our knowledge, this is the first documented instance of CU re-enablement on a production AMD APU through software-only means.

1. Introduction

1.1 Background

Semiconductor yield management routinely produces dies where a subset of functional units fail quality or power targets. Rather than discard these dies, manufacturers *harvest* (disable) the failing units and sell the part as a lower-tier SKU. This practice is well-documented across AMD, NVIDIA, and Intel product lines and is a key economic lever in modern chip manufacturing.

The AMD BC-250 represents an unusual case. Originally fabricated as a full Navi 10 die—the same 7nm TSMC silicon used in the Radeon RX 5700 XT and PlayStation 5—these APUs were binned out of Sony’s PS5 qualification process. AMD repurposed them for cryptocurrency mining by pairing the die with 6 active Zen 2 CPU cores, 16 GB of unified GDDR6 memory, and a minimal standalone PCB manufactured by ASRock Rack. The result is a complete single-board computer requiring no host system, available on the secondary market for \$50–150.

Each BC-250 ships with 24 of 40 GPU CUs active (40% harvested), organized as 6 active CUs per shader

array across 4 shader arrays. With the mining market’s collapse, these boards have found a second life as ultra-low-cost AI inference nodes, where GPU CU count directly constrains throughput. An active community of over 3,500 members has developed custom BIOS firmware, GPU power governors, cooling solutions, and software tooling to support this use case.

1.2 Prior Assumptions

The prevailing model for AMD CU harvesting holds that:

1. On-die fuses are blown during manufacturing to permanently disable defective or surplus CUs.
2. The Platform Security Processor (PSP) reads these fuses during early boot and programs the GPU’s configuration registers accordingly.
3. The resulting CU mask is a read-only hardware constant that the OS driver can only observe, never modify.
4. `GC_USER_SHADER_ARRAY_CONFIG` allows software to disable *additional* CUs but cannot re-enable fuse-disabled ones, since the effective mask is the logical OR of the hardware and user masks.

Our work challenges assumptions 2, 3, and 4.

1.3 Contribution

We present:

- A register-level characterization of the CU harvesting mechanism on the BC-250 APU.
- Proof that `CC_GC_SHADER_ARRAY_CONFIG` is software-writable from inside the kernel driver.
- Discovery of a second required register (`SPI_PG_ENABLE_STATIC_WGP_MASK`) that gates hardware dispatch to re-enabled CUs.
- A controlled 4-state experiment isolating each register’s contribution.
- Performance and stability characterization under real LLM inference workloads.
- Verification across three separate boards, all showing identical results.

2. Hardware Platform

The BC-250 is a unified APU—CPU, GPU, and memory share a single die and a single 16 GB GDDR6 memory pool. Table 1 summarizes the key specifications.

Table 1: BC-250 APU specifications.

Parameter	Value
Die	Navi 10 (“Ariel”), 7nm TSMC
GPU architecture	RDNA 2 (gfx1013)
Shader engines	2
Shader arrays / SE	2 (4 total)
Physical CUs / SA	10 (40 total)
Active CUs (stock)	6 / SA (24 total)
WGPs / SA	5 (1 WGP = 2 CUs)
Memory	16 GB GDDR6, 256-bit
Bandwidth	~448 GB/s
CPU	6× Zen 2 cores, 12 threads
TDP	220W rated
Market price	\$50–150

The GFX10 architecture organizes CUs into Work-Group Processors (WGPs), each containing 2 CUs. The BC-250 has 5 WGPs per shader array, of which 3 are active at stock. The IP discovery table reports the full 40-CU physical die (`gc_num.tcps = 40`) with no harvest entries—the harvesting is applied at a layer above hardware discovery.

Three physically separate boards (bc250-1, bc250-2, bc250-3) were tested. All three exhibited identical harvest patterns. Figure 1 illustrates the die topology and harvest pattern.

3. Methodology

3.1 Approach Evolution

Direct register access from an external kernel module was attempted first and proved unworkable. The GPU’s power management (GFXOFF) and the `amdgpu` driver’s exclusive ownership of indirect register access paths created an inherent race condition. Three separate attempts crashed the board via deadlock with the driver’s `pcie_idx_lock`.

