2021 IEEE 6th International Conference on Intelligent Computing and Signal Processing (ICSP 2021)

Microprocessor Architecture and Design in Post Exascale Computing Era

WANG Di[1,a], LI Hong-Liang[2,a*]
[a]Jiangnan Institute of Computing Technology
Wuxi, China
*Corresponding author: wzc0425@mail.ustc.edu.cn

***Abstract*—In the post exascale computing era, the energy efficiency improvement speed of traditional complementary metal-oxide-semiconductor (CMOS) process has slowed down significantly. In order to realize the zettascale computing capacity in 2035, a great innovation is needed in the microprocessor architecture and design. This paper selects four aspects, which are low power consumption technology, near data processing (NDP) technology, interconnection centered design method and domain specific architecture (DSA), which has a broad development prospect, and focus on the energy efficiency benefits of each technology. Firstly, we analyze various traditional low power consumption technologies and near threshold computing (NTC) technology; secondly, we analyze the NDP technologies such as near memory computing, in memory computing and in network computing; thirdly, we analyze the low overhead network on chip (NOC), NOC supported by new process and cache coherent NOC technology; finally, we take the popular artificial intelligence (AI) processor as an example to analyze the DSA.**

Keywords-post exascale computing; microprocessor; architecture; design

I. INTRODUCTION

High performance computing has become the core support force in national security and national defense construction, economic construction, major projects, basic scientific research and other fields, which is of great significance to national industrial upgrading and structural adjustment. According to the international top 500 high performance computers (TOP500) released in November 2020, Japan's Fugaku high performance computer occupy the top with 442.01Pflop/s (1P=10^{15}) Linpack test performance[1], and there are plans to further upgrade to exascale (10^{18}) in the future. In fact, the HPL-AI test performance of Fugaku has reached 1.42Eflop/s, which has exceeded the exascale. Moreover, the United States, Europe and China have plans to build exascale computers in 2021-2023. It can be said that we are entering the post exascale computing era.

According to the data of the previous TOP500 rankings, the performance of the world's top high performance computers has basically increased by 1000 times every 10 years since the 1990s, exceeding the speed of Moore's Law[2]. In the middle and late years of 2010s, with the failure of Dennard Scaling[3] and the gradual slowing down of Moore's Law, the development speed slowed down significantly, and it is expected to increase by about 100 times every 10 years. Taking Sunway TaihuLight[4] released in 2016 as the symbol of 100Pflop/s, people expect to reach the level of zettascale (10^{21}) around 2035.

The characteristic size of complementary metal-oxide-semiconductor (CMOS) process has reached nanometer level, and the influence of short channel effect, quantum effect, parasitic effect and parameter instability on device performance has become very significant[5]. According to the prediction of the International Roadmap for Devices and Systems (IRDS)[6], there will be about six generations of process progress from 2017 to 2033. If the power consumption of each generation is reduced by 35%, the energy efficiency of the process can be improved by 13 times. However, the performance improvement from 100Pflop/s to 1Zflop/s needs 10000 times. Considering the contribution of process progress, there is still a huge gap of nearly three orders of magnitude, which needs to be filled in the microprocessor architecture and design level.

The purpose of this paper is to look forward to the architecture and design of microprocessor in the era of post exascale computing. In Section 2, various low power consumption technologies are analyzed, including traditional low power consumption technologies and near threshold computing (NTC) technology, In Section 3, various near data processing (NDP) technologies are analyzed, including near memory computing, in memory computing and in network computing. In Section 4, the interconnection centered design method is analyzed, and low overhead network on chip (NOC), NOC supported by new process and cache coherent NOC technology are introduced. In Section 5, we take the popular artificial intelligence (AI) processor as an example to analyze the domain specific architecture (DSA) is analyzed. Finally, summarizes the whole paper.

II. LOW POWER CONSUMPTION TECHNOLOGY

With the increasing integration and working frequency of microprocessors, the requirements of power consumption on power supply and cooling capacity, as well as the stability and reliability problems caused by thermal effect, have increasingly become the bottleneck of the development of high performance computing[7].

According to the generation type, the power consumption of integrated circuit can be divided into dynamic power consumption and static power consumption. Dynamic power consumption is caused by the change of input signal, including the charge and discharge of load capacitor and short-circuit current, which are called flip power consumption and short-circuit power consumption respectively. Static power

978-1-6654-0413-6/21/$31.00 ©2021 IEEE

2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP) | 978-1-6654-0413-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICSP51882.2021.9408861

consumption is caused by leakage current, including leakage current of gate, source and drain and subthreshold current when gate voltage is less than threshold[8].

