EDITOR: Joshua J. Yi, josh@joshuayipatentlaw.com

DEPARTMENT: MICRO LAW

Review of Patents Issued to Computer Architecture Companies in 2021—Part II

Joshua J. Yi, *The Law Office of Joshua J. Yi, PLLC, Austin, TX, 78750, USA*

In my last article, which appeared in the March/April 2022 issue of *IEEE Micro*, I analyzed 1) the number of all patents and the number of computer architecture patents issued in 2021 and 2) the distribution of those patents across patent classes, for 18 computer architecture companies. I then presented a notable patent for five of those companies. This article presents a notable patent for the remaining companies.

As I described in more detail in the previous article, each of the following patents is from the G06F ("Electric digital data processing") patent class as that patent class had the largest number of patents for all but three companies and because it was the only patent class where all companies had a least one patent. Furthermore, the below patents are from large patent families, which are generally considered to be more valuable.

Assignee: Dell+EMC
Number: 10,929,171
Filed: June 18, 2019
Issued: February 23, 2021
Title: Distributed forwarding for performing service chain operations
Inventors: S. Gokhale (Santa Clara, CA), C. Lecuyer (Mountain View, CA), R. Nair (Newark, CA), K. Mundaragi (Pune, IN), R. Mishra (Mountain View, CA), P. Rolando (Santa Clara, CA), J. Jain (Cupertino, CA), R. Koganty (San Jose, CA)

Abstract: Some embodiments provide novel methods for performing services for machines operating in one or more datacenters. For instance, for a group of related guest machines (e.g., a group of tenant machines), some embodiments define two different forwarding planes: 1) a guest forwarding plane and 2) a service forwarding plane. The guest forwarding plane connects to the machines in the group and performs L2 and/or L3 forwarding for these machines. The service forwarding plane 1) connects to the service nodes that perform services on data messages sent to and from these machines; and 2) forwards these data messages to the service nodes. In some embodiments, the guest machines do not connect directly with the service forwarding plane. For instance, in some embodiments, each forwarding plane connects to a machine or service node through a port that receives data messages from, or supplies data messages to, the machine or service node. In such embodiments, the service forwarding plane does not have a port that directly receives data messages from, or supplies data messages to, any guest machine. Instead, in some such embodiments, data associated with a guest machine is routed to a port proxy module executing on the same host computer, and this other module has a service plane port. This port proxy module in some embodiments indirectly can connect more than one guest machine on the same host to the service plane (i.e., can serve as the port proxy module for more than one guest machine on the same host).

Figure 1 "illustrates a data message between two guest virtual machines (GVMs) being redirected along a service path to be processed by service virtual machines (SVMs)" ('171 Patent).

Assignee: IBM
Number: 10,884,742
Filed: August 27, 2019
Issued: January 5, 2021
Title: Handling unaligned load operations in a multi-slice computer processor
Inventors: S. Chadha (Austin, TX), R. Cordes (Austin, TX), D. Hrusecky (Cedar Park, TX), H. Le (Austin, TX), J. Leenstra (Bondorf, DE), D. Nguyen (Austin, TX), B. Thompto (Austin, TX), A. Van Norstrand, Jr. (Round Rock, TX)

Abstract: Handling unaligned load operations, including: receiving a request to load data stored within a range of addresses; determining that the range of addresses includes addresses associated with a plurality of caches, wherein each of the plurality of caches is associated with a distinct processor slice;

0272-1732 © 2022 IEEE
Digital Object Identifier 10.1109/MM.2022.3163434
Date of current version 18 May 2022.

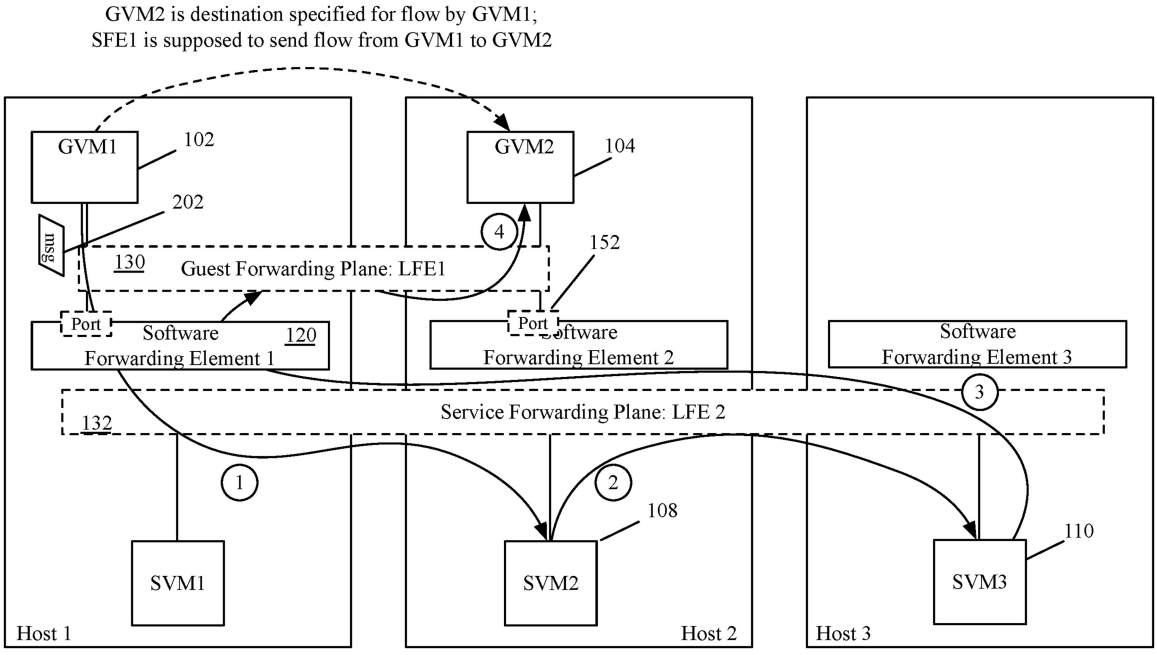

FIGURE 1. Representative figure from U.S. Patent no. 10,929,171 issued to Dell.

issuing, to each distinct processor slice, a request to load data stored within a cache associated with the distinct processor slice, wherein the request to load data stored within the cache associated with the distinct processor slice includes a portion of the range of addresses; executing, by each distinct processor slice, the request to load data stored within the cache associated with the distinct processor slice; and receiving, over a plurality of data communications busses, execution results from each distinct processor slice, wherein each data communications busses is associated with one of the distinct processor slices.