We pivoted to patching the `amdgpu` kernel driver source directly (Linux 6.19.14, Debian), placing all register reads and writes inside the driver’s own GFX initialization path where the GRBM bank-selection lock is already held.

3.2 Baseline Characterization

We instrumented the driver’s per-shader-array WGP bitmap function (`gfx_v10_0_get_wgp_active_bitmap_per_sh`) to log raw register state during CU enumeration:

```
SE0 SH0: cc_raw=0xfff80000
          cc_inactive=0x0000fff8
          active_wgp=0x00000007
SE0 SH1: [identical]
SE1 SH0: [identical]
SE1 SH1: [identical]
```

The out-of-range mask (`0xffffffe0`) marks WGP bits 5+ as physically nonexistent. Within the valid 5-WGP range, `cc_inactive=0xfff8` disables WGPs 3 and 4 in every shader array on our three boards.

3.3 Community Harvest Map Survey

Following publication of the unlock patch, the BC-250 Discord community collected CU harvest maps from 55+ boards using a standardized script authored by Studebaker (Figure 2). Key findings:

- **100% of boards** have exactly 24/40 CUs active (6 per SA).
- **~74%** show the identical contiguous pattern across all 4 SAs: CUs 0–5 active, CUs 6–9 fused.
- **~26%** have at least one SA with a non-contiguous harvest pattern.
- At least one board with a “clean” contiguous map exhibited 3 faulty CU pairs after unlock.

The uniform 24-CU count across all boards confirms deliberate SKU binning. The mix of contiguous and non-contiguous patterns suggests AMD applied a fixed 6-CU-per-SA quota but allowed the specific CUs disabled to

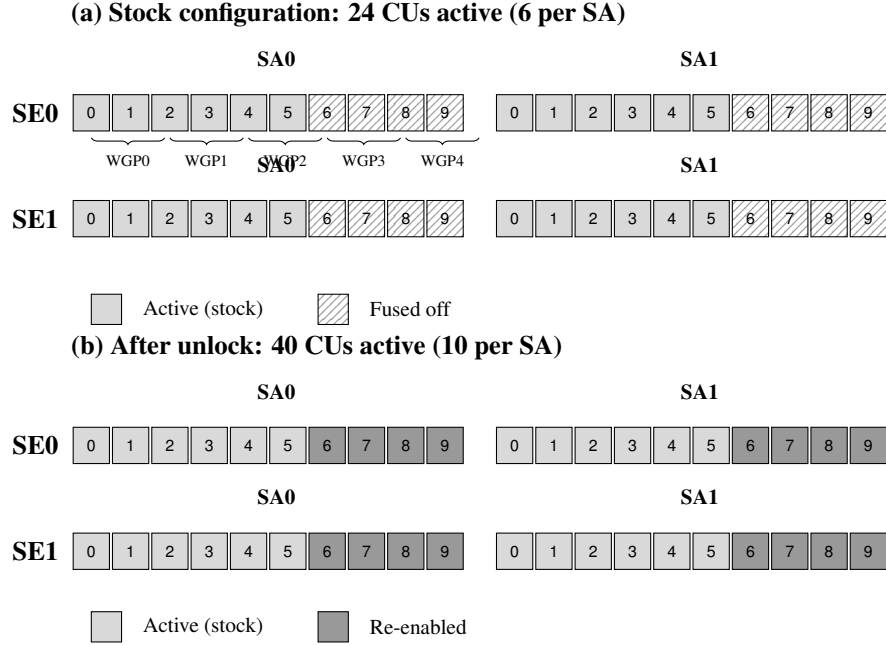


Figure 1: BC-250 die topology showing CU harvest pattern across 2 shader engines $\times$ 2 shader arrays. (a) Stock: CUs 0–5 active, CUs 6–9 fused off per SA (24 total). (b) After register modification: all 40 CUs active. Pattern is identical across all three boards tested.

vary—likely reflecting actual defect locations on some dies while using a default contiguous mask on fully functional dies. The existence of boards with faulty CUs after unlock underscores the importance of compute verification (Section 4.3).