Flip power consumption

$$P_{sw} = \alpha C_L V_{DD}^2 f \quad (1)$$

where, α is the switching activity, C_L is the load capacitance, V_{DD} is the power supply voltage, and f is the frequency.

Short-circuit power consumption

$$P_{dp} = \alpha t_{sc} V_{DD}^2 I_{peak} f \quad (2)$$

where, t_{sc} is the time for PMOS and NMOS to turn on at the sametime, I_{peak} is the maximum short-circuit current.

Static power consumption

$$P_{stat} = (I_{leak} + I_{sub}) V_{DD} \quad (3)$$

where, I_{leak} is the leakage current and I_{sub} is the subthreshold current. The subthreshold current is closely related to the threshold voltage. The lower the threshold voltage is, the higher the subthreshold current is. In addition, the static power consumption is exponentially related to temperature.

All kinds of low power consumption techniques are based on the analysis of the above power consumption sources.

A. *Traditional Low Power Consumption Technology*

1) *Dynamic Power Consumption Manage Technology*

To reduce the dynamic power consumption, the power supply voltage, load capacitance and switching activity can be reduced.

a) Power Supply Voltage (V_{DD}) Reduction

Reducing the power supply voltage can significantly reduce the dynamic power consumption. However, with the decrease of the power supply voltage, the delay of the circuit increases, which leads to the performance degradation. In order to maintain the performance of the processor, the following methods can be adopted: applying low voltage in non-critical circuits; reducing the threshold voltage to ensure the speed of the circuit; using parallel or pipeline structure to compensate for the reduction of circuit speed[9].

- Multi Voltage Domain: Low voltage is used for the device in the fast path, high voltage is used for the device driving large capacitance, and converter is inserted between low voltage domain and high voltage domain[10].
- Dynamic Voltage and Frequency Scaling: The processor has a very uneven workload when it is running. Keep high power supply voltage and high clock frequency at high load, and reduce power supply voltage and clock frequency at the same time when performing low load work, so as to effectively save energy consumption[11].
- Multi Threshold Transistor: The design is based on high threshold devices, and low threshold devices are used to optimize the timing critical path[12]. This technology is also an important method to reduce static power consumption.
- Structure Optimization: A parallel or pipelined structure can be used to replace the original circuit. Although the latter is larger and more complex, it can achieve the same performance at a lower power supply voltage and get positive benefits[10].

b) Load Capacitance (C_L) Reduction

The input capacitance of CMOS devices is directly proportional to the size of the device. Reducing the size of the device will reduce the speed, which needs to be tradeoff between performance and power consumption.

- Transistor Size Adjustment: Using small size transistor in non-critical path and large size transistor in critical path can effectively reduce the overall power consumption[10].

c) Switching Activity (α) Reduction

Reducing power consumption by reducing unnecessary signal flipping is the most important low power consumption means in architecture design and logic design, including different granularity methods from gate level to system level.

- Transistor Reordering: According to the characteristics of signal activity, the corresponding transistor reordering can greatly reduce their turnover. If the transistor with frequent flip is close to the output end of the circuit, it can prevent the high flip rate of one transistor from spreading to more transistors and bring more power consumption[7].
- Signal Path Equalization: Glitch is the main source of power consumption in complex structures such as arithmetic unit, which may be caused when the length of signal path varies greatly. Choosing the structure with signal path equalization can greatly reduce the probability of glitch[10].
- Operands Isolation: For the functional units without operation, the input is kept at 0 to prevent the turnover of the output signal, so as to reduce the unnecessary dynamic power consumption[7].
- Low Power Consumption Coding: The control logic of the processor is mainly realized by the finite state machine. One of the ways to reduce the power consumption of the control logic is to optimize the state coding mode of the finite state machine, and reduce the signal turnover rate of the circuit by reducing the average distance between two adjacent states[7].
- Clock Gating: Clock power consumption is an important part of processor power consumption. At present, clock power consumption of high performance processor can reach one fourth of the whole chip[13]. Shielding the clock signal in the idle module or reducing it to a very low frequency can save a lot of

power consumption. The flip-flop in the circuit will not flip, but its state value is still saved[12].

- Asynchronous Design: There is no global clock signal in asynchronous design, and the operation of the system is generated by the handshake signal between various components to drive a series of events, which reduces unnecessary flipping in synchronous design[7]. At present, the tradeoff between synchronous circuit and asynchronous circuit, the same frequency and different phase (mesochronous) and globally asynchronous locally synchronous (GALS) are widely used[13].

