signal65



Azure Cobalt 200 Outperforms Cobalt 100 Per Instance and Per Core

AUTHORS

Jonathan Fellows
Performance Validation Engineer | Signal65

Ryan Shrout
President and GM | Signal65

COMMISSIONED BY

Microsoft

NOVEMBER 2026

Executive Summary

Signal65 testing across developer, agentic AI and HPC workloads shows Azure Cobalt 200 delivering 1.43x to 1.93x the throughput of Cobalt 100 per instance and 1.07x to 1.45x per vCPU, with up to 5.5x per instance in Microsoft's own chip-design evaluation.

Key Highlights

- **1.43x to 5.53x per instance**, from molecular dynamics to Microsoft's RTL simulation tests
- **1.76x the work per instance** on developer and system tasks (Terminal-Bench 2)
- **1.59x to 1.93x function-calling throughput** per instance, with identical accuracy
- **3.43x to 5.53x RTL simulation jobs per hour** per instance in Microsoft's own chip-design evaluation

Azure Cobalt 200 is the second generation of Microsoft's custom Arm-based server processor. By designing a custom CPU offering like Cobalt, Microsoft can tune it for the workloads Azure customers run and engineer it together with the servers, software and services the company offers around it. Customers are also asking general-purpose instances to carry more of their work, from software builds and data processing to AI agents and scientific simulation. The Cobalt 200 instance tested here carries 128 Arm Neoverse V3 cores with 3 MB of L2 cache per core, compared with 96 Neoverse N2 cores and 1 MB of L2 cache per core on Cobalt 100.

Signal65, an independent AI benchmarking and validation firm, ran both generations through developer and system tasks, function calling and molecular dynamics. We also reviewed chip-design results that Microsoft's silicon engineering team produced on its own design flows.

Cobalt 200 led Cobalt 100 on all seven Signal65 results in this report, delivering between 1.43x and 1.93x the throughput per instance. The Cobalt 200 instance has 1.33x the vCPUs of its predecessor, so each Cobalt 200 vCPU also completed more work, by 1.07x to 1.45x. In Microsoft's chip-design evaluation, a fully loaded Cobalt 200 instance completed up to 5.5x as many RTL simulation jobs per hour as a Cobalt 100 instance.

Key Findings

- **Developer and system tasks:** Cobalt 200 completed 1.76x the work per instance across the Terminal-Bench 2 task suite and ran each task 1.32x as fast per vCPU.
- **Agentic AI:** function-calling throughput rose 1.59x to 1.93x per instance (1.20x to 1.45x per vCPU) with identical accuracy.
- **Scientific computing:** GROMACS molecular dynamics rose 1.43x to 1.46x per instance (1.07x to 1.09x per vCPU).
- **Chip design:** in Microsoft's own tests, reviewed by Signal65, a fully loaded Cobalt 200 instance completed 3.43x to 5.53x as many RTL logic simulation jobs per hour as a Cobalt 100 instance (2.58x to 4.15x per vCPU).

All Signal65 results compare the largest general-purpose VM of each generation, Standard_D128pds_v7 against Standard_D96pds_v6, at matched memory per vCPU. Cobalt 200 was tested in opportunistic mode, a deployment option that lets its cores scale up to 3.6 GHz. This report covers compute-intensive workloads that scale across the full instance, and Signal65 plans to extend the comparison to additional workloads in future testing.

© 2026 Signal65. All rights reserved.

Why Microsoft Invests in Cobalt

Microsoft introduced Cobalt 100 in 2023 as its first custom server processor and followed with Cobalt 200 in 2025. At the scale of Azure, even incremental improvements in performance, power efficiency and server utilization can have a substantial impact on infrastructure costs. Controlling the processor design gives Microsoft another way to optimize those economics while tailoring its hardware roadmap to the needs of its cloud platform.

Microsoft designed Cobalt 200 around the work its customers run, using workload data from Azure customers to develop more than 140 benchmark variants covering databases, web servers, storage caches, network transactions and analytics. The company evaluated more than 350,000 system configuration candidates to inform its design choices. The resulting processor combines 132 Arm Neoverse V3 cores manufactured on TSMC's 3 nm process with a larger cache hierarchy, custom compression and cryptography accelerators, memory encryption enabled by default, and independent power management for each core. These decisions reflect a focus on throughput, latency, efficiency and the characteristics of workloads running at cloud scale.