Figure 2 depicts "a flow chart illustrating an example method for handling unaligned load operations in a multi-slice computer processor" ('742 Patent).

Assignee: Intel
Number: 11,016,895
Filed: April 6, 2018
Issued: May 25, 2021
Title: Caching for heterogeneous processors
Inventors: F. Hady (Portland, OR), M. Cabot (San Francisco, CA), M. Rosenbluth (Uxbridge, MA), J. Beck (Northboro, MA)

Abstract: A multicore processor providing heterogeneous processor cores and a shared cache is presented.

Additional description: In contrast with conventional systems, in which the special-purpose processors are on separate silicon and are placed on the I/O connectors of a general-purpose processor (e.g., host) system, the core(s) of the multiprocessor 12 are integrated onto the same die as the central processing unit (CPU) core 24, and potentially, cache 18. Such integration allows the heterogeneous cores the opportunity to more efficiently share data as they are placed behind a common cache. Thus, in one embodiment, as illustrated in the figures, the processor cores, cache and interconnect reside on a single chip. Alternatively, the processor cores 20, cache 18, and interconnect 26 may be implemented as separate chips in a multichip package. In yet another alternative embodiment, the processor cores 20, cache 18, and interconnect 26 may be implemented as a combination of chip and board design.

Figure 3 shows an "exemplary heterogeneous multicore processor having a bus-based shared cache architecture" ('895 Patent).

Assignee: Marvell+Cavium
Number: 10,970,080
Filed: November 9, 2018
Issued: April 6, 2021
Title: Systems and methods for programmable hardware architecture for machine learning.
Inventors: A. Sodani (San Jose, CA), C. Chen (Santa Clara, CA), U. Hanebutte (Gig Harbor, WA), H. Ghasemi (Sunnyvale, CA), S. Durakovic (Palo Alto, CA)

Abstract: A programmable hardware architecture for machine learning (ML) is proposed, which includes at least a host, a memory, a core, a data streaming engine, a[n] instruction-streaming engine, and an

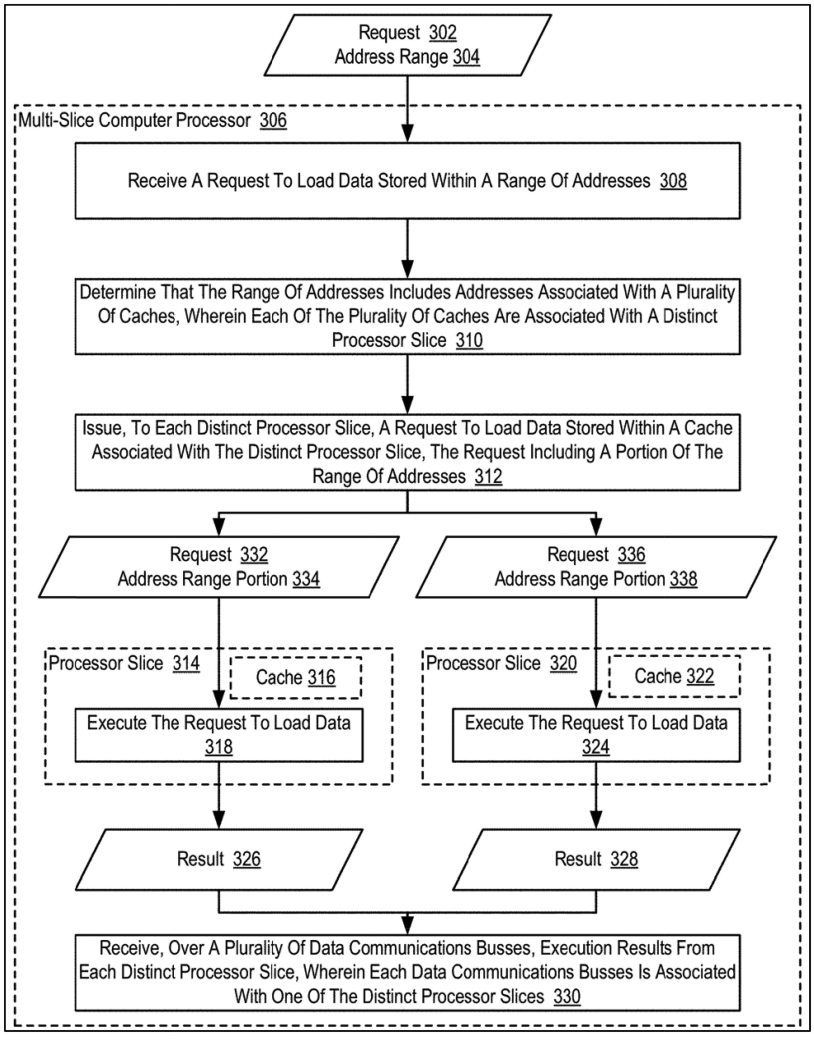

FIGURE 2. Representative figure from U.S. Patent no. 10,884,742 issued to IBM.

interference engine. The core interprets a plurality of ML commands for an ML operation and/or data received from the host and coordinate activities of the engines based on the data in the received ML commands. The instruction-streaming engine translates the ML commands received from the core and provides a set of programming instructions to the data streaming engine and the inference engines based on the translated parameters. The data steaming engine sends one or more data streams to the inference engine in response to the received programming instructions. The inference engine then processes the data streams received from the data stream engine according to the programming instructions received from the instruction-streaming engine.

