

AI Processor Management Information Tracker





Copyright Jon Peddie Research 2026. All rights reserved.

Reproduction in whole or in part is prohibited without written permission
from Jon Peddie Research.

This report is the property of Jon Peddie Research (JPR) and made available to a restricted number of clients only upon these terms and conditions.

Agreement not to copy or disclose. This report and all future reports or other materials provided by JPR pursuant to this subscription (collectively, "Reports") are protected by: (i) federal copyright, pursuant to the Copyright Act of 1976; and (ii) the nondisclosure provisions set forth immediately following.

License, exclusive use, and agreement not to disclose. Reports are the trade secret property exclusively of JPR and are made available to a restricted number of clients, for their exclusive use and only upon the following terms and conditions. JPR grants site-wide license to read and utilize the information in the Reports, exclusively to the initial subscriber to the Reports, its subsidiaries, divisions, and employees (collectively, "Subscriber"). The Reports shall, at all times, be treated by Subscriber as proprietary and confidential documents, for internal use only. Subscriber agrees that it will not reproduce for or share any of the material in the Reports ("Material") with any entity or individual other than Subscriber ("Shared Third Party") (collectively, "Share" or "Sharing"), without the advance written permission of JPR.

No material from JPR may be used for AI training. The term "AI training" refers to the process of teaching a machine learning model to make predictions or decisions based on data. That includes but is not limited to: 1. Supervised Learning: Training with labeled data (input-output pairs), such as image classification, spam detection, sentiment analysis. 2. Unsupervised Learning: Training with unlabeled data to find patterns or groupings, such as clustering (e.g. K-means), dimensionality reduction (e.g. PCA). 3. Self-Supervised Learning: A hybrid where the system generates labels from the data itself, such as contrastive learning, masked language modeling (used in GPT, BERT). 4. Reinforcement Learning (RL): Agents learn by taking actions and receiving rewards or penalties, such as game-playing AIs (e.g. AlphaGo), robotics, navigation. 5. Unsupervised Reinforcement Learning: Agents explore environments without a defined task or reward, such as Pretraining agents for general skills before fine-tuning. 6. Multi-Task Learning: Training one model to handle multiple tasks simultaneously, such as a model that can translate and summarize text. 7. Transfer Learning: Using a pre-trained model on a new, related task with minimal retraining, such as fine-tuning GPT or BERT on a specific domain. 8. Few-Shot / One-Shot / Zero-Shot Learning, Few-shot: Trained on a very small number of examples. One-shot: Trained on just one example. Zero-shot: Performs tasks without any task-specific training. 9. Meta Learning ("Learning to Learn"): Training models that can adapt to new tasks with very little data, such as Model-Agnostic Meta-Learning (MAML). 10. Curriculum Learning: Training models by gradually increasing task difficulty. 11. Online Learning: Model updates continuously as new data arrives, such as stock prediction systems. 12. Federated Learning: Training across decentralized devices without sharing raw data, such as mobile keyboard prediction models. 13. Contrastive Learning: Learns by comparing positive and negative pairs of data. Used in: Self-supervised vision models like SimCLR, CLIP. Specialized AI Training Terms, Fine-tuning: Adjusting a pre-trained model on a new dataset. Pretraining: Initial broad training, usually on large datasets (e.g. GPT pretraining). Prompt Tuning / Instruction Tuning: Adjusting models using instructions or task-specific prompts. RLHF (Reinforcement Learning from Human Feedback): Used in GPT models to align behavior with human intent. Distillation: Training a smaller model to mimic a larger model. Active Learning: Selecting the most informative data points to label next. Semi-supervised Learning: Mix of labeled and unlabeled data for training.

Subscriber shall be liable for any breach of this agreement and shall be subject to cancellation of its subscription to Reports. Without limiting this liability, Subscriber shall be liable for any damages suffered by JPR as a result of any Sharing of any Material, without advance written permission of JPR.

Important Note related to Sharing of Material: If Subscriber wishes to Share any Material with any Shared Third Party, it shall contact JPR and request permission to do so. JPR shall forthwith prepare an appropriate Sharing agreement, which shall set forth the name and contact information of the Shared Third Party, the terms and conditions of such Sharing, the compensation to JPR for such Sharing, and the agreement by Subscriber and the Shared Third Party that the Shared Third Party shall not, under any circumstances, Share any of the Material with any other entity or person.

Disclaimers. JPR Reports contain "reviews" of various products and information. The content of these Reports represents the interpretation and analysis of statistics and information that are either generally available to the public or released by entities or individuals deemed responsible. The information in the Reports is believed to be reliable but is not guaranteed as to its accuracy or completeness. The Reports do not endorse or attempt to sell any products and should not be interpreted differently. JPR assumes no responsibility for the correct or incorrect usage of any trademarks or service marks.