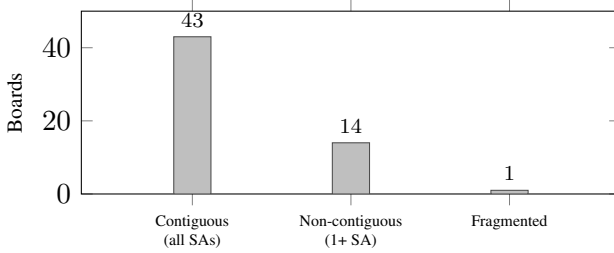


Figure 2: Harvest pattern distribution across 58 community-reported boards. Most show the standard contiguous pattern; a minority have non-contiguous harvesting in one or more shader arrays, likely reflecting genuine defect locations.

3.4 Write-Probe Experiment

We added a guarded boot parameter (`bc250_cc_write_mode`) to the patched driver with multiple modes:

- **Mode 0:** No modification (baseline).
- **Mode 1:** Write zero, read back, restore.
- **Mode 2:** Clear SE0/SH0, no restore.
- **Mode 3:** Clear all 4 SAs, no restore.

- **Mode 4:** Write/read/restore all 4 SAs.

Mode 1 (write-then-restore on SE0/SH0) yielded:

```
original    = 0xffff80000
after_zero  = 0xffe00000
after_restore = 0xffff80000
```

The upper out-of-range bits remained set, but the valid inactive-WGP bits cleared. This proved that `CC_GC_SHADER_ARRAY_CONFIG` is **not** a read-only hardware fuse latch on this silicon.

3.5 Boot Safety Mechanism

To prevent a failed CU-enable from persisting across re-boots, we used a self-deleting one-shot modprobe rule:

```
install amdgpu /bin/sh -c \
'rm -f /etc/modprobe.d/bc250-cc-\
oneshot.conf; \
exec /sbin/modprobe --ignore-install \
amdgpu bc250_cc_write_mode=3'
```

The rule removes itself before invoking the driver, ensuring that a crash during the enable attempt does not loop on subsequent power cycles.

3.6 Discovery of the Second Register

Initial mode-3 boots produced 40-CU enumeration and correct Vulkan compute results, but LLM inference throughput showed no improvement. Investigation using

AMD’s UMR (User-Mode Register debugger) revealed a second gate: `SPI_PG_ENABLE_STATIC_WGP_MASK` (offset `0x1277`).

At stock, this register is `0x7`—enabling hardware dispatch only to WGP 0–2. Even with the CC mask cleared, the shader processor continued routing work to only the original 3 WGP. Setting this register to `0x1F` via UMR at runtime, combined with the CC clear at boot, unlocked actual compute scaling.

A companion register, `RLC_PG_ALWAYS_ON_WGP_MASK`, must also be set to `0x1F` to prevent the Run List Controller from power-gating re-enabled WGP during idle periods. Figure 3 illustrates the dual-register gating architecture.

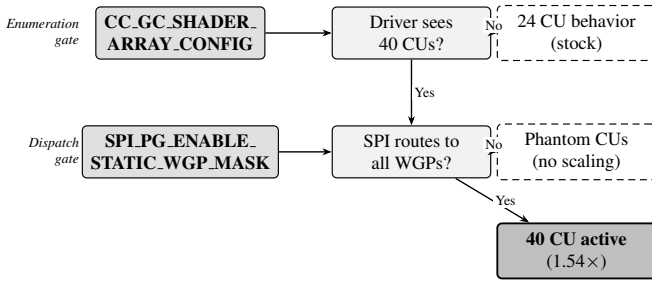


Figure 3: Dual-register gating architecture. Both the CC enumeration mask and SPI dispatch mask must be set; either alone produces no performance change.

4. Results

4.1 4-State Controlled Experiment

Table 2 presents the controlled experiment isolating each register’s contribution, measured via 512-token LLM prefill at 2 GHz on bc250-2.