Microsoft also designs Cobalt 200 as part of a broader infrastructure platform that incorporates Azure Boost for networking and storage acceleration, the Azure Integrated HSM for security, and software optimizations informed by its own services and customer workloads. That integration lets Microsoft improve performance and efficiency across the whole system, with the CPU as one part of it. Our testing measures how those design choices perform in real applications.

Understanding the Results

Two Views of Every Result

Each result is reported per instance and per vCPU. Per instance compares the whole VMs, 128 vCPUs on Cobalt 200 against 96 on Cobalt 100, and shows what a customer gets from the largest VM of each generation. Per vCPU divides that ratio by 1.33, the difference in vCPU count, and shows the gain from each core. The per-vCPU figure compares one Cobalt 200 vCPU with one Cobalt 100 vCPU, so it reflects the newer core, the larger caches, the higher clock and the other microarchitectural improvements in Cobalt 200 together.

Every Signal65 workload in this report runs across the full instance on both VMs, which makes the division a fair per-core measure, and every performance figure in the report is labeled as per instance or per vCPU. Microsoft's RTL simulation throughput tests also filled both VMs with concurrent jobs, so those results are reported the same way. The remaining chip-design results compare a single job on each generation at the same thread count and are labeled per job.

Uplift varies widely by workload, from 1.43x per instance on molecular dynamics to 5.53x per instance in Microsoft's RTL simulation tests, and from 1.07x to 4.15x per vCPU. The gain a customer sees depends on how its workload uses cores, cache and memory. We encourage customers to run their own workloads on both Cobalt 100 and Cobalt 200 to measure the improvement that applies to them.

© 2026 Signal65. All rights reserved.