2) Static Power Consumption Manage Technology

To reduce static power consumption, we can use power off, increase transistor threshold and control chip temperature.

- Power Gating: Power gating is the most effective means to reduce static power consumption. Put the modules with basic synchronization in the working period in a power supply partition. When all the modules in a power supply domain do not need to work, the working voltage of the power supply domain can be turned off to eliminate the static power consumption of the module[8].
- Dynamic Threshold Adjustment: The transistor is used as a four-terminal device, and the threshold of transistor is controlled by substrate bias. For the calculation of low delay, the threshold value is reduced to its minimum value, while for low speed calculation, the threshold value can be increased to minimize leakage current[9].
- Dynamic Thread Assignment and Transfer: Because of the different computing tasks undertaken by each functional unit of the processor, the temperature distribution in space and time is uneven. The operation is transferred from the core with higher temperature to the core with lower temperature, which can effectively control the static power consumption[7].

B. Near Threshold Computing (NTC) Technology

1) Basic Concepts of NTC

Reducing the power supply voltage is the most direct means of low power consumption. Other low power consumption technologies will lead to the aggravation of the "dark silicon" problem, that is, although the chip integrates more transistors, only a small part of the transistors can work at the same time. However, the range of voltage reduction is limited. On the one hand, the decrease of voltage leads to the decrease of conduction current and the increase of circuit delay; on the other hand, the static power consumption in the subthreshold region increases exponentially with the decrease of voltage, and the decrease of voltage will increase the total power consumption.

NTC refers to the circuit where the power supply voltage drops to near the threshold voltage, and its voltage value is between the conventional voltage and the subthreshold voltage[14]. NTC is a tradeoff between performance and power consumption, which can achieve the optimal performance of energy efficiency. When the chip works near the threshold voltage, the energy efficiency can be significantly improved compared with the conventional voltage, as shown in Fig. 1.

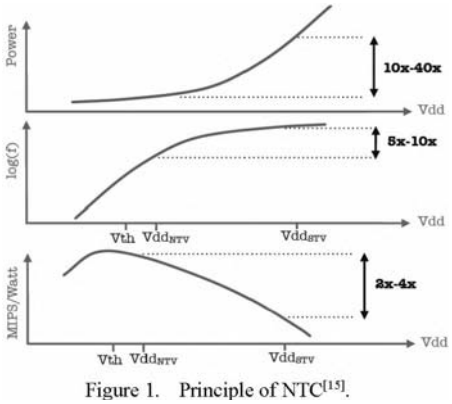

Figure 1. Principle of NTC[15].

Intel released a 32nm process IA-32 architecture processor in 2012[16]. The operating voltage range is 0.28V to 1.2V. When operating at the near threshold voltage of 0.45V, the energy efficiency is 4.7 times higher than that of the standard voltage. In 2015, the KAIST released the target recognition processor with 65nm process[17]. The operating voltage range is 0.5V to 1.2V. When operating at the near threshold voltage of 0.5V, the energy efficiency is 5.8 times higher than that of the standard voltage.

2) Main Challenges of NTC

Although the NTC can achieve 10 times of the standard voltage in power consumption, it will also cause many problems, including the significant decline of circuit performance, the increase of uncertainty of circuit behavior and the significant increase of the risk of circuit functional failure[18]. These problems come from delay deviation which is caused by process deviation, voltage deviation, temperature deviation and aging effect (PVTA). Delay bias has become the biggest challenge of NTC[19].

The process deviation comes from the manufacturing process, which results in the fluctuation of transistor length, width, oxide layer thickness and threshold voltage. The voltage deviation comes from the partial voltage of the interconnect resistance. The more transistors flipped at the same time, the higher the resistance voltage drop. The temperature deviation comes from the change of ambient temperature and the heating of the circuit itself. The aging deviation comes from the decay of transistor life cycle, which leads to the slowdown of transistor speed, the decrease of reliability, the increase of leakage current and the failure of function.

There are a lot of researches on anti-deviation technology in academic circles[19,20]. For example, the anti-fluctuation design technology at the process and device level can enhance the anti-deviation ability of the integrated circuit itself; the static anti-deviation technology can compensate the deviation of the critical path delay by statistical analysis optimization or chip test adjustment, so as to reduce the time series allowance reserved in the design; the dynamic anti-deviation technology based on time series prediction can dynamically predict the time series error through sensor, monitoring circuit, architecture and software information, so that the system can

adjust the voltage and frequency adaptively, so as to reduce the time series margin reserved in the design; the dynamic anti-deviation technology based on timing fault tolerance can detect and correct timing errors in real time, so that the system can still work normally in the case of timing errors, so as to adaptively eliminate the timing margin reserved in the design during operation, and fully improve the energy efficiency of the system.