Table 2: 4-state A/B test: register contributions.

CC	SPI	tok/s	W	Scale
Stock	0x7	302	56	1.00×
Stock	0x1F	302	56	1.00×
Clear	0x7	302	56	1.00×
Clear	0x1F	466	181	1.54×

Neither register alone produces any gain. Both must be set: `CC_GC_SHADER_ARRAY_CONFIG` controls CU *enumeration* (driver visibility), while `SPI_PG_ENABLE_STATIC_WGP_MASK` controls hardware *dispatch* (physical wavefront routing).

4.2 Operating Point Optimization

At 2 GHz, the 40-CU configuration draws 181W and approaches thermal limits. Table 3 shows the sustainable 1500 MHz / 900 mV point.

Table 3: 24-CU vs 40-CU at 1500 MHz.

Config	tok/s	W	°C	Scale
24 CU @ 1500	230	55	71	1.00×
40 CU @ 1500	372	125	83	1.61×

The $1.61\times$ scaling at 1500 MHz exceeds the $1.54\times$ at 2 GHz because the lower clock avoids power limiters, allowing better utilization of the wider shader array. GPU frequency management is handled by the community-developed `cyan-skillfish-governor-smu` (Section 6). Figure 4 summarizes the performance results.

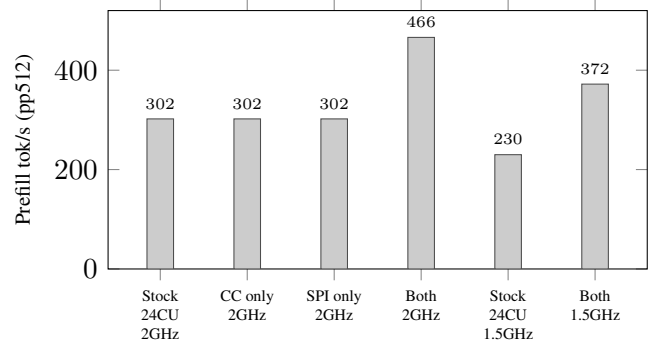


Figure 4: LLM prefill throughput across all tested configurations. Only the combined CC+SPI modification produces a gain. The 1.5 GHz point achieves better scaling ($1.61\times$) than 2 GHz ($1.54\times$) due to reduced power limiting.

4.3 Compute Correctness

Vulkan compute verification was performed at each state:

- **28-CU** (1 SA cleared): 1M and 4M element Vulkan shader dispatch, zero errors.
- **40-CU** (all SAs cleared): 1M and 4M element dispatch, zero errors.
- **Stress:** 10 consecutive 4M-element runs at 40 CUs—zero errors, zero GPU recovery events, zero ring timeouts, zero VM faults.

No `amdgpu` reset or fault messages appeared in kernel logs during any test. GPU temperature during sustained 40-CU stress: 70–71°C at 1500 MHz.

4.4 Reproducibility Across Boards

The enable procedure was validated across multiple boot cycles on three physically separate boards:

1. Boot to baseline 24 CUs (verified via RADV).
2. Install one-shot modprobe config, reboot.
3. Verify 40 CUs: all SAs report `cu_mask = 0x3ff`.
4. Set SPI dispatch mask via UMR.
5. Run workload, verify scaling.
6. Reboot—returns to baseline 24 CUs.

All three boards (bc250-1, bc250-2, bc250-3) showed identical behavior and identical harvest patterns. The boards were independently sourced at different times and showed no variation in the CC mask, SPI mask, or CU layout.

4.5 Cluster Deployment

Following single-board validation, all three boards were deployed as a Vulkan RPC inference cluster, each running at 40 CUs. The cluster runs a 22 GB Qwen3.6 35B-A3B MoE model split across boards, achieving 187 tok/s prefill and 28 tok/s generation at 128-token context, with 32K context achievable using Q4.0 KV cache quantization. The boards boot disklessly via PXE/iSCSI from a NAS appliance.