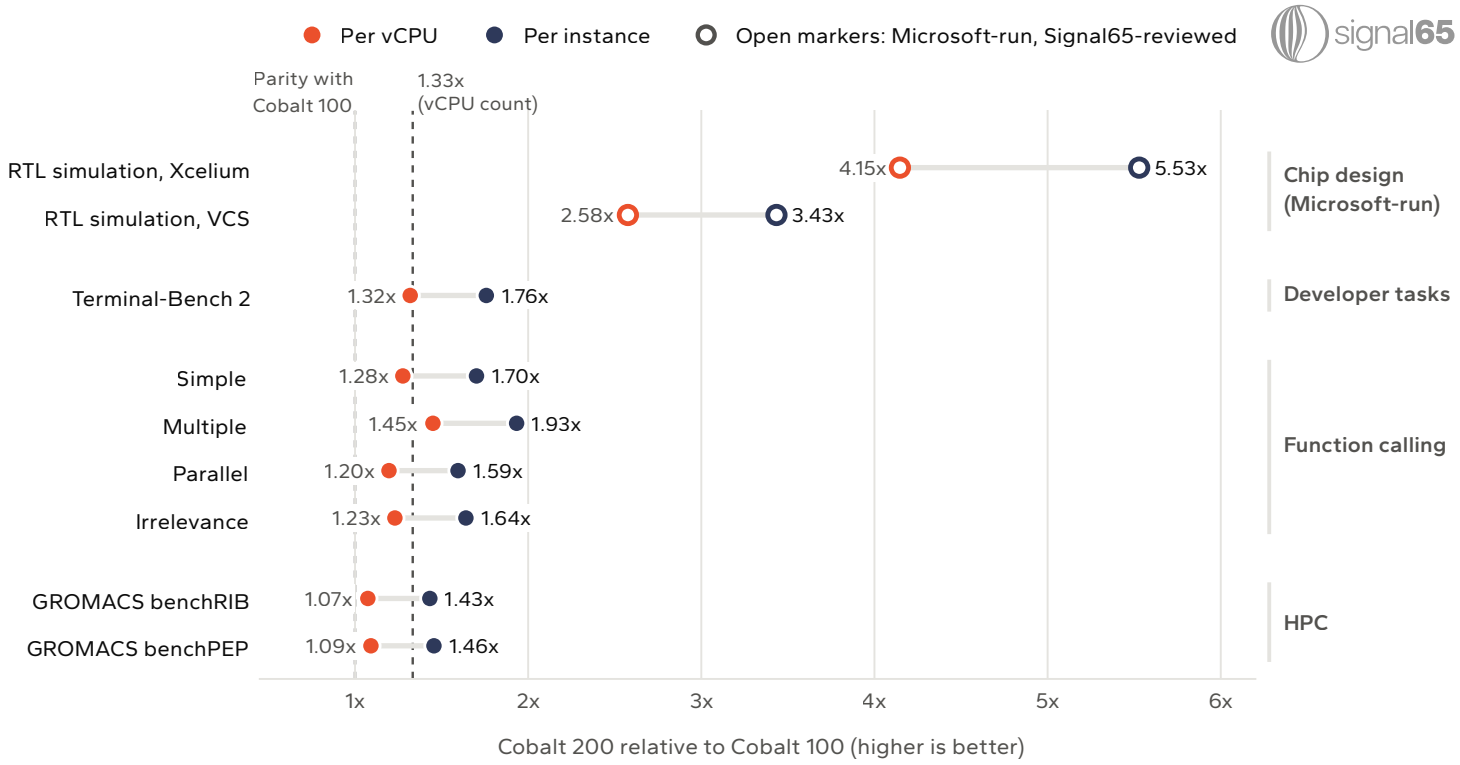


Figure 1. *Cobalt 200 relative to Cobalt 100, per instance and per vCPU. The seven Signal65 results range from 1.43x to 1.93x per instance and from 1.07x to 1.45x per vCPU. Microsoft's RTL simulation throughput results, shown with open markers, reach 3.43x and 5.53x per instance. Higher is better.*

Tested Configurations

Specification	Cobalt 100	Cobalt 200
Azure VM size	Standard_D96pds_v6	Standard_D128pds_v7
vCPUs	96	128
Memory	377 GiB	502 GiB
Processor core	Arm Neoverse N2	Arm Neoverse V3
L2 cache per core	1 MB	3 MB
L3 cache	One 128 MB instance	Two 96 MB instances
Clock measured during testing	3.4 GHz	3.6 GHz (opportunistic mode)
Region	Central US, zone 3	Central US, zone 3

Both VMs ran the same Linux kernel and identical software versions, and every result is the median of three complete runs on each VM.

Cobalt 200 VMs default to deterministic mode, which holds cores at 3.2 GHz. Customers can select opportunistic mode when they deploy a VM, which lets cores scale up to 3.6 GHz depending on the power draw of the workloads on the node. All Cobalt 200 results in this report were measured in opportunistic mode, and our clock checks before and after every benchmark block read 3.6 GHz on one core and on all 128 vCPUs.

Memory bandwidth across the full instance rose between 1.22x and 1.53x in our STREAM-style tests, depending on the access pattern.

© 2026 Signal65. All rights reserved.

Developer and System Tasks

Terminal-Bench 2 is a suite of containerized tasks that compile software, boot virtual machines, build and query databases, configure web servers, recover archive passwords and process data. Each task runs its reference solution, which makes the suite a measure of CPU and container execution across a broad set of common server work. We scored 88 tasks on both systems.

With download time removed, Cobalt 200 completed each task 1.32x as fast as Cobalt 100 on a single vCPU, measured as the median across the 32 tasks that log their download time. Each copy of a task holds one vCPU, so this is the per-core result. Running one copy per vCPU across the whole instance, 128 on Cobalt 200 against 96 on Cobalt 100, the new generation completed 1.76x the work.

Measure	Cobalt 200 vs. Cobalt 100
Per task with download time removed (per vCPU)	**1.32x**
Per instance, one task per vCPU	**1.76x**
Measured suite throughput with downloads included (per instance)	1.40x

Package downloads and container setup do not speed up with the CPU, and with those included the suite's measured throughput rose from 0.258 to 0.363 trials per second, or 1.40x per instance. Jobs that spend much of their time downloading will land between 1.40x and 1.76x per instance.

Results for individual tasks varied widely per vCPU, with virtual machine boot under QEMU running 3.7x to 3.8x as fast, file compression 1.64x and archive password recovery 1.61x, while a small number of tasks ran slower per copy on Cobalt 200.