Table of Contents

How the AI Processor information management tracker is organized.	1
Table of Contents	3
Table of Figures	3
Table of Tables	4
How the AI Processor information management tracker is organized	5
The challenge of Taxonomy	5
Cross-cutting classifications	6
AI wearables	7
Device inference	8
Why AI smartphones and PCs are Device inference	9
How AI changes the Device inference definition	9
Autonomous Systems.....	10
AI Training.....	12
AI Cloud inference.....	13
The AIP zoo	15
GPU (graphics processing unit)	16
NPU (neural processing unit).....	17
Dataflow processor	18
CIM (compute-in-memory)	18
Neuromorphic processor	19
RISC-V processor	20
x86 CPU.....	20
Arm CPU	21
Hybrid processor	22
Analog-based processor	22
Appendix.....	24

Table of Figures

Figure 1. AI Processor Tracker market-segment taxonomy.....	5
Figure 2. All the IoT things one can wear	7
Figure 3. Device inference hardware—AI PC, smartphone, or XR headset running local models	9

Figure 4. Autonomous Systems hardware—AV, drone, AMR, or humanoid robot in operation...	11
Figure 5. AI Training hardware—GPU cluster, TPU pod, or wafer-scale training system	12
Figure 6. AI Cloud inference hardware—rack-scale inference server or hyperscaler GPU bank	14
Figure 7. GPU generalized block diagram.....	17
Figure 8. NPU generalized block diagram.....	17
Figure 9. Dataflow generalized block diagram.....	18
Figure 10. CIM generalized block diagram	19
Figure 11. Neuromorphic generalized block diagram	19
Figure 12. RISC-V generalized block diagram.....	20
Figure 13. x86 generalized block diagram.....	21
Figure 14. Arm CPU generalized block diagram.....	21
Figure 15. Hybrid generalized block diagram	22
Figure 16. Analog generalized block diagram	23

Table of Tables

Table 1. IoT subsegments.....	24
-------------------------------	----

How the AI Processor information management tracker is organized

We first subdivide the industry into five major market segments: Autonomous Systems, IoT, AI Training, AI Cloud Inference, and Device Inference.

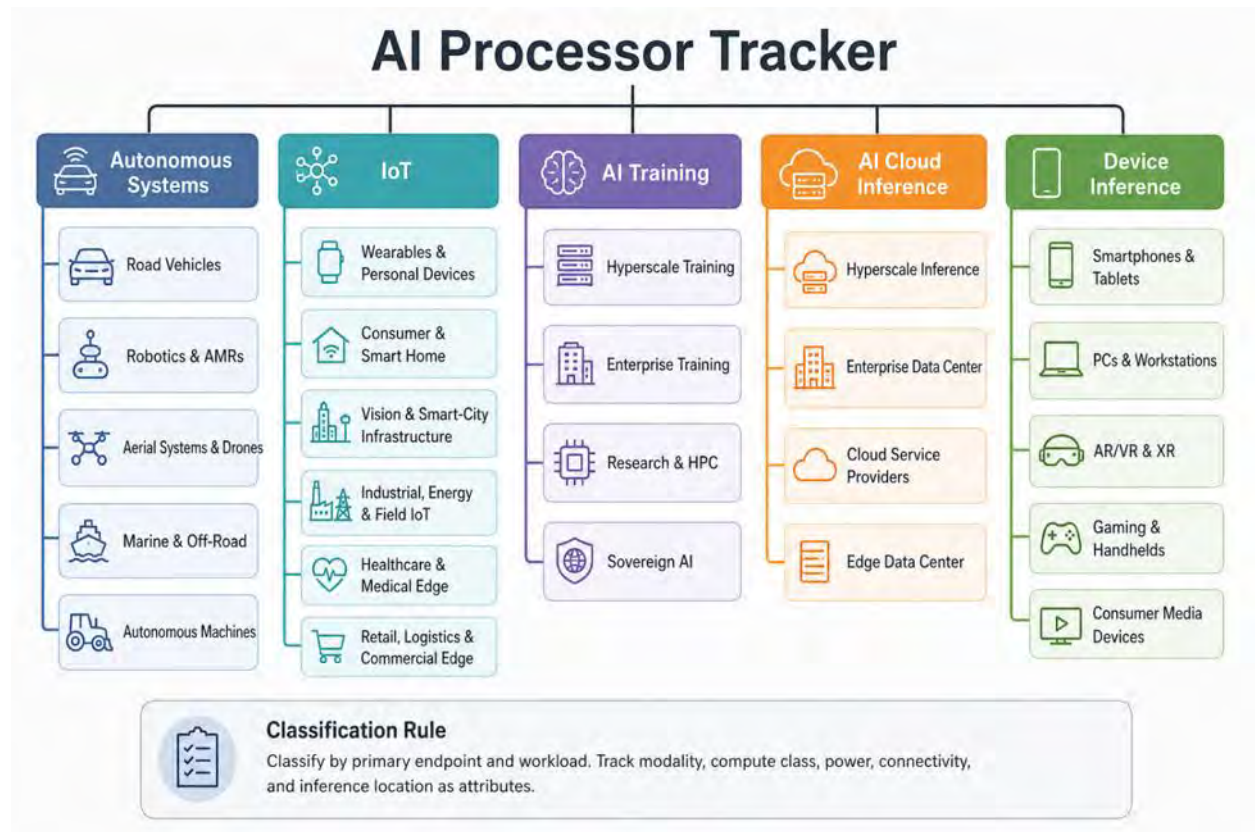


Figure 1. AI Processor Tracker market-segment taxonomy

The framework divides AI processors into five primary markets—Autonomous Systems, IoT, AI Cloud Inference, Device Inference, and AI Training—then assigns each market to endpoint- and workload-specific subcategories. It distinguishes processors by their primary deployment role, while capturing compute class, modality, power source, connectivity, and inference location as cross-cutting attributes.

The challenge of Taxonomy

The difficult classification cases merit explicit rules:

- **Autonomous Systems:** AV cars, drones, AMRs, humanoids, autonomous tractors, and autonomous warehouse vehicles are in Autonomous Systems when perception, navigation, or safety control drives the design.
- **IoT:** This segment has the most devices in it. Smart cameras and machine-vision systems are in IoT, even when they use powerful AI SoCs, because they operate as fixed sensing

endpoints. Smart cameras process tracking, scene changes, object recognition, and text locally across retail, manufacturing, security, and smart-city deployments. Edge gateways are in IoT when they aggregate sensor data and execute local analytics, protocol translation, or limited inference.