5. Discussion

5.1 Why Are These CUs Functional?

The community harvest map survey (Section 3.3, $n = 58$) reveals a nuanced picture. Approximately 74% of boards show a perfectly contiguous pattern across all 4 SAs, consistent with a deliberate SKU mask applied to fully functional silicon. The remaining 26% show at least one SA with a non-contiguous pattern, suggesting genuine defect-driven harvesting on those dies.

All boards share the same 24/40 CU quota, confirming a fixed SKU definition. The BC-250 uses Navi 10 dies that failed PS5 qualification (which requires 36/40 CUs). AMD appears to have applied a uniform 6-CU-per-SA quota, using a default contiguous mask on defect-free dies and mapping actual defect locations on others.

On our three boards (all contiguous-pattern), re-enabled CUs compute correctly and survive stress testing. However, at least one community member with a contiguous map reported 3 faulty CU pairs after unlock—demonstrating that the harvest map alone is not a reliable predictor of CU health, and compute verification is essential.

5.2 Dual-Register Gating Architecture

The two-register mechanism provides defense in depth:

1. **CC_GC_SHADER_ARRAY_CONFIG** controls CU *enumeration*. When WGP are marked inactive, the driver reports fewer CUs and the runtime allocates work accordingly.
2. **SPI_PG_ENABLE_STATIC_WGP_MASK** controls hardware *dispatch*. Even if the driver sees 40 CUs, the shader processor only routes wavefronts to WGP enabled here.

Clearing only the enumeration mask creates a “phantom CU” state—the driver believes additional CUs exist, but work is silently routed to the enabled subset. This produces no errors and no performance change.

5.3 Economics of Salvage Compute

Table 4 compares the per-CU economics at current secondary market prices (\$50–150).

Table 4: Economics: stock vs. unlocked BC-250 (at \$150).

Metric	24 CU	40 CU
pp512 @ 1500 MHz	230 tok/s	372 tok/s
\$/CU	\$6.25	\$3.75
Perf/\$ (tok/s/\$)	1.53	2.48
tok/s/W	4.18	2.98

At the low end of the price range (\$50), the unlocked 40-CU configuration delivers 7.44 tok/s/\$—competitive with consumer GPUs costing an order of magnitude more, provided the workload fits in 16 GB of unified memory.

5.4 Broader Applicability

The methodology—instrumenting the driver’s own init path to test register writability under proper locking—is applicable to other AMD products. Whether other harvested parts (e.g., RX 7800 XT vs RX 7900 XTX) use similarly writable CC masks is an open question. The gfx1013 variant’s origin as a repurposed console APU may make it uniquely permissive.

5.5 Limitations

- **Persistence:** The CC mask reverts on reboot; the SPI mask must be set post-boot. A production deployment should integrate both writes into the driver init path.

- **CPU cores:** The APU’s CPU harvesting (6/8 Zen 2 cores) is PSP-enforced and not clearable via this technique. INIT/SIPI signals to disabled cores return error status 0x4.
- **Thermal ceiling:** 40 CUs at 2 GHz draw 181W. The 1500 MHz / 900 mV point (125W) is recommended for sustained workloads.
- **Sample size:** Three boards tested with identical results. Population-level confidence requires broader community replication.
- **Defect risk:** Some boards may carry genuinely defective CUs. Extended compute verification is essential before relying on re-enabled CUs.

6. Community Ecosystem

This work builds on a substantial open-source ecosystem developed by the BC-250 community (3,500+ members on Discord). While the CU re-enablement technique is novel, the platform’s viability as a compute node depends on prior community contributions:

Custom BIOS firmware. TuxThePenguin0 (“Seg-fault”) developed modified BIOS images that unlock dynamic VRAM allocation and chipset configuration menus, enabling the flexible memory split between CPU and GPU that makes inference workloads practical on a 16 GB unified memory system [2].

GPU power management. FilippoR’s `cyan-skillfish-governor-smu` [3] provides SMU-based GPU clock/voltage scaling. Without a governor, the BC-250 GPU locks at 1500 MHz regardless of load. The governor enables the variable clock states used in our operating-point analysis. `mothenjoyer69`’s earlier `oberon-governor` [4] established the initial approach.