Chip Design EDA Performance

The results in this section were generated by Microsoft's silicon engineering team on Microsoft-internal chip designs, using Cobalt 200 preview and pre-production systems with tools from Synopsys, Cadence and Siemens. The tests ran on full-size VMs, with 128 vCPUs on Cobalt 200 and 96 on Cobalt 100. Signal65 did not run these tests. We reviewed Microsoft's raw run logs and confirmed the results summarized here.

Chip design is among the most demanding and license-intensive workloads in the cloud, since commercial simulation and verification tools are typically licensed per concurrent job. With a single job running on each VM, RTL logic simulation, the largest consumer of compute in most chip projects, completed about twice as fast per job on Cobalt 200. Static timing analysis ran 1.7x to 1.9x as fast per job, gate-level simulation 1.4x to 1.7x, design-rule checking 1.3x to 1.4x and analog circuit simulation 1.2x to 1.3x.

© 2026 Signal65. All rights reserved.

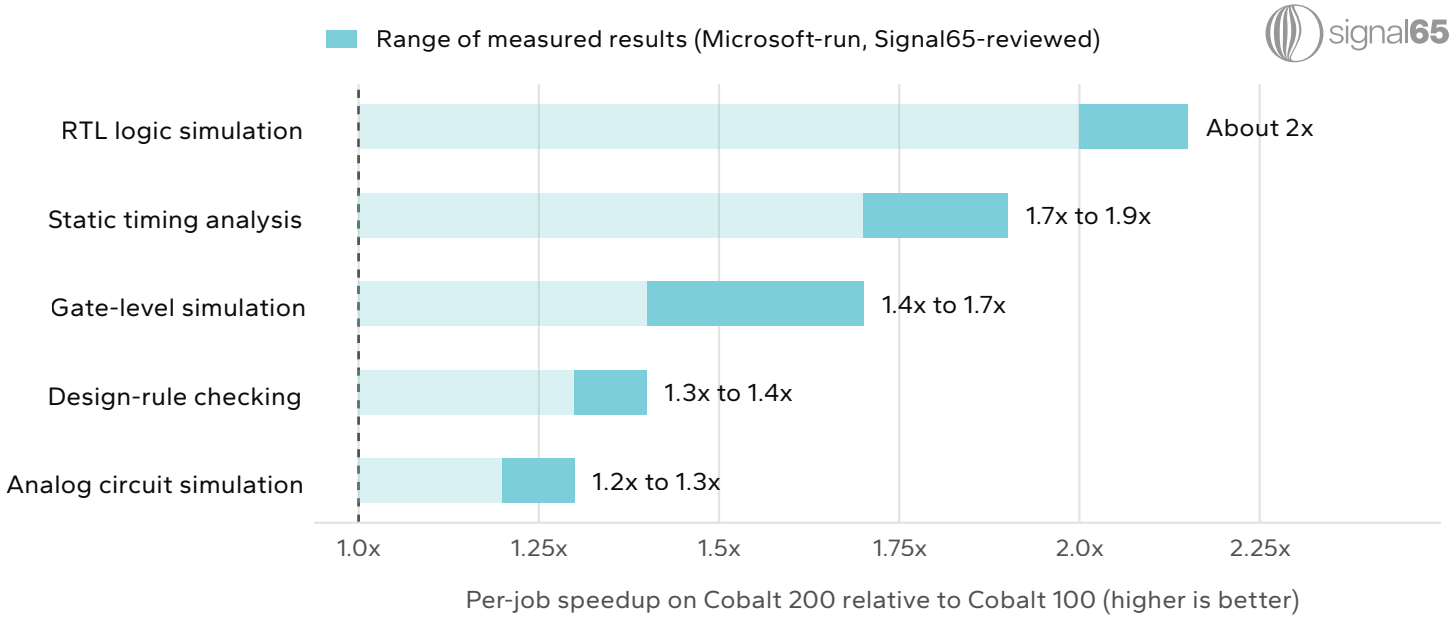


Figure 2. *Per-job speedup on Cobalt 200 by chip-design tool category, with a single job running on each VM. Microsoft-run, Signal65-reviewed. Higher is better.*

Verification teams run RTL simulation as a regression farm that fills every core with single-threaded jobs. Microsoft tested that case with one simulation job per vCPU on each VM, 128 on Cobalt 200 against 96 on Cobalt 100. The Cobalt 200 instance completed 3.43x as many jobs per hour with Synopsys VCS and 5.53x as many with Cadence Xcelium, which works out to 2.58x and 4.15x per vCPU.