- **AI Cloud Inference:** Regional or rack-scale servers are in the AI Cloud inference segment, as well as the hyperscalers that swap out workloads across their banks of GPUs, mostly used for training. Also, separate edge-server overlay if they serve multiple sites.
- **Device Inference:** This segment includes smartphones, tablets, PCs, and consumer client devices in Device inference, even when they control nearby IoT endpoints.
- **AI Training:** This segment gets the most press attention and includes accelerators and supporting processors deployed primarily to create, pre-train, fine-tune, retrain, or evaluate AI models. It covers hyperscale training clusters, enterprise and private-cloud training systems, research and HPC installations, and sovereign AI infrastructure where an organization or government keeps training data, models, and operational control within a defined jurisdiction. Classify a system as AI Training when model development consumes the majority of its compute time, even if the same hardware later supports testing or limited inference. Training clusters typically prioritize large-scale parallel compute, high-bandwidth networking and storage, and sustained access to large datasets; production-serving systems belong in AI Cloud inference when real-time model execution defines their primary role.
- **Hybrid or split categories:** Medical wearables from consumer wearables whenever reimbursement, clinical validation, privacy requirements, or regulated design cycles influence processor segment selection.

Cross-cutting classifications

Some dimensions cut across every IoT subsegment rather than create more market buckets:

- **Sensor modality:** vision, audio, radar, lidar, physiological, location, vibration, or environmental.
- **Compute class:** MCU/TinyML, low-power NPU SoC, vision SoC, application processor, or edge gateway/server.
- **Inference location:** on-sensor, endpoint, local gateway, or regional edge.
- **Power source:** battery, wired, energy-harvesting, vehicle, or industrial power.
- **Connectivity:** no network, Bluetooth, Wi-Fi, cellular, LPWAN, industrial Ethernet, or private 5G.
- **Safety/regulatory profile:** consumer, industrial, medical, automotive-adjacent, or public infrastructure.

This approach makes it easier to compare vendors that compete across different endpoints. Ambiq, Nordic Semiconductor, Synaptics, Qualcomm, MediaTek, NXP, STMicroelectronics, Renesas, Texas Instruments, Himax, Sony Semiconductor, Hailo, Axera, Rockchip, and others may overlap in “AI IoT,” but they compete in very different power, memory, model-size, and ASP bands.

AI wearables

AI wearables are considered IoT devices because they connect to the internet, collect sensor data, and exchange information with other systems.



Figure 2. All the IoT things one can wear

Why AI wearables are IoT devices

- **Connectivity:** They connect via Bluetooth, Wi-Fi, or cellular networks to sync data.
- **Sensors:** They feature built-in hardware like heart-rate monitors, GPS, and microphones to track real-world data.
- **Network Interaction:** They send and receive information from cloud servers or companion smartphones.

How AI changes the IoT definition

- **Edge Processing:** Modern AI wearables often run small intelligence models locally on the device (like Google's lightweight models) rather than relying entirely on remote servers.

- Smart Action: Instead of just reporting raw data (like step counts or temperature), AI wearables process context to trigger actions, make recommendations, or hold conversations directly from your wrist.

Beyond wearables, IoT segment can cover any endpoint or near-endpoint device that runs AI inference locally or through a gateway. IoT captures fixed, portable, and low-power connected endpoints and their local gateways, and not autonomous systems.

- Wearables and personal devices
 - Hearables, smartwatches, smart rings, health patches, smart clothing, smart footwear, AR glasses, and health-monitoring devices
- Consumer and smart home
 - Speakers, displays, cameras, doorbells, home hubs, appliances, thermostats, and consumer cleaning or lawn devices
- Vision and smart-city infrastructure
 - Surveillance, traffic, parking, access control, environmental monitoring, and municipal cameras/sensors
- Industrial, energy, and field IoT
 - Factory automation, machine vision, predictive maintenance, grid and utility endpoints, mining, oil and gas, agriculture, and environmental systems
- Healthcare and medical edge
 - Regulated medical wearables, home-care hubs, portable diagnostics, imaging endpoints, and clinical monitoring devices
- Retail, logistics, and commercial edge
 - POS, kiosks, smart shelves, inventory cameras, vending, asset trackers, handheld terminals, and cold-chain monitoring.

Device inference

Edge AI commonly appears in sensors, cameras, industrial equipment, wearables, smart appliances, healthcare systems, retail deployments, agriculture, energy management, and smart-city infrastructure. Its core value comes from local processing where latency, privacy, bandwidth, connectivity, or power constraints make a cloud round trip less desirable.

AI-enabled smartphones and PCs are considered Device inference because they run trained AI models locally on general-purpose compute hardware, answering user requests without sending data to a cloud server.



Figure 3. Device inference hardware—AI PC, smartphone, or XR headset running local models

Why AI smartphones and PCs are Device inference

Local Compute: They carry dedicated NPU, GPU, or CPU silicon capable of running trained models directly on the device.

On-Device Models: They increasingly run trimmed, quantized versions of larger models, the same pattern behind Gemini Nano and Apple Intelligence running locally on a phone.

User-Facing Interaction: They deliver results instantly at the point of use, for tasks like voice assistants, photo and video processing, and writing tools.

How AI changes the Device inference definition

Edge Processing: Modern smartphones and PCs run substantial portions of what used to be pure cloud-hosted AI workloads locally instead.

Smart Action: Instead of just running the operating system and installed apps, devices interpret context and intent to summarize, generate, and assist directly, without a round trip to an API.