3) Near Threshold Cache

Cache is the key module of the processor. In recent years, the proportion of area and power consumption is increasing, and the impact on the overall performance of the processor is becoming more and more significant. In the near threshold region, the stability and performance of static random access memory (SRAM) are very sensitive to process deviation, resulting in a sharp increase in the failure probability of SRAM cells[21]. This means that the pursuit of system energy efficiency greatly increases the design difficulty of cache (cache is mainly constructed by SRAM on chip). Research on improving the reliability of near threshold cache focuses on circuit level and architecture level[22].

The research of circuit level mainly focuses on the high reliability design of SRAM circuit at near threshold voltage, which improves the reliability of memory cells at near threshold voltage with large area overhead and delay increase.

The architecture level research mainly includes hybrid cell cache design, error correction, data mapping, data redundancy and cache management strategy redesign. The design of hybrid cell cache is based on the results of circuit level work, which designs cache with high reliability at near threshold voltage. Based on the research of error correction methods, different error correction mechanisms are usually designed by using various error correction code technologies to achieve high cache availability. Data mapping technology maps data to error free areas in the cache. The cache structure based on data redundancy places the redundant data in the low voltage area and the unique data in the high voltage area. Cache management strategy redesign technology can avoid storing data in the wrong cache line by adjusting cache manage strategy.

III. Near Data Processing (NDP)

In the traditional von Neumann architecture, computing, memory and communication are separated. With the development of semiconductor process, the improvement of memory and communication bandwidth is far behind the improvement of computing performance. Since 1990s, the "memory wall" and "communication wall" have become more and more key factors to restrict the performance improvement of processor. The more serious problem is that the energy consumption of data movement has even exceeded the calculation energy consumption. Taking Intel research on 7nm process processor as an example[23], the memory access power consumption reaches 45.5%, and the communication power consumption reaches 18.2%. This means that data movement is becoming a bottleneck for the improvement of processor performance and power efficiency.

Increasing cache level and capacity can reduce the data movement quantity to a certain extent, but it is far from fundamentally solving this problem. Therefore, the idea of NDP that closely integrated with data and processing, which has been widely concerned by academia and industry.

A. Near Memory Computing

The basic idea of near memory computing is to close the data to the computing unit, so as to reduce the delay and power consumption of data movement. In near memory computing, logic or processing units are closer to memory. However, memory and computing units are still separate parts[24].

1) Near Memory Computing Based on SRAM

Near memory computing based on SRAM is mainly a multi-level memory architecture. A series of cache or local memory are inserted between the computing unit and the main memory. The temporal and spatial locality of the program are used to reduce the data movement distance.

2) Near Memory Computing Based on DRAM

In the 1990s, in order to break through the limitation of memory wall, a large number of research on near memory computing technology based on dynamic random access memory (DRAM) appeared. For example, the IRAM proposed by Patterson et al.[25] is manufactured by standard DRAM process, and vector processor is integrated into memory chip to greatly reduce access delay of processor to memory and make full use of memory bandwidth; the FlexRAM proposed by Kang et al.[26] adopts a tightly coupled architecture, and the computing array composed of 64 reduced instruction set computing (RISC) processor cores and DRAM cells are interleaved, which can make deep use of DRAM memory bandwidth and obtain significant performance improvement in data mining, decision system and other applications. Due to the incompatibility between the DRAM process and the logic process of the processor core, and the problem of DRAM access is alleviated to a certain extent by increasing and optimized cache design, the related research has not been well applied[27].

In recent years, with the emergence of advanced packaging technology and three dimensional integration, DRAM based near memory computing technology has gained a new development opportunity. Hybrid memory cube (HMC) and high bandwidth memory (HBM) are two new types of stacked DRAM based on through silicon via technology.

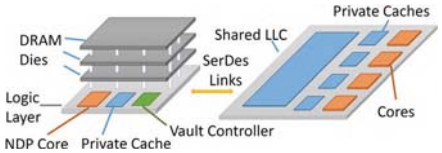

Figure 2. Near memory computing based on HMC[28].

HMC and HBM use stack to improve memory density and memory capacity, and improve memory bandwidth by high speed serial transmission or parallel width. They have the advantages of high integration, high bandwidth and high energy efficiency. Moreover, with the aid of stacking technology, the logical layer and memory layer of different manufacturing processes can be stacked together, and vertical

multiple memory layers correspond to one logical layer. Encapsulating the logical layer and memory layer reduces data access latency and power consumption[29].