Figure 4 depicts a "diagram of an example of the architecture of the first type of processing unit" ('080 Patent).

Assignee: Microsoft
Number: 11,100,423
Filed: January 26, 2017
Issued: August 24, 2021
Title: Artificial intelligence engine hosted on an online platform
Inventors: M. Hammond (Berkeley, CA), K. Browne (Berkeley, CA), M. Campos (Carlsbad, CA), M. Brown (San Francisco, CA), R. Kong (Berkeley, CA), M. Adams (San Francisco, CA)
Abstract: Provided herein in some embodiments is an artificial intelligence ("A") engine hosted on one or

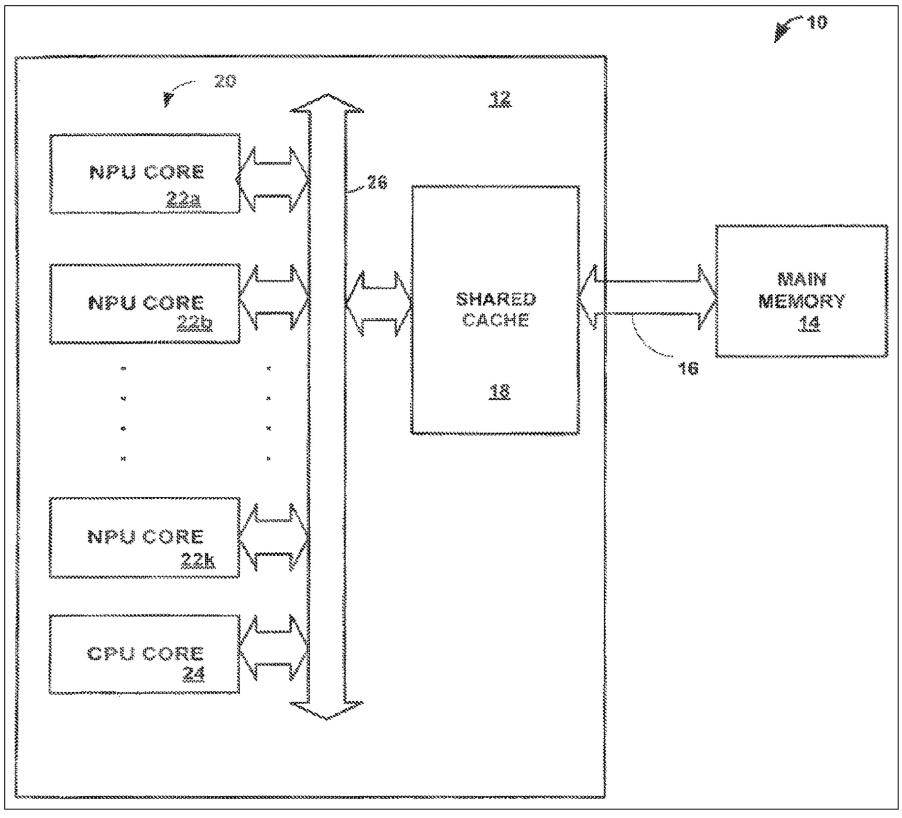

FIGURE 3. Representative figure from U.S. Patent no. 11,016,895 issued to Intel.

more remote servers configured to cooperate with one or more databases including one or more AI-engine modules and one or more server-side client-server interfaces. The one or more AI-engine modules include an instructor module and a learner module configured to train an AI model. An assembly code can be generated from a source code written in a pedagogical programming language describing a mental model of one or more concept modules to be learned by the AI model and curricula of one or more lessons for training the AI model. The one or more server-side client-server interfaces can be configured to enable client interactions from a local client such as submitting the source code for training the AI model and using the trained AI model for one or more predictions.

Figure 5 is a "schematic illustrating an AI system" in accordance with some aspects of the invention ('423 Patent).

Assignee: NVIDIA
Number: 10,997,496
Filed: March 14, 2017
Issued: May 4, 2021
Title: Sparse convolutional neural network accelerator

Inventors: W. Dally (Los Altos Hills, CA), A. Parashar (Northborough, MA), J. Emer (Acton, MA), S. Keckler (Austin, TX), L. Dennison (Mendon, MA)

Abstract: A method, computer program product, and system perform computations using a sparse convolutional neural network accelerator. Compressed-sparse data is received for input to a processing element, wherein the compressed-sparse data encodes nonzero elements and corresponding multidimensional positions. The nonzero elements are processed in parallel by the processing element to produce a plurality of result values. The corresponding multidimensional positions are processed in parallel by the processing element to produce destination addresses for each result value in the plurality of result values. Each result value is transmitted to a destination accumulator associated with the destination address for the result value.

Figure 6 illustrates "a block diagram of a SCNN accelerator" ('496 Patent).