RTL simulation throughput at full load	Per instance	Per vCPU
Synopsys VCS	**3.43x**	2.58x
Cadence Xcelium	**5.53x**	4.15x

Cobalt 100 slowed far more than Cobalt 200 as the Xcelium workload filled the instance. Each job took 94% longer on a fully loaded Cobalt 100 instance than on a lightly loaded one, compared with 17% longer on Cobalt 200. With VCS the two generations slowed by similar amounts, 15% and 17%. These throughput runs used a later pre-production Cobalt 200 system than the single-job RTL runs, so the two sets of results are reported separately. Each job holds a license for as long as it runs, so faster jobs let every license complete more verification work.



Agentic AI: Function Calling

Agentic applications call a language model to decide which tool or function to run, and that step often runs on the same CPU instances that host the application.

We ran the Berkeley Function Calling Leaderboard (BFCL) v4 tests, with 150 prompts in each of four categories and the Qwen3-8B model served through vLLM. Cobalt 200 processed 1.59x to 1.93x as many requests per second per instance as Cobalt 100, or 1.20x to 1.45x per vCPU. Accuracy was identical on both platforms in every category, because the same model at the same settings returns the same answers.

© 2026 Signal65. All rights reserved.

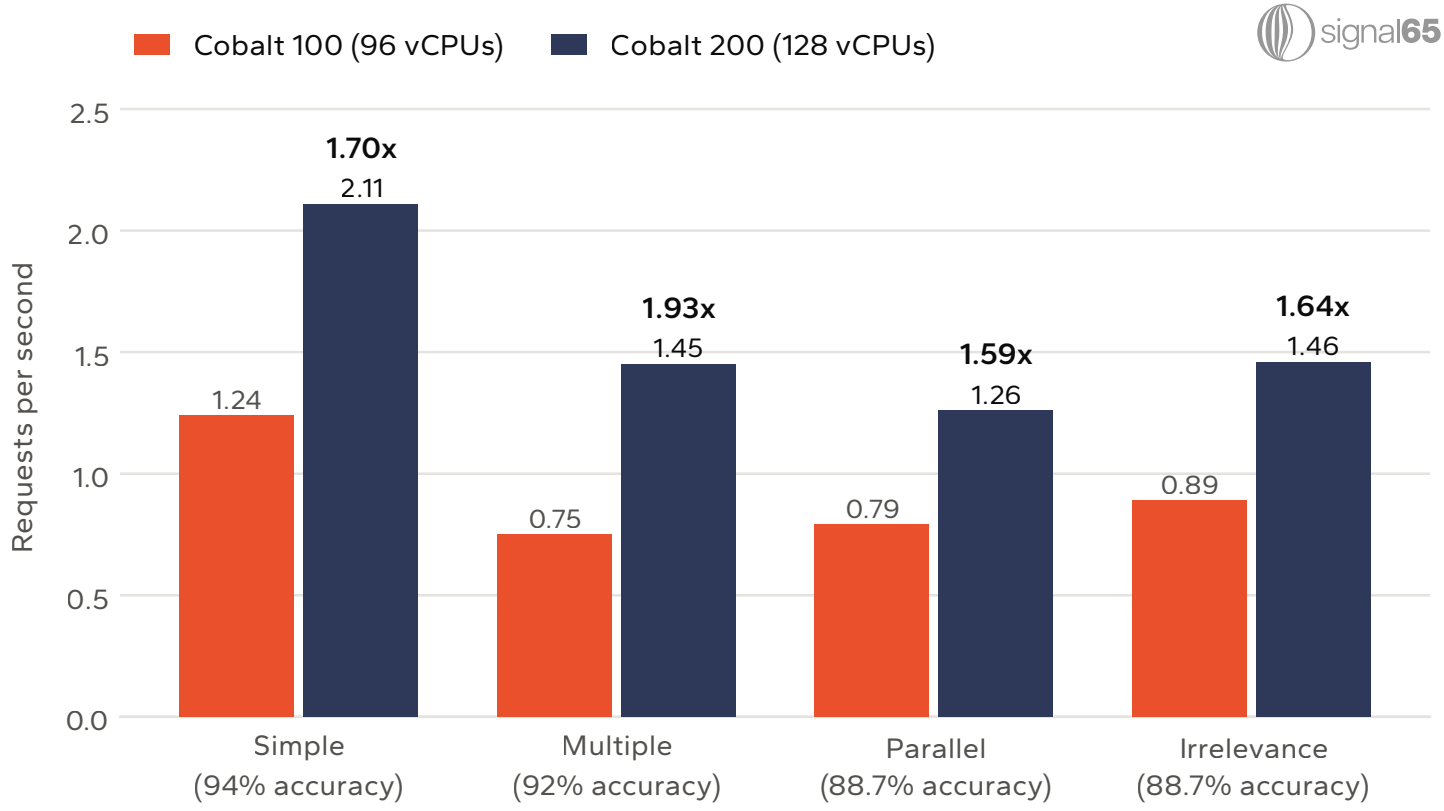


Figure 3. *BFCL v4 function-calling throughput in requests per second by test category, with the accuracy measured on both platforms. Ratios above the bars are per instance. Higher is better.*

Scientific Computing

GROMACS is one of the most widely used molecular dynamics packages in research and industry, and it makes heavy use of vector (SIMD) instructions, so it serves as this report's representative for vector-heavy technical computing. We used a build with the GROMACS SVE SIMD option enabled and tuned each platform to its best rank and thread layout. Cobalt 200 simulated 1.43x as many nanoseconds per day per instance on the 2-million-atom benchRIB model and 1.46x as many on the 12-million-atom benchPEP model. Both per-instance gains exceed the 1.33x increase in vCPU count, so each Cobalt 200 core also completed more simulation work, by 1.07x and 1.09x per vCPU.