B. In Memory Computing

There is still data movement from memory to computing unit in near memory computing. In order to eliminate the cost of data movement, a large number of in memory computing research has appeared in the academic circles. In memory computing is to perform computation in memory array, which realizes the complete integration of memory and computing.

1) In Memory Computing Based on SRAM

Jeloka et al.[30] divides the word line of 6T SRAM into left and right lines. At the same time, the differential sensitive amplifier is transformed into two single end cross coupled sensitive amplifiers. Through the control of two word lines and two sensitive amplifiers and the addition of logic gates at the output end of the sensitive amplifier, the basic logic of and, or, and non is realized. On this basis, Aga et al.[31] implements a compute cache that supports no carry multiplication. Eckert et al.[32] further implements addition, multiplication and subtraction operations, and proposed neural cache for deep learning algorithm.

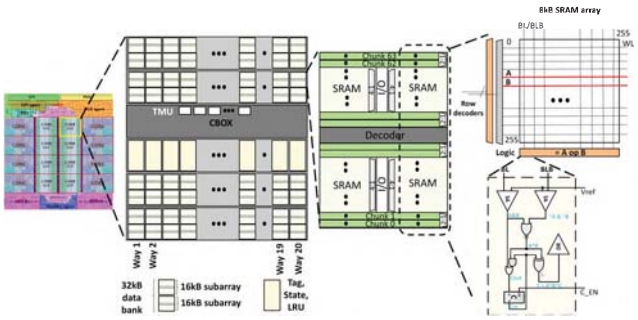

Figure 3. Neural cache architecture[32].

Kang et al.[33] vertically stored *n*-bit binary numbers by bit, so that different bits share the same bit line. Through the control signal, all memory bits are read at the same time, and the word line gating time from high bit to low bit of the data is binary weighted, so that there is a digital to analog conversion relationship between the voltage drop on the bit line and the stored number. Then the analog processing circuit module is used to process the voltage drop signal, so as to quickly calculate |A−B| and A•B (A and B are two vectors), and then the calculated value is converted into digital signal by analog-to-digital converter (ADC). By accelerating the calculation of vector distance and dot product in SRAM, the operation efficiency of AI algorithm can be greatly improved.

2) In Memory Computing Based on DRAM

The DRISA architecture proposed by Li et al.[34] implements convolutional neural network (CNN) computation based on DRAM process, providing large scale on chip memory and high computational performance. By redesigning the bit line of DRAM array, DRISA realizes simple logic and shift circuits, and uses these circuits to support complex operations such as addition and multiplication. Compared with GPU, DRISA can improve the energy efficiency of integer operation by 15 times.

3) In Memory Computing Based on NVM[29]

In recent years, nonvolatile memories (NVMs) such as flash, STTRAM, PCM and RRAM have developed rapidly. Due to the natural fusion of computing and memory, NVM is very suitable for in memory computing[35]. All kinds of NVMs are gradually moving towards the practical stage, and it is possible to apply them in microprocessor in the future.

In 2016, IBM created the first artificial nano scale random phase change neuron, which can be used to create artificial neurons. The membrane potential of the artificial neuron can be expressed by the phase structure of the nano phase change device. In 2018, IBM proposed to accelerate the training of fully connected neural network by PCM to perform calculation in the data storage location. The energy efficiency of the chip is 280 times that of GPU, and it can achieve 100 times of computing performance in the same area.

In 2010, HP Labs annouced that RRAM has Boolean logic operation function, which means that computing and memory functions can be integrated in RRAM. The first example of using RRAM to realize logical storage fusion is IMP proposed by Borghetti et al. In 2018, the PRIME architecture proposed by Xie et al. implemented neural network computing based on RRAM. The power consumption of PRIME can be reduced by 20 times and the speed can be increased by 50 times when it is fabricated in 15nm process.

C. In Network Computing

In network computing is a frontier topic in the field of high performance computing and AI. It effectively solves the problems of collective communication and point-to-point bottleneck in application, and provides a new idea and scheme for the scalability of data center. In network computing use network cards, switches and other network devices to calculate data online during data transmission, so as to reduce communication delay and improve overall computing efficiency[36]. The idea of in network computing can also be applied to the NOC with a single processor.