Assignee: NXP+Freescale
Number: 10,979,033
Filed: July 27, 2020
Issued: April 13, 2021

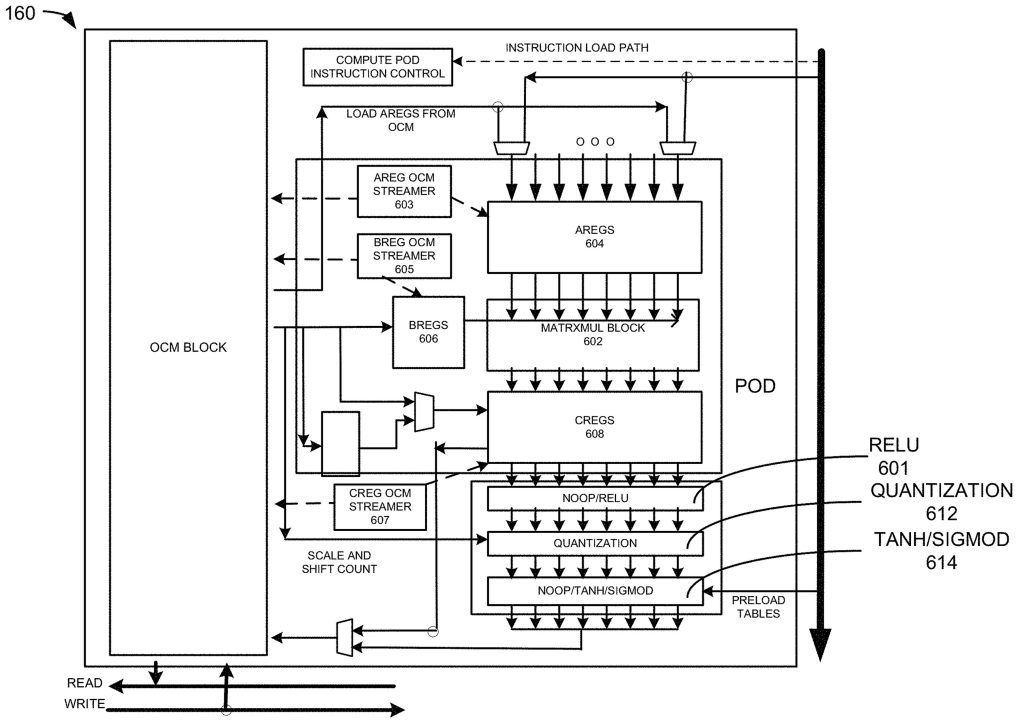

FIGURE 4. Representative figure from U.S. Patent no. 10,970,080 issued to Marvell.

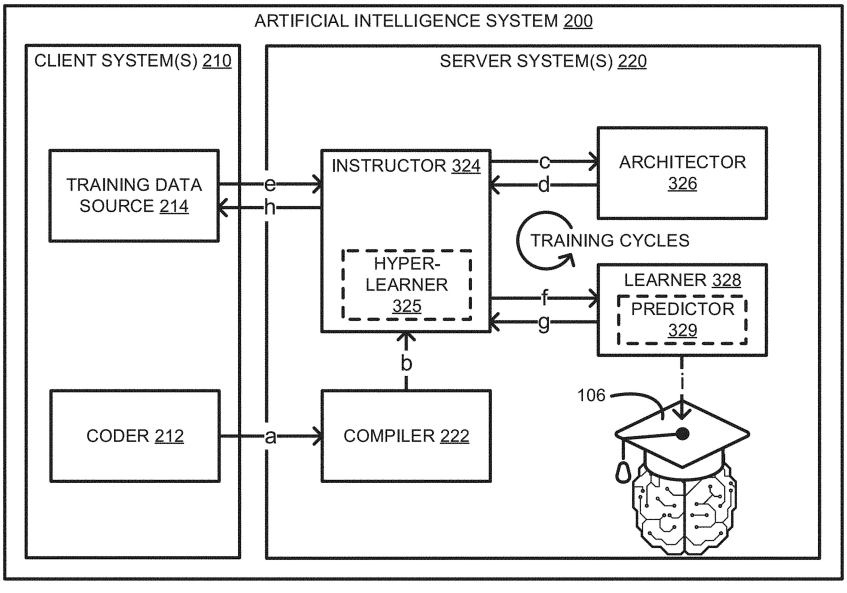

FIGURE 5. Representative figure from U.S. Patent no. 11,100,423 issued to Microsoft.

Title: Current-controlled oscillator

Inventors: A. Vilas Boas (Amparo, BR), F. Clayton (Campinas, BR)

Abstract: A current-controlled oscillator receives an input current. Ramp voltage generating circuitry generates first and second ramp voltages in response to the input current. Selecting circuitry selects one of the first and second ramp voltages depending on their relative values. Switching circuitry receives a selected ramp voltage, generates a signal based on the selected