CPU overclocking and ACPI. The `bc250-collective`’s `bc250-smu-oc` tool enables Zen 2 CPU frequency adjustment via SMU mailbox. Their `bc250-acpi-fix` provides SSDT tables for proper C-State and P-State enumeration [5, 6].

Kernel contributions. Magnap identified the critical TTM page limit fix (`ttm.pages_limit=4194304`) that unlocks full 16 GB memory allocation; without this parameter, the GPU is capped at ~7.4 GiB.

Documentation. `elektricM` compiled comprehensive documentation from over 9,700 Discord messages [8]. `mothenjoyer69` authored the original hardware documentation [9]. `akandr` developed and published inference benchmarks across 31 models, establishing the BC-250’s performance envelope [10].

Hardware ecosystem. `provd` developed a ray tracing compatibility patch. `Fred78290` authored the `nct6687d` kernel module [7] for fan/thermal control via the Nuvoton NCT6686D SuperIO. Over 145 3D-printable case designs exist on community repositories.

Community replication. Following publication of the unlock patch, 58 community members submitted CU harvest maps (Section 3.3), and multiple users have successfully applied the unlock procedure to their boards. At least one user reported faulty CU pairs after unlock, confirming that compute verification is essential.

7. Conclusion

We demonstrated that the AMD BC-250 APU’s GPU CU harvesting is enforced by two software-writable configuration registers rather than irreversible hardware fuses. By clearing `CC_GC_SHADER_ARRAY_CONFIG` during kernel driver initialization and setting `SPI_PG_ENABLE_STATIC_WGP_MASK` to enable dispatch to all work-group processors, we re-enabled all 40 physical CUs on a part shipped with 24—a 67% increase in GPU compute resources from a purely software modification.

The re-enabled CUs are fully functional: they produce correct results, survive sustained stress testing, and deliver $1.54\text{--}1.61\times$ throughput improvement on LLM inference workloads. The technique is reproducible across boot cycles and across three separate boards now operating as a production inference cluster.

This finding challenges the assumption that semiconductor harvesting always represents a permanent physical constraint. For the growing community repurposing salvaged silicon for AI inference, it demonstrates that register-level analysis of hardware gating mechanisms can recover significant compute left on the table by the original manufacturer.

Acknowledgments

This work was conducted as independent research with hardware purchased at secondary market prices. The kernel driver patch and enable tooling are published at <https://github.com/duggasco/bc250-40cu-unlock>. AMD’s open-source amdgpu kernel driver and Mesa/RADV Vulkan driver made this work possible—the ability to read, instrument, and modify the driver source was the key enabler.

Special thanks to the community contributors whose work made the BC-250 a viable compute platform: TuxThePenguin0 for custom BIOS development [2], FilippoR for the cyan-skillfish-governor-smu [3], mothenjoyer69 for hardware documentation and the original GPU governor [9, 4], elektrikM for compiling community knowledge from over 9,700 Discord messages [8], akandr for publishing inference benchmarks across 31 models [10], Magnap for the critical TTM page limit kernel fix, the bc250-collective for SMU overclocking and ACPI tooling [5, 6], Fred78290 for the fan control kernel module [7], and the 58 Discord community members (discord.gg/8eZfFWhczz) who submitted CU harvest maps for the population survey.