Model	Cobalt 100 (ns/day)	Cobalt 200 (ns/day)	Per instance	Per vCPU
benchRIB, 2 million atoms	6.78	9.71	**1.43x**	1.07x
benchPEP, 12 million atoms	0.544	0.792	**1.46x**	1.09x

© 2026 Signal65. All rights reserved.

Consolidation

Each Cobalt 200 instance completed the work of 1.43 to 1.93 Cobalt 100 instances across the workloads Signal65 tested, including 1.76 on the developer task suite, 1.59 to 1.93 on function calling and 1.43 to 1.46 on GROMACS. A fleet sized for a fixed load on those workloads therefore needs roughly 30% to 48% fewer instances.

Chip-design verification shows a larger reduction, since each Cobalt 200 instance completed the work of 3.43 to 5.53 Cobalt 100 instances in Microsoft's RTL simulation tests. A simulation farm sized for a fixed regression load would need roughly 71% to 82% fewer instances.

Deployment Guidance

Default settings in several of the software stacks we tested underused these full-node VMs.

- **Opportunistic mode:** Cobalt 200 VMs default to deterministic mode at 3.2 GHz. Selecting opportunistic mode at deployment lets cores scale up to 3.6 GHz, which is the configuration tested here.
- **NUMA placement:** VMs larger than 64 vCPUs present two virtual NUMA nodes, and Cobalt 200 maps them to physical NUMA nodes. Each process should keep its cores and memory on the same node.
- **Worker layout:** software that places one worker on each NUMA node uses the whole instance. vLLM with tensor parallelism of 2 engaged all 128 vCPUs, where its default configuration engaged 63.
- **Application tuning:** GROMACS configurations tuned on each platform beat the built-in auto-tuner on both generations.

© 2026 Signal65. All rights reserved.

Conclusion and Recommendations

Cobalt 200 demonstrates the value of Microsoft's continued investment in custom server silicon. Across developer and system tasks, function calling and scientific simulation, Cobalt 200 delivered 1.43x to 1.93x the throughput per instance of Cobalt 100 while exceeding its predecessor on a per-vCPU basis in every test. Microsoft's own chip-design evaluation showed larger gains, with a fully loaded Cobalt 200 instance completing up to 5.5x as many logic simulation jobs per hour.