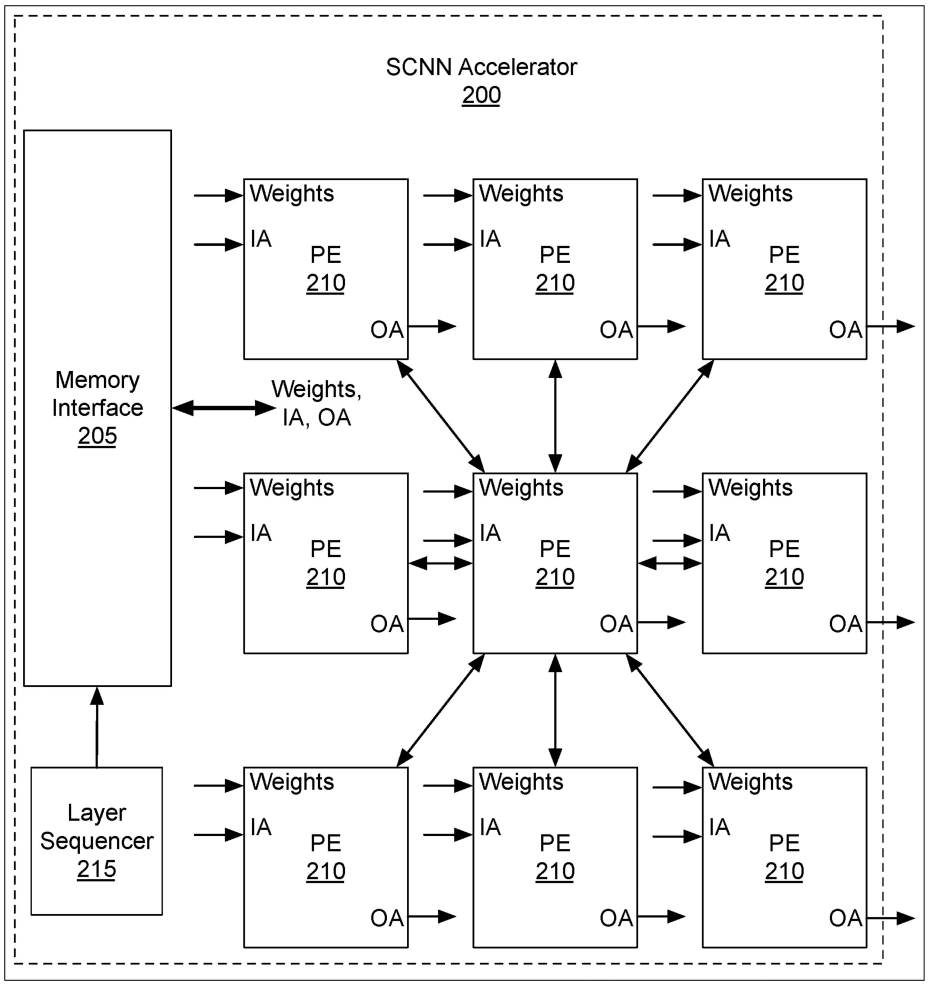

FIGURE 6. Representative figure from U.S. Patent no. 10,997,496 issued to NVIDIA.

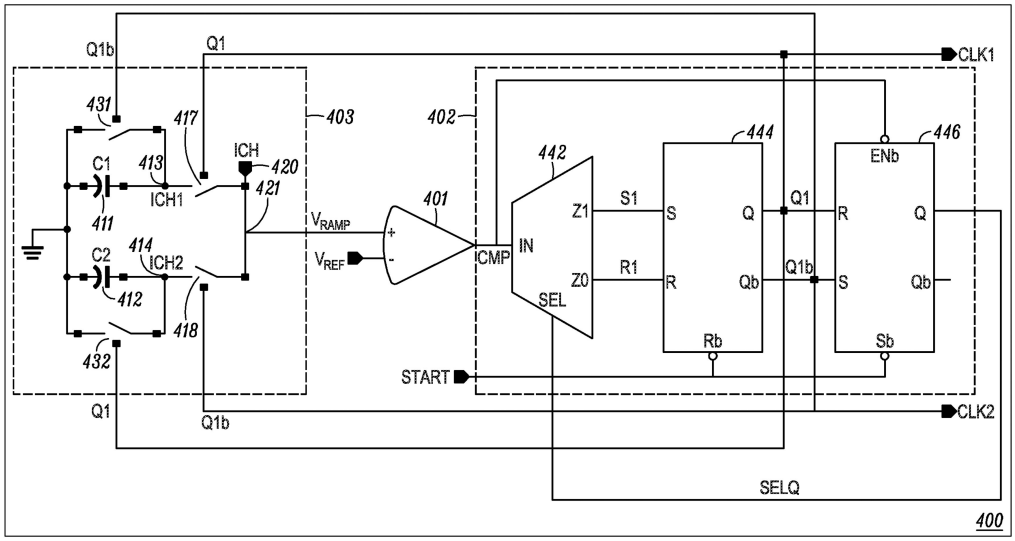

FIGURE 7. Representative figure from U.S. Patent no. 10,979,033 issued to NXP.