Zheng et al.[37] proposed a multi-mode data-flow transmission for many core processors. The data-flow transmission is asynchronous with the core pipeline, which makes it easy to prefetch data and effectively supports memory access delay hiding. The function of data-flow transmission is mainly completed by the stream transmission engine, which supports the concurrent processing of multiple transmissions. One of the main characteristics of data stream transmission is to support multiple stream transmission modes according to the structure characteristics and application requirements. These transmission modes can make the data distributed in a multi-dimensional way, effectively improve the efficiency of data localization and save memory access bandwidth.

Huang et al.[38] studied the idea of mapping the computing kernel to the memory network, so as to play a role in the data-flow mode of in network computing through NDP. They proposed an in network computing architecture called Active-Routing. By using the aggregation mode of arithmetic

operation intermediate results, the computing can be carried out in the process of approaching data processing. The architecture utilizes large-scale memory level parallelism and network concurrency to optimize aggregation operations along a dynamically constructed active routing tree. Compared with the advanced memory processing architecture, Active-Routing reduces the energy delay product by 80%.

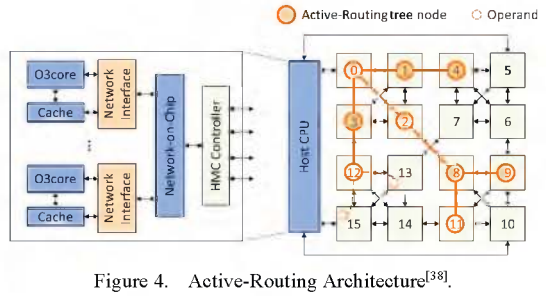

Figure 4. Active-Routing Architecture[38].

Compression processing in the process of data transmission is an important means to improve memory capacity and bandwidth and reduce data transmission power consumption. IBM has added a memory compression acceleration module NXU[39] to Z15 processor, which only increases the area of the processor by less than 0.5%, improves the compression efficiency by 388 times, and achieves a compression bandwidth of 280GB/s. NVIDIA also proposes a data compression technology called Buddy Compression for GPU to improve the effective capacity of on-chip memory and application performance.

IV. Interconnection Centered Design Method

At present, the development of processor has entered the stage of many core processor integrating dozens or even hundreds of cores. The design of many core processor needs to adopt the "interconnection centered" design method[41]. On the one hand, in the case of abundant computing resources, the efficiency of interconnection layer largely determines the performance of many core processors. On the other hand, the interconnection layer brings a lot of power consumption, so the design of low power many core processor must reduce the power consumption of the interconnection layer.

The improvement of processor integration means that the width of interconnects inside the chip becomes smaller and smaller, and the delay of signal transmission per unit distance increases accordingly. In the early CMOS circuits, the influence of the connection on the circuit performance and power consumption is ignored. The connection transmits signals at an almost infinite speed without power consumption and coupling effect. With the development of process technology, the influence of wire connection gradually appears. The parasitic effects such as capacitance, resistance and inductor affect the performance, power consumption and reliability of the system. In particular, the impact of global connection on delay and power consumption increases with the reduction of process scale [13].

At the beginning of the 21st century, NOC was proposed as a new design paradigm of interconnection communication on chip[42,43,44]. NOC replaces the traditional bus architecture with the packet communication architecture of point-to-point communication, which can reduce the chip area and power consumption, and improve the performance and scalability of the system.

A. Low Overhead NOC

In general, NOC uses routing nodes to connect processor cores into an interconnection network, and message exchange is used to communicate between cores, so as to form the on-chip communication system of processor. In the past 20 years, researchers have done a lot of research on topology, routing algorithm, router structure, switching technology, flow control technology, virtual channel technology, buffer implementation, error correction and coding, transmission link, network interface, QoS, program mapping, etc. Among them, bufferless router and router free network on chip are two important theoretical developments.

1) Bufferless Router[45]

Before bufferless router was put forward, network on chip (NOC) used wormhole or virtual channel router more often. The characteristic of NOC is that every input or output port of router contains buffered packets. Although the buffer can effectively improve the bandwidth utilization of the network, reduce packet loss and bypass routing, it also has the problems of consuming a lot of energy, complex flow control strategy and occupying a large area. Therefore, bufferless router emerges as the times require, which provides a low overhead solution for network on chip.

In the bufferless router, there is no extra buffer in the router except pipeline register, which can greatly reduce the energy consumption and area overhead of the router, and simplify the design of the router.

Bufferless routers can be divided into drop based routers and deflection based routers. In a packet loss based bufferless router, if a header microchip arrives at the router and the required output port is busy, all the microchips of the packet are discarded. In the bufferless router based on deflection routing, the router immediately forwards the packet to the next router after receiving it. In the case of competition, the packet can deviate from the shortest path routing.