Beyond smartphones and PCs, Device inference segment can cover any general-purpose computing device with enough onboard compute to run trained AI models locally for user-facing applications. Device inference captures consumer and professional computing devices built for general productivity, computing, or entertainment, not purpose-built connected endpoints and not systems designed for autonomous physical control.

- Smartphones and tablets
 - Flagship and midrange smartphones, tablets, and foldables running on-device NPUs for camera, voice, and generative features.

- PCs and laptops
 - AI PCs, Copilot+ PCs, MacBooks with on-device neural engines, and workstations running local LLM inference for productivity and creative tools.
- Gaming and entertainment devices
 - Consoles, handheld gaming devices, and streaming devices running local AI for upscaling, frame generation, and content recommendation.
- Extended reality and computing wearables
 - AR, VR, and MR headsets and smart glasses running real-time local inference for spatial tracking, hand and eye tracking, and scene understanding.
- Embedded and single-board AI compute
 - Development boards and embedded compute modules used to prototype or deploy local inference in products still built around general-purpose compute.
- Professional and creative workstations
 - Local AI acceleration for content creation, CAD, video editing, and other professional software increasingly embeds on-device model inference.

Device inference commonly appears in smartphones, PCs, laptops, tablets, gaming consoles, and XR headsets. Its core value comes from immediate, private, and offline-capable AI features at the point of use, where waiting for a cloud response would degrade the experience even when a connection is available.

Autonomous Systems

AV cars, drones, AMRs, humanoids, autonomous tractors, and autonomous warehouse vehicles are in Autonomous Systems when perception, navigation, or safety control drives the design.

AV cars, drones, AMRs, humanoids, autonomous tractors, and autonomous warehouse vehicles are considered Autonomous Systems because perception, navigation, or safety control drives the fundamental design, not just data collection or user-facing computation.



Figure 4. Autonomous Systems hardware—AV, drone, AMR, or humanoid robot in operation

Why AV cars and robots are Autonomous Systems

- Perception: They fuse camera, lidar, radar, and other sensor data in real time to build a continuous model of their surroundings.
- Navigation: They plan and execute paths through physical space, adjusting for obstacles, terrain, and other moving agents.
- Safety Control: They carry dedicated safety-critical compute and redundant systems designed to prevent collisions and handle failure conditions.

How AI changes the Autonomous Systems definition

- Real-Time Decision-Making: Modern autonomous systems increasingly run full perception-to-action pipelines locally, replacing rule-based control with learned behavior models.
- Physical Action: Instead of just reporting sensor data or waiting for instructions, autonomous systems act directly in the physical world, steering, gripping, walking, or flying without a human in the loop for each decision.

Beyond AV cars, the Autonomous Systems segment can cover any machine that perceives its environment and acts on it with reduced or no direct human control. Autonomous Systems captures vehicles and robots built primarily to navigate and operate in physical space, not fixed or handheld devices that report data or run user-facing applications.

- Automotive and road vehicles
 - Passenger AVs, robotaxis, autonomous trucks, and driver-assist systems that take over core driving functions
- Aerial systems

- Consumer, commercial, and industrial drones performing autonomous flight, inspection, delivery, and mapping
- Ground robotics and AMRs
 - Autonomous mobile robots, last-mile delivery robots, and autonomous warehouse vehicles navigating shared human spaces
- Humanoid and legged robots
 - Bipedal and quadrupedal robots designed for locomotion and manipulation in human-built environments
- Agricultural and off-road vehicles
 - Autonomous tractors, harvesters, mining equipment, and construction machinery operating with reduced supervision
- Maritime and specialized autonomous systems
 - Autonomous vessels, underwater vehicles, and other specialized platforms operating without continuous human control

Autonomous Systems commonly appear in AV cars, drones, AMRs, humanoids, autonomous tractors, and warehouse vehicles. Its core value comes from removing continuous human control from physical operation, where perception, navigation, and safety-critical decisions must happen fast enough and reliably enough to act in the real world without waiting on a person.

AI Training

Data center GPU clusters, TPU pods, and dedicated training accelerators are considered AI Training because they run the iterative process that builds or updates a model's weights, not because they serve an already-trained model to end users.



Figure 5. AI Training hardware—GPU cluster, TPU pod, or wafer-scale training system

Why GPU clusters and TPU pods are AI Training

- **Massive Parallelism:** They link thousands of accelerators together to split a single training run across the whole cluster.
- **High-Precision Math:** They support the higher-precision number formats, like FP32, TF32, and BF16, that gradient calculations need to converge correctly.
- **Sustained Workloads:** They run continuously for days, weeks, or months at a time, rather than the millisecond bursts typical of inference.

How AI changes the AI Training definition

- **Distributed Training:** Modern training runs split a single model across thousands of chips at once, moving beyond a single powerful machine to a coordinated cluster.
- **Efficient Scaling:** Instead of just adding more chips, training systems now depend on high-bandwidth interconnects and smarter parallelism strategies to keep utilization high as clusters grow.

Beyond hyperscale GPU clusters, the AI Training segment can cover any system built primarily to run the compute-intensive process of building or updating model weights. AI Training captures the infrastructure that creates a model, not the infrastructure that serves it afterward, which falls under AI Cloud inference instead.