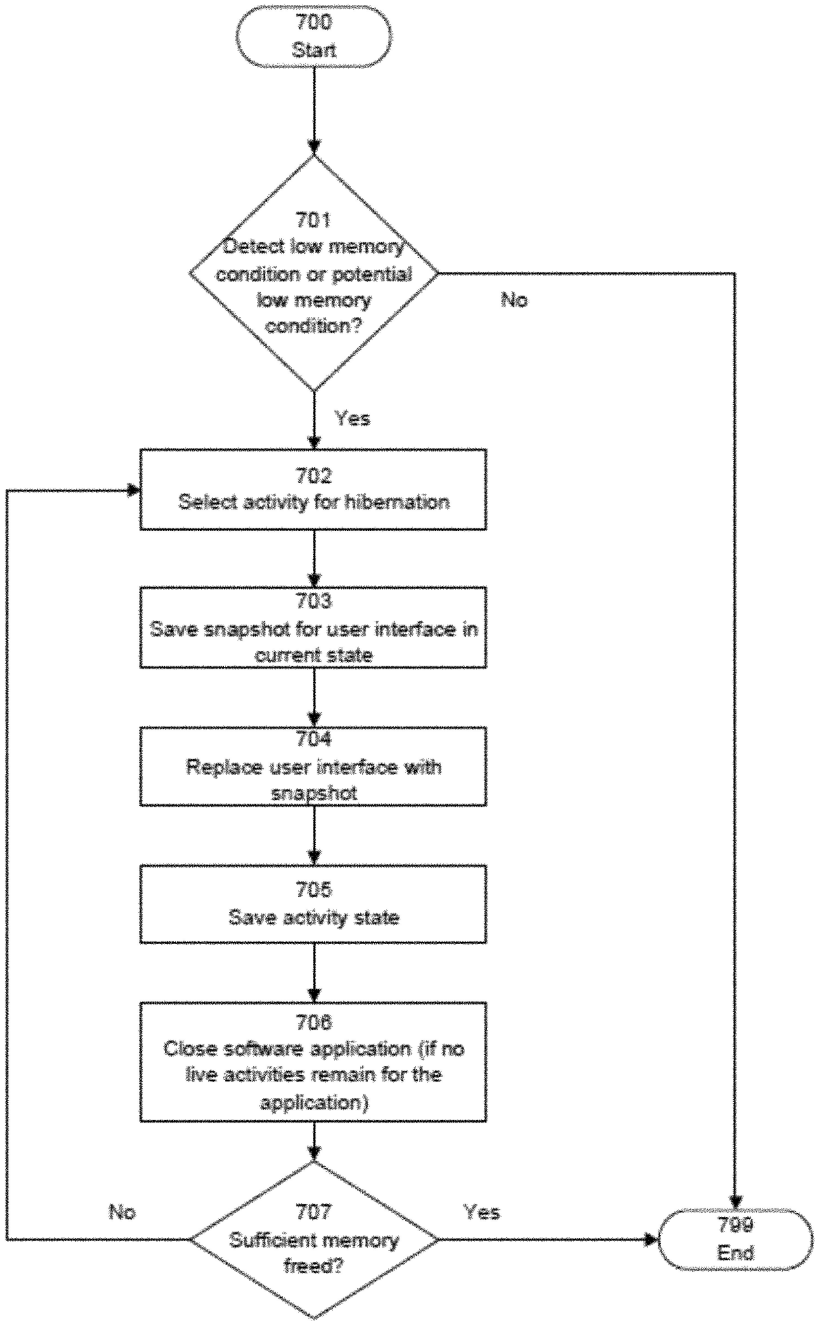

FIGURE 8. Representative figure from U.S. Patent no. 10,901,602 issued to Qualcomm.

ramp voltage relative to a reference voltage, and outputs a clock signal. In one embodiment, a comparator receives the reference voltage, one of the first and second ramp voltages, and outputs a comparison signal. Logic circuitry controls the ramp voltage generating circuitry to output one of the ramp voltages during one half of a clock cycle and to output the other ramp voltage during another half cycle of the clock signal based on the comparison signal and logic states of the logic circuitry.

Figure 7 is a "schematic of an exemplary current-controlled oscillator" ('033 Patent).

Assignee: Qualcomm
Number: 10,901,602
Filed: February 13, 2019
Issued: January 26, 2021

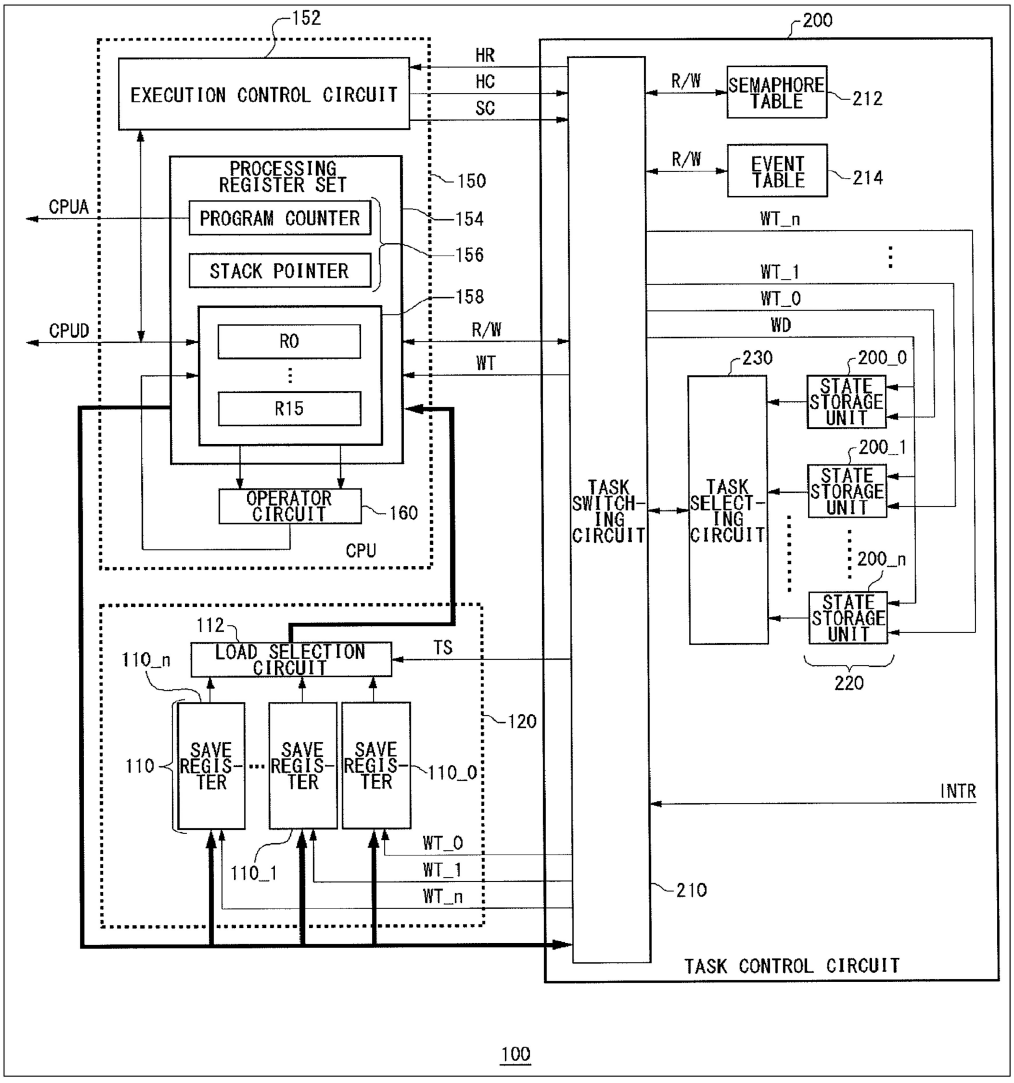

FIGURE 9. Representative figure from U.S. Patent no. 10,949,249 issued to Renesas.