2) Routerless NOC

Ring topology has long been considered as poor scalability. However, the isolated multi ring (IMR) architecture proposed by Liu et al.[46] can even support 1024 core processors. In IMR, any pair of cores are connected through at least one isolation ring, so that each packet can reach the destination without being transmitted from one ring to another. Therefore, IMR no longer needs expensive routers to build the grid network, which not only improves the network performance, but also reduces the hardware overhead. The experimental results show that IMR has significant advantages in bandwidth and delay, while reducing area and power consumption.

On the basis of IMR, Alazemi et al.[47] proposed the concept of routerless NOC. Routerless NOC completely eliminates the router, cleverly uses the routing resources, and achieves the same hop count and scalability as router based NOC. The evaluation results show that compared with the traditional grid,

the power consumption, area, zero load packet delay and bandwidth of the routerless NOC are reduced by 9.5 times, 7.2 times, 2.5 times and 1.7 times respectively.

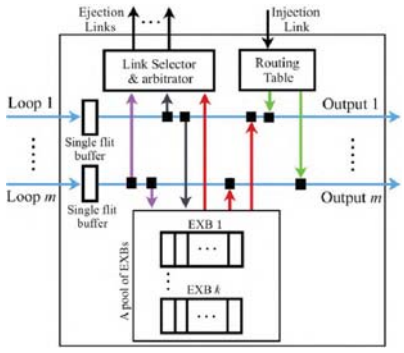

Figure 5. Interface module of routerless NOC[47].

From the development status, EDA tool support and physical design friendliness are still problems to be solved in routerless NOC.

B. *The Combination of NOC and New Process*

New processes bring new opportunities for the development of network on chip. The development of three dimensional integration technology and the progress of optical interconnection technology on chip will bring great changes to the network on chip architecture.

1) *Three Dimensional NOC*

Three dimensional integration is a technology to realize vertical interconnection between through silicon vias, which has the advantages of reducing the global interconnect length and increasing the interconnect density. Limited by the traditional two-dimensional architecture, NOC still cannot fundamentally avoid a series of related problems, such as global connection too long, connection delay, power consumption and so on. 3D NOC technology, which combines NOC and 3D integration technology, realizes inter core interconnection with 3D architecture to obtain better performance[48].

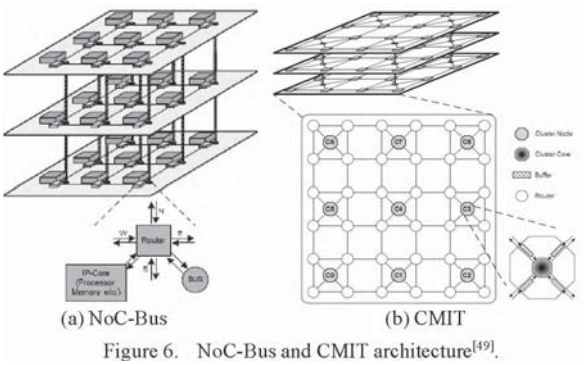

Figure 6. NoC-Bus and CMIT architecture[49].

Li et al.[50] proposed a structure called NOC-Bus for TSV delay and power consumption which are far less than the global interconnects in silicon chips. The structure uses mesh structure in each silicon chip, and uses bus to interconnect between silicon chips. Compared with 3D mesh, this structure can reduce the area and power consumption, and reduce the zero load delay. On the basis of NOC bus, Masoud et al.[49] proposed a 3D NOC structure called CMIT. The interconnection on each silicon chip also uses mesh network. Four nodes on each silicon chip are connected to a collection node, and the collection nodes on different silicon chips are connected by bus.

Park et al.[51] proposed a three-dimensional router architecture MIRA, which distributes the data channels between routers on different silicon wafers, and can reduce the area requirement and power consumption of 3D NOC. Kim et al.[52] proposed a dimension decomposition 3D NOC router 3D DIMDE, which decomposes the crossbar in the router into three modules, and the scale of each module is 2×2. Every time a message crosses a dimension, it will increase the delay of a clock cycle, which can achieve better performance under the dimension routing algorithm. Feng et al.[53] proposed a single cycle high-performance bufferless router for 3D NOC, which uses three segment permutation network instead of continuous switch distributor and 7×7 cross switch.

2) *Optical NOC*

In the field of high performance computer, optical interconnection has already shown its advantages in the network interconnection among cabinets, printed circuit boards and even chips. The progress of CMOS compatible nanophotonics technology provides the conditions for the development of optical network on chip. The bit rate transparency of optical media makes high-speed and low-power data transmission possible. The low loss characteristic of signal propagation in optical waveguide can increase the distance bandwidth product of the network, making the data can be transmitted further[11].