References

- [1] S. Duggan, “BC-250 40-CU Unlock,” <https://github.com/duggasco/bc250-40cu-unlock>, 2026.
- [2] TuxThePenguin0, “BC-250 Modified BIOS,” <https://gitlab.com/TuxThePenguin0/bc250-bios/>, 2025.
- [3] FilippoR, “Cyan Skillfish Governor (SMU),” <https://github.com/filippor/cyan-skillfish-governor>, 2025.
- [4] mothenjoyer69, “Oberon Governor,” <https://gitlab.com/mothenjoyer69/oberon-governor>, 2025.
- [5] bc250-collective, “BC-250 SMU Overclock,” https://github.com/bc250-collective/bc250_smu_oc, 2025.
- [6] bc250-collective, “BC-250 ACPI Fix,” <https://github.com/bc250-collective/bc250-acpi-fix>, 2025.
- [7] Fred78290, “NCT6687D Kernel Module,” <https://github.com/Fred78290/nct6687d>, 2025.
- [8] elektrikM, “AMD BC-250 Documentation,” <https://github.com/elektricM/amd-bc250-docs>, 2025.
- [9] mothenjoyer69, “BC-250 Documentation,” <https://github.com/mothenjoyer69/bc250-documentation>, 2024.
- [10] akandr, “BC-250 Inference Benchmarks,” <https://github.com/akandr/bc250>, 2026.

A. Register Details

CC_GC_SHADER_ARRAY_CONFIG

DWORD offset 0x2042C, per-instance 0x0226f.

Bits	Field	Stock	Cleared
31:20	Out-of-range WGP mask	0xFFE	0xFFE
19:16	Inactive WGPs (valid)	0xF (WGP 3–4 off)	0x0 (all active)
15:0	Reserved	0x0000	0x0000

Full raw value: 0xFFF80000 (stock) → 0xFFE00000 (cleared).

SPI_PG_ENABLE_STATIC_WGP_MASK

Offset 0x1277. Each bit enables hardware shader dispatch to one WGP within each shader array.

Bit	WGP	CUs	Stock	Enabled
0	WGP 0	CU 0–1	1	1
1	WGP 1	CU 2–3	1	1
2	WGP 2	CU 4–5	1	1
3	WGP 3	CU 6–7	0	1
4	WGP 4	CU 8–9	0	1

Stock value: 0x7 (WGPs 0–2 enabled). Unlocked value: 0x1F (all 5 WGPs enabled).

RLC_PG_ALWAYS_ON_WGP_MASK

Must be set to 0x1F concurrently with SPI_PG_ENABLE_STATIC_WGP_MASK to prevent the Run List Controller from power-gating re-enabled WGPs during idle periods.

B. Harvest Pattern Visualization

Per-SA CU layout showing the dominant (74%) contiguous harvest pattern:

CU index:	0	1	2	3	4	5	6	7	8	9
	[*]	[*]	[*]	[*]	[*]	[*]	[]	[]	[]	[]
	--- active (stock) ---					--- fused ---				
	WGP 0		WGP 1		WGP 2		WGP 3		WGP 4	

Non-contiguous variants observed in ~26% of boards (examples from community survey):

Variant A:	[*]	[*]	[]	[]	[*]	[*]	[*]	[*]	[]	[]
Variant B:	[]	[]	[*]	[*]	[*]	[*]	[*]	[*]	[]	[]
Variant C:	[*]	[*]	[*]	[*]	[]	[]	[*]	[*]	[]	[]

All variants maintain exactly 6 active CUs (3 active WGPs) per shader array. After re-enablement, all 10 CUs per SA are active and dispatching compute work regardless of the original harvest pattern.

C. Enable Procedure

```
# 1. Build patched amdgpu module with bc250_cc_write_mode parameter
# (see https://github.com/duggasco/bc250-40cu-unlock)

# 2. Install one-shot boot config:
echo 'install amdgpu /bin/sh -c '\''rm -f /etc/modprobe.d/bc250-cc-oneshot.conf; \
exec /sbin/modprobe --ignore-install amdgpu bc250_cc_write_mode=3'\'' \
> /etc/modprobe.d/bc250-cc-oneshot.conf

# 3. Reboot

# 4. Set SPI dispatch mask (requires umr):
umr -w cyan_skillfish.gfx1013.mmSPI_PG_ENABLE_STATIC_WGP_MASK 0x1f
umr -w cyan_skillfish.gfx1013.mmRLC_PG_ALWAYS_ON_WGP_MASK 0x1f

# 5. Verify:
RADV_DEBUG=info vulkaninfo 2>&1 | grep cu_mask
# Expected: cu_mask = 0x3ff (all SAs)
```