Title: Selective hibernation of activities in an electronic device

Inventors: D. Shiplacoff (Los Angeles, CA), M. Duarte (Sunnyvale, CA), J. Lyon (Sunnyvale, CA)

Abstract: In an electronic device capable of running multiple software applications concurrently, applications, documents, cards, or other activities can be selected for hibernation so as to free up system resources for other activities that are in active use. A determination is made as to which activities should hibernate, for example, based on a determination as to which activities have not been used recently or based on relative resource usage. When an activity is to hibernate, its state is preserved on a storage medium such as a disk, so that the activity can later be revived in the same state and the user can continue with the same task that was being performed before the activity entered hibernation.

Figure 8 is a "flow diagram depicting a method for causing one or more activities to hibernate" ('602 Patent).

Assignee: Renesas+Dialog+IDT+Intersil

Number: 10,949,249

Filed: July 1, 2019

Issued: March 16, 2021

Title: Task processor

Inventor: N. Maruyama (Tokyo, JP)

Abstract: A task processor includes a CPU, a save circuit, and a task control circuit. A task control circuit is provided with a task selection circuit and state storage units associated with respective tasks. When executing a predetermined system call instruction, the

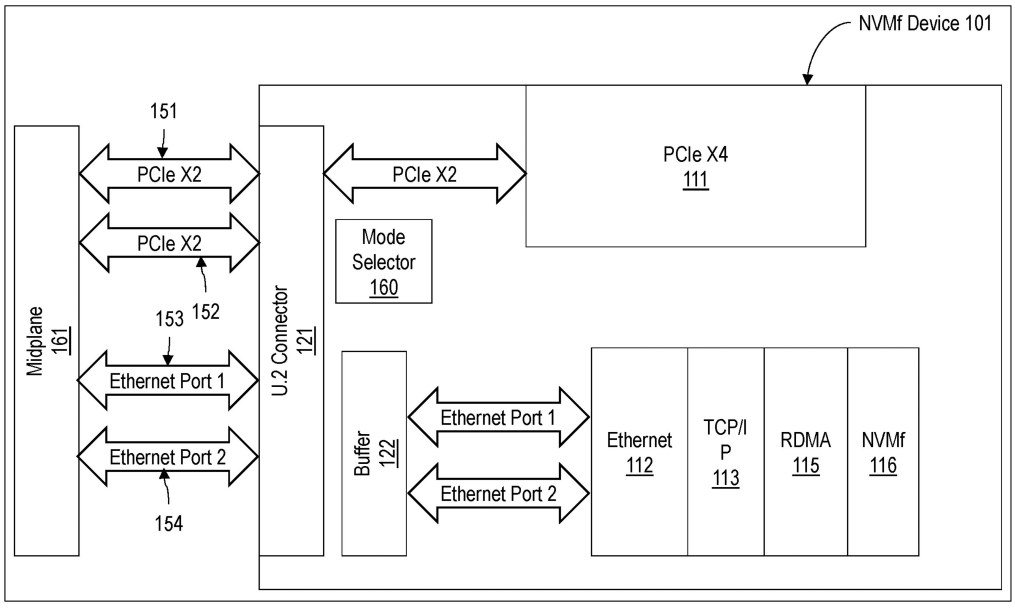

FIGURE 10. Representative figure from U.S. Patent no. 11,100,024 issued to Samsung.

CPU notifies the task control circuit accordingly. When informed of the execution of a system call instruction, the task control circuit selects a task to be subsequently executed in accordance with an output from the selection circuit. When an interrupt circuit receives a high-speed interrupt request signal, the task switching circuit controls the state transition of a task by executing an interrupt handling instruction designated by the interrupt circuit.

Figure 9 is a "circuit diagram of the task processor according to the basic implementation" ('249 Patent).

Assignee: Samsung
Number: 11,100,024
Filed: November 17, 2021
Issued: August 24, 2021
Title: System and method for supporting multipath and/or multimode NVMe over fabrics devices
Inventors: S. Olarig (Pleasanton, CA), F. Worley (San Jose, CA), S. Pham (San Ramon, CA)

Abstract: A system includes a fabric switch including a motherboard, a baseboard management controller (BMC), a network switch configured to transport network signals, and a PCIe switch configured to transport PCIe signals; a midplane; and a plurality of device ports. Each of the plurality of device ports is configured to connect a storage device to the motherboard of the fabric switch over the midplane and carry the network signals and the PCIe signals over the midplane. The storage device is configurable in multiple modes based on a protocol established over a fabric connection between the system and the storage device.

Figure 10 illustrates a "block diagram of an example NVMeoF device" ('024 Patent).

Assignee: SiFive
Number: 11,048,515
Filed: August 28. 2019
Issued: June 29, 2021
Title: Way predictor and enable logic for instruction tightly-coupled memory and instruction cache
Inventors: K. Asanovic (Berkeley, CA), A. Waterman (Berkeley, CA)

Abstract: Disclosed herein are systems and method for instruction tightly-coupled memory (iTIM) and instruction cache (iCache) access prediction. A processor may use a predictor to enable access to the iTIM or the iCache and a particular way (a memory structure) based on a location state and program counter value. The predictor may determine whether to stay in an enabled memory structure, move to and enable a different memory structure, or move to and enable both memory structures. Stay and move predictions may be based on whether a memory structure boundary crossing has occurred due to sequential instruction processing, branch or jump instruction processing, branch resolution, and cache miss processing. The program counter and a location state indicator may use feedback and be updated each instruction-fetch cycle to determine which memory structure(s) needs to be enabled for the next instruction fetch.