- **Hyperscale training clusters**
 - Nvidia H100, B200, and GB200 clusters, Google TPU pods, and AWS Trainium clusters running foundation-model pretraining.
- **Enterprise and private training infrastructure**
 - On-premises GPU clusters used by companies to fine-tune models on proprietary data without sending it to a public cloud
- **Research and academic supercomputers**
 - National lab and university systems used to train large models for scientific and research purposes
- **Specialized training accelerators**
 - Purpose-built training silicon, such as wafer-scale engines, designed specifically for the backpropagation workload rather than general compute
- **Fine-tuning and post-training infrastructure**
 - Smaller-scale clusters used for instruction tuning, reinforcement learning from human feedback, and other post-training steps after initial pretraining
- **High-speed interconnect and networking**
 - NVLink, NVSwitch, InfiniBand, and other fabric technologies that synchronize gradients across thousands of chips during a training run

AI Cloud inference

Regional and rack-scale servers running trained models are considered AI Cloud inference because they serve predictions and generated outputs to end users and applications, not because they update a model's weights.



Figure 6. AI Cloud inference hardware—rack-scale inference server or hyperscaler GPU bank

Why regional servers and hyperscaler GPU banks are AI Cloud inference

- Centralized Serving: They host trained models in one location and answer requests from many users and applications over the network.
- Elastic Capacity: They scale up or down by adding or reallocating GPU capacity to match real-time demand, rather than running a single fixed workload.
- Multi-Tenant Infrastructure: They serve many different applications and customers from the same shared pool of hardware, rather than dedicating hardware to one user.

How AI changes the AI Cloud Inference definition

- Dynamic Workload Allocation: Hyperscalers increasingly swap the same GPU banks between training and inference duty, running inference on infrastructure primarily built and provisioned for training whenever demand shifts.
- Distributed Serving: Instead of routing every request to one central data center, cloud inference increasingly spans regional and edge-adjacent servers to cut latency for users in different locations.

Beyond hyperscaler data centers, AI Cloud inference segment can cover any server-class system that serves trained models to remote users or applications over a network. AI Cloud inference captures centralized and regional serving infrastructure, not the infrastructure that builds a model, which falls under AI Training, and not devices that run inference locally for a single user, which fall under Device inference or Autonomous Systems. Edge servers count as a separate overlay within this segment when they serve multiple sites or locations, rather than running inference for a single local device or user.

- Hyperscaler inference infrastructure
 - Nvidia, AMD, and custom-silicon GPU or accelerator banks inside major cloud providers, reallocated between training and inference as demand shifts
- Regional and rack-scale inference servers
 - Dedicated inference server racks deployed in regional data centers to serve users closer to their location
- Model-serving platforms and APIs
 - Managed inference services and API endpoints that route requests to shared GPU infrastructure on behalf of many applications
- Multi-tenant co-location and neocloud infrastructure
 - GPU capacity rented from co-location providers or specialized neocloud operators, serving inference workloads for multiple customers
- Edge-server overlay
 - Server-class inference infrastructure deployed at telecom, retail, or enterprise edge locations that serve multiple sites at once, distinct from a single on-device or single-site deployment

AI Cloud inference commonly appears in hyperscaler data centers, regional server racks, and multi-site edge-server deployments. Its core value comes from serving a trained model efficiently to many users and applications at once, where the goal is high request throughput and low response latency across a shared pool of hardware rather than finishing a single training run.

The AIP zoo

300 creatures vying for your attention.

With 10 processor families and 153 companies chasing design wins, no single architecture will win outright. GPUs anchor large-scale training, NPUs and CIM chips push inference to the edge, dataflow and analog designs chase efficiency, and RISC-V gives silicon teams a customizable foundation. For ISVs and CIOs, the real work ahead is matching each workload to the right architecture, not betting on a single winner.



When we began our AI processor information system tracker service in mid-2025, we thought it would be fairly easy—just find who was making an NPU, HW, or IP. To our surprise, we found 53 companies—53! But even more surprising, and a little disturbing, was that we found several alternative AIPs besides NPUs.

Today we have cataloged 10 different types of AI processors plus the IP equivalents, for a total of 180 entrants when the acquired and failed companies are included. Currently, 153 companies are actively selling 295 AIP devices.

Ten different processors are used for AI processing (training and/or inference): GPU, NPU, dataflow, CIM, neuromorphic, RISC-V, x86, Arm CPU, hybrid, and analog-based.

The following is a brief description of each one.

GPU (graphics processing unit)

- **Description:** Originally built for parallel graphics rendering, GPUs pack thousands of small cores that execute matrix and vector operations concurrently. This makes them the dominant engine for AI.



Figure 7. GPU generalized block diagram

- **Role:** Massively parallel processor vital for deep learning.
- **Function:** Excels at matrix multiplications and floating-point operations.
- **Usage:** Powers large-scale deep learning model training and inference.
- **Pros:** Massive parallel throughput; mature software stack (CUDA); handles both training and inference at scale.
- **Cons:** High power draw; costly; overkill for simple edge inference.

NPU (neural processing unit)

- **Description:** A purpose-built accelerator optimized specifically for the multiply-accumulate operations that dominate neural network inference. They trade general-purpose flexibility for superior performance per watt.



Figure 8. NPU generalized block diagram

- **Role:** Dedicated hardware built for neural network calculations.
- **Function:** Optimizes tensor operations with high energy efficiency.
- **Usage:** Accelerates real-time AI inference locally in smartphones, laptops, and edge devices.

- **Pros:** Excellent performance per watt; runs models locally without cloud dependency; small footprint.
- **Cons:** Limited flexibility; mostly inference-only; often locked to vendor toolchains.

Dataflow processor

- **Description:** Organizes computation around the movement of data through a network of processing elements rather than a fixed instruction sequence. Operations fire as soon as inputs arrive, minimizing idle cycles.