Microsoft is building Cobalt as a CPU platform shaped by the characteristics of the workloads running in Azure, with control over processor architecture, system integration and software optimization. The transition from Neoverse N2 to V3, the move to a 3 nm manufacturing process, expanded cache capacity and additional compute resources represent substantial architectural improvements, and the performance gains across different kinds of applications indicate that Microsoft's approach is delivering broad benefits.

For a fixed amount of work, the throughput improvements measured in our testing translate to approximately 30% to 48% fewer VM instances required, assuming comparable scaling and utilization. Microsoft's RTL simulation results point to a larger reduction for verification farms, at roughly 71% to 82% fewer instances. We did not measure cost or power consumption, so those may not fall by the same amount, but fewer instances give customers room to reduce deployment footprints or increase the work completed within an existing resource budget. At cloud scale, improvements of this size are central to the economic argument for custom silicon.

For organizations already running on Cobalt 100, Cobalt 200 should be a priority for evaluation, particularly for compute-intensive general-purpose applications, AI inference and agentic workloads, scientific computing and chip-design verification. Teams planning new Arm-based deployments on Azure should consider Cobalt 200 as a starting point, while validating performance, compatibility and cost against their specific requirements. The results presented here reflect the VM sizes, software configurations and opportunistic-mode settings tested, and actual gains will vary by workload. Customers should run their own workloads on both generations to confirm the improvement, and Signal65 plans to extend this comparison to additional workloads in future testing.

Cobalt 200 shows that Microsoft's custom silicon strategy is maturing into an important part of Azure's competitive infrastructure. Microsoft is using its control of the processor design to deliver measurable improvements in the performance and efficiency of its cloud platform.

© 2026 Signal65. All rights reserved.

Appendix A: Results

Signal65-tested workloads

Signal65 results on Standard_D96pds_v6 (Cobalt 100) and Standard_D128pds_v7 (Cobalt 200, opportunistic mode). Values are medians of three complete runs on each VM. Per vCPU is the per-instance ratio divided by 1.33. Higher is better in every row.

Workload	Test	Metric	Cobalt 100	Cobalt 200	Per instance	Per vCPU
Developer tasks	Terminal-Bench 2, download time removed	Relative task speed	1.00	1.32	1.76x	1.32x
Function calling	BFCL v4 simple	Requests/s	1.24	2.11	1.70x	1.28x
Function calling	BFCL v4 multiple	Requests/s	0.75	1.45	1.93x	1.45x
Function calling	BFCL v4 parallel	Requests/s	0.79	1.26	1.59x	1.20x
Function calling	BFCL v4 irrelevance	Requests/s	0.89	1.46	1.64x	1.23x
HPC	GROMACS benchRIB	ns/day	6.78	9.71	1.43x	1.07x
HPC	GROMACS benchPEP	ns/day	0.544	0.792	1.46x	1.09x

The Terminal-Bench 2 per-instance figure multiplies the per-task speed by the copies each VM runs at once (128 against 96).

Microsoft-run workloads, reviewed by Signal65

Chip-design results generated by Microsoft on 96-vCPU Cobalt 100 and 128-vCPU Cobalt 200 VMs and reviewed by Signal65. Throughput tests ran one single-threaded simulation job per vCPU, and the other tests ran a single job on each VM at the same thread count. For run times, lower is better and the ratio divides the Cobalt 100 time by the Cobalt 200 time. Static timing and design-rule checking ran on the same pre-production Cobalt 200 system as the throughput tests.

Workload	Test	Metric	Cobalt 100	Cobalt 200	Cobalt 200 vs. Cobalt 100
RTL simulation throughput	Synopsys VCS, one job per vCPU	Jobs/hour	285.5	980.5	3.43x per instance, 2.58x per vCPU
RTL simulation throughput	Cadence Xcelium, one job per vCPU	Jobs/hour	257.2	1,422.1	5.53x per instance, 4.15x per vCPU
RTL logic simulation	Synopsys VCS, single job	Run time (s)	2,040	951	2.15x per job
RTL logic simulation	Cadence Xcelium, single job	Run time (s)	1,336	627.5	2.13x per job
Static timing analysis	Synopsys PrimeTime, 1 core	Run time (s)	1,674	889.5	1.88x per job

© 2026 Signal65. All rights reserved.