Figure 11 is a "diagram of an example flow and branch predictor" ('515 Patent).

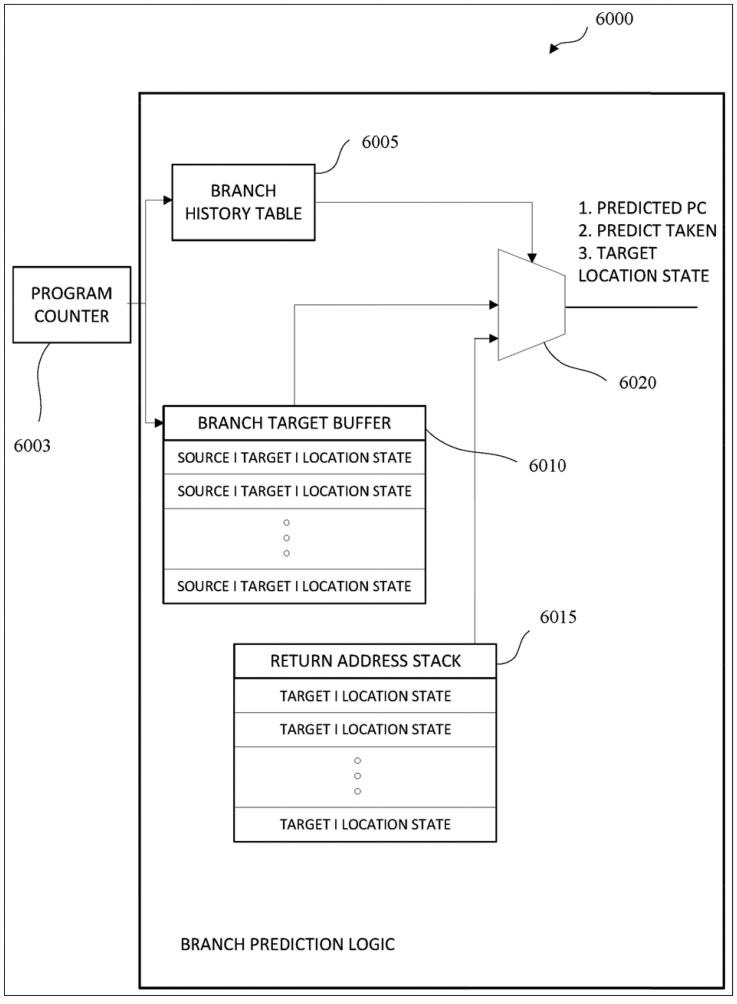

FIGURE 11. Representative figure from U.S. Patent no. 11,048,515 issued to SiFive.

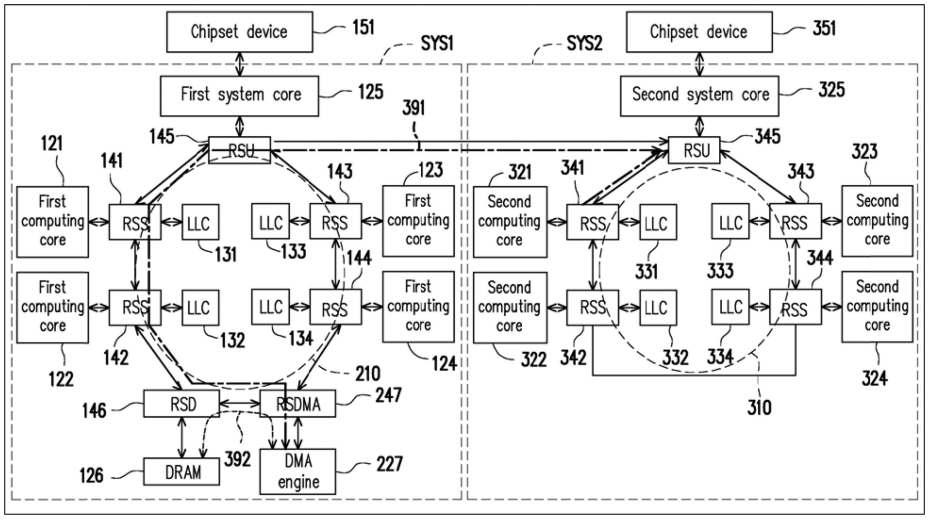

FIGURE 12. Representative figure from U.S. Patent no. 10,909,056 issued to Via.

Assignee: Via+Cyrix
Number: 10,909,056
Filed: December 22, 2018
Issued: February 2, 2021
Title: Multi-core electronic system
Inventor: W.-P. Chiang (New Taipei, TW)
Abstract: An architecture of a multicore electronic system is provided. The architecture includes a plurality of first computing cores, a first ring bus, a direct memory access (DMA) engine, and a DMA ring controller. The first computing cores are connected to the first ring bus. The DMA ring controller connects the DMA engine to the first ring bus. The first computing cores communicate with the DMA engine through the first ring bus and make the DMA engine perform a memory operation.

Figure 12 is a "schematic diagram of an architecture of a multi-core electronic system" ('056 Patent).

Part III of my series on the analysis of historical patenting behavior and patent characteristics of computer architecture companies will appear in the next issue. In particular, I will analyze the claims—including the number of total, independent, and dependent claims—in patents issued to these 18 companies that were filed between January 1, 1996, and December 31, 2020.

JOSHUA J. YI is a solo practitioner with The Law Office of Joshua J. Yi, PLLC, Austin, TX, 78750, USA, who serves as a court-appointed technical advisor for the Honorable Alan D Albright, United States District Judge for the Western District of Texas, Waco Division, Waco, TX, USA. His research interests include microarchitecture and performance methodology. Yi received a Ph.D. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, and a J.D. degree from the University of Texas at Austin, Austin, TX. Contact him at josh@joshuayipatentlaw.com.