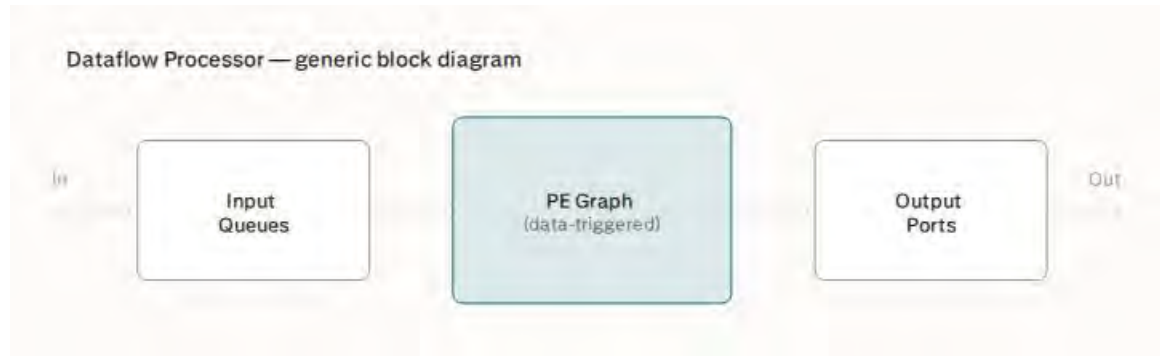


Figure 9. Dataflow generalized block diagram

- **Role:** Architecture that routes data directly between functional units.
- **Function:** Eliminates traditional memory bottlenecks by bypassing cache hierarchies.
- **Usage:** Maximizes throughput for complex, data-heavy neural network graphs and streaming workloads.
- **Pros:** Minimizes memory traffic; efficient for tensor-heavy pipelines; scales well.
- **Cons:** Less mature compiler/software ecosystem; harder to program than GPUs.

CIM (compute-in-memory)

- **Description:** Embeds arithmetic directly inside or beside memory arrays, letting weights stay in place while computation happens. This collapses the traditional divide between memory and processor.

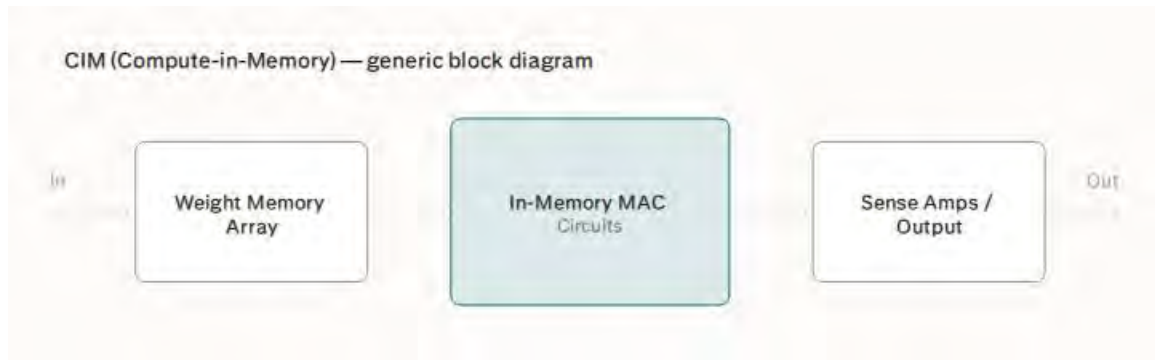


Figure 10. CIM generalized block diagram

- **Role:** Processor that integrates data storage and computation on a single unit.
- **Function:** Reduces energy loss and latency caused by moving data back and forth.
- **Usage:** Boosts efficiency for memory-bound edge AI and deep learning workloads.
- **Pros:** Collapses memory-compute divide; very low energy per inference; ideal for edge devices.
- **Cons:** Precision and analog-noise limitations; still mostly in research or early commercial stages.

Neuromorphic processor

- **Description:** Mimics the brain's spiking neurons and synapses in silicon, encoding and processing information as sparse, asynchronous electrical spikes rather than dense numerical tensors.

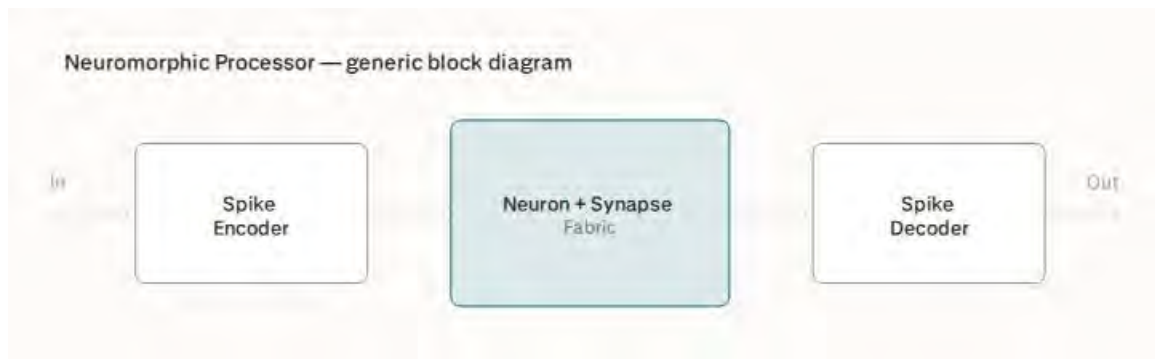


Figure 11. Neuromorphic generalized block diagram

- **Role:** Chip architecture inspired by biological human brains and neural biology.
- **Function:** Processes information via spiking neural networks and event-driven logic.
- **Usage:** Handles ultra-low-power pattern recognition and cognitive computing tasks.
- **Pros:** Extremely energy-efficient for sparse, event-driven tasks; brain-inspired efficiency.