Workload	Test	Metric	Cobalt 100	Cobalt 200	Cobalt 200 vs. Cobalt 100
Static timing analysis	Synopsys PrimeTime, 4 cores	Run time (s)	553	305	1.81x per job
Static timing analysis	Synopsys PrimeTime, 8 cores	Run time (s)	347	207	1.68x per job
Gate-level simulation	Synopsys VCS, single job	CPU time (s)	676	470	1.44x per job
Gate-level simulation	Cadence Xcelium, single job	Run time (s)	352	206	1.71x per job
Design-rule checking	Siemens Calibre, 16 cores	Run time (s)	1,803	1,392	1.30x per job
Design-rule checking	Siemens Calibre, 4 cores	Run time (s)	4,528	3,270	1.38x per job
Analog circuit simulation	Cadence Spectre, 16 threads	Run time (s)	1,710	1,330	1.29x per job

Appendix B: Calculations and Test Environment

Calculations

Measure	Calculation	Result
Per vCPU	Per-instance ratio / (128 / 96)	1.07x to 1.45x
Developer tasks, per instance	1.32 × (128 / 96)	1.76x
Fewer instances for a fixed load, Signal65-tested workloads	1 – 1 / per-instance ratio, at 1.43x and 1.93x	30% to 48%
Fewer instances for a fixed RTL simulation load (Microsoft-run)	1 – 1 / per-instance ratio, at 3.43x and 5.53x	71% to 82%
Chip design RTL simulation, per instance (Microsoft-run)	(128 / mean job run time on Cobalt 200) / (96 / mean job run time on Cobalt 100), one job per vCPU	3.43x and 5.53x

Test environment

- **Software:** Linux kernel 6.17.0-1022-azure, vLLM 0.28.0, PyTorch 2.13.0 and GROMACS 2026.3 on both VMs. GROMACS was built with its SVE SIMD option (GMX_SIMD=ARM_SVE), and vLLM ran on the standard PyTorch CPU build.
- **Function calling:** Qwen3-8B served through vLLM at concurrency 8, with both VMs run on the same day.
- **Memory bandwidth:** full-instance results from our STREAM-style test were 394 → 603 GB/s for copy (1.53x), 411 → 502 GB/s for triad (1.22x) and 443 → 560 GB/s for read (1.26x).
- **Chip design:** Microsoft ran Synopsys VCS and PrimeTime, Cadence Xcelium and Spectre, and Siemens Calibre and AFS on its own designs. The RTL simulation throughput tests ran one single-threaded job per vCPU on each VM. Signal65 reviewed the run logs.

© 2026 Signal65. All rights reserved.

Important Information About this Report

CONTRIBUTORS

Jonathan Fellows
Performance Validation Engineer | Signal65

Ryan Shrout
President and GM | Signal65

PUBLISHER

Ryan Shrout
President and GM | Signal65

INQUIRIES

Contact us if you would like to discuss this report and Signal65 will respond promptly.

CITATIONS

This paper can be cited by accredited press and analysts, but must be cited in-context, displaying author's name, author's title, and "Signal65." Non-press and non-analysts must receive prior written permission by Signal65 for any citations.

LICENSING

This document, including any supporting materials, is owned by Signal65. This publication may not be reproduced, distributed, or shared in any form without the prior written permission of Signal65.

DISCLOSURES

Signal65 provides research, analysis, advising, and lab services to many high-tech companies, including those mentioned in this paper.

ABOUT SIGNAL65

Signal65 exists to be a source of data in a world where technology markets and product landscapes create complex and distorted views of product truth. We strive to provide honest and comprehensive feedback and analysis for our clients in order for them to better understand their own competitive positioning and create optimal opportunities to market and message their devices and services.

COMMISSIONED BY

Microsoft

CONTACT INFORMATION
Signal65 | signal65.com

© 2026 Signal65. All rights reserved.