- **Cons:** Hard to program; narrow use cases; lacks broad software and tooling support.

RISC-V processor

- **Description:** An open, royalty-free instruction-set architecture that chipmakers extend with custom vector and matrix instructions. Its openness lets vendors build tailored AI cores without licensing fees.



Figure 12. RISC-V generalized block diagram

- **Role:** Open-standard instruction set architecture utilized for custom AI chips.
- **Function:** Allows hardware designers to add specialized matrix extension instructions.
- **Usage:** Powers custom, scalable AI accelerators and embedded smart edge-inference designs.
- **Pros:** Open, royalty-free, customizable with AI-specific instructions; fast-growing ecosystem.
- **Cons:** Fragmented tooling across vendors; younger AI-software ecosystem than x86 or Arm.

x86 CPU

- **Description:** The general-purpose architecture from Intel and AMD, historically optimized for sequential workloads. Modern iterations add wide vector extensions like AVX-512 and AMX for matrix math.



Figure 13. x86 generalized block diagram

- **Role:** General-purpose processor standard in personal computers and servers.
- **Function:** Executes sequential logic, orchestrates workflows, and supports heavy software ecosystems.
- **Usage:** Manages data pre-processing, orchestration, and light AI inference in CPU-only pipelines.
- **Pros:** Extremely mature ecosystem; handles general compute plus AI; no extra accelerator needed for lighter workloads.
- **Cons:** Lower power efficiency; falls significantly behind GPUs and NPUs for heavy training or high-throughput inference.

Arm CPU

- **Description:** A power-efficient, general-purpose architecture dominant in mobile computing and expanding into servers. Arm cores handle AI via vector/matrix extensions (SVE2, SME) or by orchestrating attached accelerators.

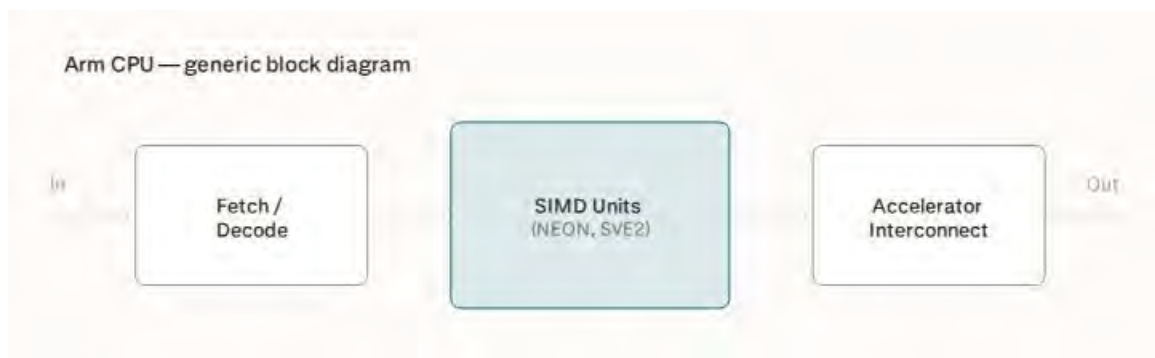


Figure 14. Arm CPU generalized block diagram

- **Role:** Energy-efficient reduced instruction set computer (RISC) processor architecture.
- **Function:** Delivers high performance per watt for constrained environments.
- **Usage:** Runs mobile AI algorithms, smart home devices, and server-edge inference.

- **Pros:** Strong power efficiency; dominant in mobile/edge; flexible orchestration of attached accelerators.
- **Cons:** Requires a paired NPU or GPU accelerator to achieve competitive speeds on massive AI models.

Hybrid processor

- **Description:** Combines two or more compute paradigms such as CPU cores paired with GPU, NPU, or dataflow tiles on a single die or package. They route each part of an AI workload to the most efficient engine.

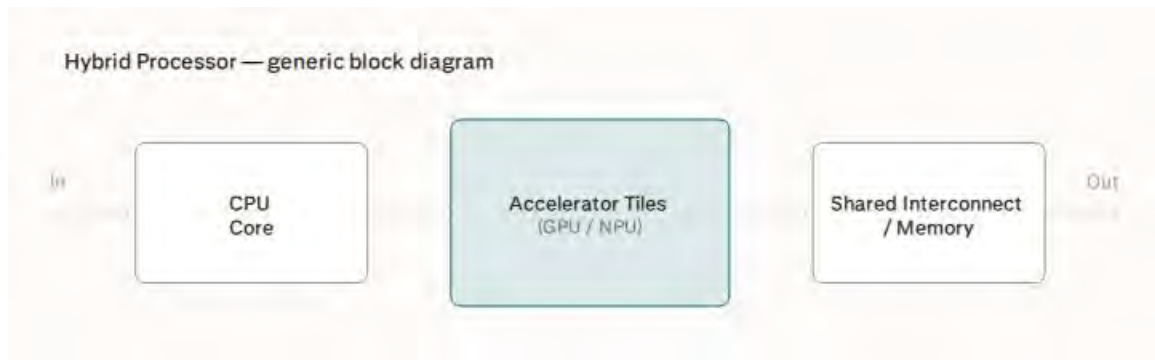


Figure 15. Hybrid generalized block diagram

- **Role:** Multi-architecture system combining different compute cores on one die.
- **Function:** Pairs general-purpose CPU cores with specialized accelerators.
- **Usage:** Balances heavy application processing with dedicated low-power AI tasks.
- **Pros:** Optimally balances flexibility, throughput, and power by routing sub-tasks dynamically.
- **Cons:** Higher design complexity; requires intelligent software compilers to split workloads efficiently.

Analog-based processor

- **Description:** Performs computation using continuous electrical signals like current or voltage rather than discrete digital logic, exploiting Ohm's and Kirchhoff's laws to execute matrix multiplication directly.

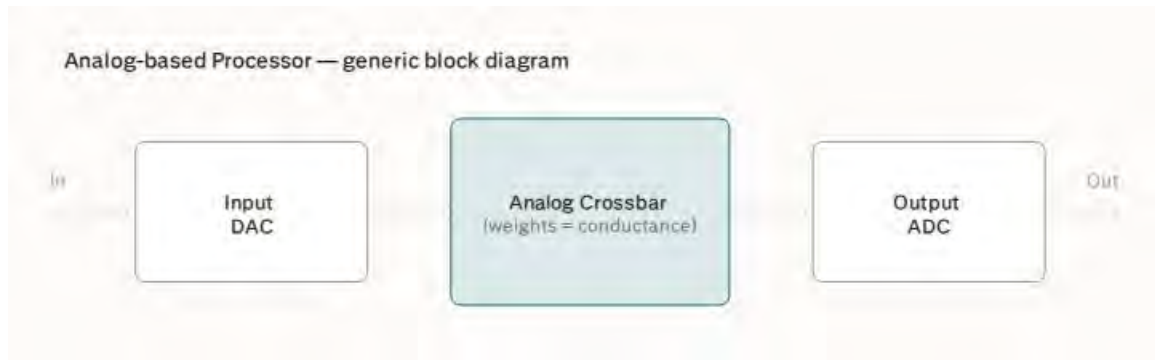


Figure 16. Analog generalized block diagram

- **Role:** Chip that computes data using continuous physical analog electrical signals.
- **Function:** Represents weights and activations via voltages or currents directly in the analog domain.
- **Usage:** Speeds up matrix multiplication tasks with minimal power consumption.
- **Pros:** Offers dramatic energy savings for inference; speeds up matrix math with minimal power.
- **Cons:** Faces precision, noise, thermal drift, and manufacturing-variability challenges.

This diversity is healthy, not chaotic. No single architecture serves every workload, power budget, or price point, so specialization is the rational engineering outcome. Expect consolidation among weaker entrants over the next two years, while GPUs, NPUs, and hybrid designs keep the largest revenue share. Analog and neuromorphic chips remain promising, but still early and unproven.

Inflection point: Does 180 competing AI processor designs signal an inflection point? We think so. Compute demand is outpacing what general-purpose chips can efficiently deliver, forcing architectural experimentation not seen since the early GPU era. This proliferation of GPUs, NPUs, dataflow, CIM, neuromorphic, RISC-V, and analog designs marks a genuine inflection point for the industry: AI workloads are now diverse enough to justify dozens of specialized silicon bets, reshaping how ISVs design software and how CIOs plan infrastructure for the next decade.

Appendix

Expanded IoT taxonomy

IoT edge category	Typical AI function	Example endpoints	Tracker treatment
Smart cameras and vision sensors	Detection, tracking, OCR, anomaly detection	Security cameras, machine-vision cameras, doorbells, body cameras, dashcams	Large, distinct category; vision often drives higher ASP silicon
Industrial IoT and machine control	Predictive maintenance, inspection, process control	PLC-adjacent controllers, vibration sensors, machine cameras, CNC equipment	Separate category; ruggedization, deterministic latency, and long lifetimes matter
Smart home and consumer appliances	Voice, vision, sensing, personalization	Smart speakers, displays, thermostats, appliances, robot vacuums, doorbells	High-unit, low-to-mid ASP category
Healthcare and medical devices	Vital-sign analysis, detection, imaging, decision support	CGMs, insulin pumps, remote-monitoring hubs, portable ultrasound, diagnostic tools	Separate from consumer wearables when regulation and medical validation affect silicon choice
Retail and hospitality endpoints	Inventory, checkout, loss prevention, customer analytics	POS terminals, smart shelves, kiosks, vending machines, digital signage	Often vision-driven; include the endpoint processor and local gateway where relevant
Smart-city infrastructure	Traffic, public safety, environmental sensing	Traffic cameras, intersections, parking systems, air-quality monitors, smart meters	Combine device silicon and edge gateways only if you avoid double counting
Energy, utilities, and grid edge	Load forecasting, fault detection, asset monitoring	Smart meters, inverters, substation monitors, EV chargers, battery systems	Separate category; long deployment cycles and reliability requirements shape demand
Agriculture and environmental monitoring	Crop analysis, livestock monitoring, precision control	Field cameras, irrigation controllers, soil sensors, weather stations	Lower volume, varied silicon; include gateways and vision systems
Logistics and asset intelligence	Location, condition monitoring, routing, recognition	Asset trackers, cold-chain sensors, handheld scanners, fleet telematics, parcel scanners	Do not place autonomous warehouse robots here; keep those in Autonomous Systems
Communications and network edge	Traffic optimization, security, radio optimization	AI gateways, routers, small cells, customer-premises equipment	Define clearly against AI cloud inference; track device-side/network-edge processors only
Robotics appliances and fixed automation	Local perception and task execution	Cleaning robots, lawn robots, warehouse sorters, food-service machines	Split fixed/mobile autonomous robots into Autonomous Systems if navigation dominates
Audio and speech endpoints	Wake-word, noise suppression, translation, health audio	Smart speakers, hearing aids, conferencing bars, intercoms, earbuds	You can either retain this within wearables/smart home or report it as a cross-cutting audio category

Table 1. IoT subsegments