Some researches use passive wavelength switching to realize optical network on chip. Briere et al.[54] proposed a multilevel non-blocking optical router: λ-router. λ-router uses a passive switching structure based on wavelength, and uses wavelength division multiplexing (WDM) technology to realize non-blocking switching. Then, based on λ-router, an optical NOC using different wavelengths to realize optical switching is proposed. Batten et al.[55] proposed an optical interconnection structure based on silicon-based nano optical communication technology and using local mesh/global switching (LMGS) method, which connects the on-chip processor cores interconnected by grid structure to the off chip global switching structure.

There are also some researches on the implementation of on-chip optical networks using active optical switching units. Shacham et al.[56] proposed an optoelectronic hybrid network on chip, in which the optical interconnection network is used for high bandwidth message transmission, and the electrical network with the same topology is used to control the optical network and send short messages. Before sending the message through the optical network, the short control message is transmitted to the destination node through the electrical network, so as to reserve the optical device resources on the path. Mo et al.[57] proposed a hierarchical hybrid optoelectronic NOC architecture home, which uses optoelectronic hybrid router to realize wormhole switching in local network. At the same time, circuit switching is used to serve the global network.

C. Cache Coherent NOC

Cache coherence protocol is a mechanism to propagate the newly written value of one core to other cores to ensure that all cores see the coherent shared storage content. Considering that cache coherent programming mode can reduce the burden of programmers compared with message passing programming mode, cache coherence protocol will continue to exist in order to be compatible with a large number of historical code based on cache coherent programming mode. Therefore, it is necessary to provide effective communication support for cache coherence protocol[58].

Cheng et al.[59] observed that different cache coherence messages have different sensitivity to delay and bandwidth in the collaborative design of NOC and cache coherence protocol. Therefore, heterogeneous connection is proposed to transmit different types of messages. Delay sensitive messages are transmitted through low delay connection, and bandwidth sensitive messages are transmitted through high bandwidth connection. Eisley et al.[60] proposed to store the directory information of the directory cache coherence protocol in the NOC router, so as to reduce the latency of cache read-write transactions. Agarwal et al.[61] implemented message ordering at the NOC layer to support the implementation of broadcast cache coherence protocol on unordered network. On this basis, Agarwal et al.[62] further proposed to set a filter on the router to eliminate some unnecessary listening messages. Based on a similar idea, Jerger[63] uses a filter to eliminate redundant cache line void messages in coarse-grained directory cache coherence protocol.

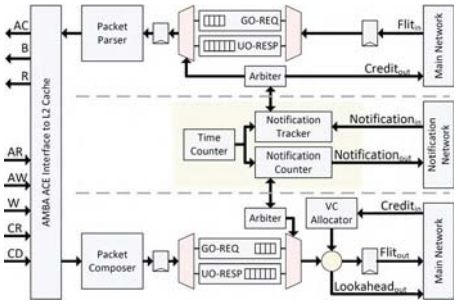

Figure 7. Network interface controller of SCORPIO architecture[64].

SCORPIO proposed by Daya et al.[64] is an architecture using broadcast cache coherence protocol, which has an independent fixed delay, bufferless mesh structure NOC and can realize distributed global sorting. Message delivery is separated from sorting, allowing messages to arrive at any time and in any order, while still maintaining the correct order. The main network is an unordered network and is responsible for broadcasting the actual coherence request to all other nodes. The notification network is used to broadcast the notification message of each coherence request sent in the primary network to all nodes. The notification message uses bit vector to represent the request source, so broadcasting can combine bit vectors by bit-or operation without competition.

Hu et al.[65] proposed a cooperative design of heterogeneous interconnection communication and cache coherence based on transmission line. Using transmission line to build NOC, it can provide low delay and low power consumption NOC; combining transmission line NOC with traditional mesh network, it can build on-chip heterogeneous interconnection system. It optimizes the adaptability of cache coherence protocol and reduces the maintenance cost of directory based cache consistency. Through the hardware real-time monitoring of the time locality of data, the system dynamically adjusts the storage strategy of shared read-write data to reduce the maintenance of cache coherence. According to the time delay sensitivity, messages with different characteristics can dynamically select the appropriate interconnection network transmission, so as to reduce the on-chip delay and improve the adaptability of cache coherence.

V. Domain Specific Architecture (DSA)

In the past 20 years, the technology of instruction level parallelism has not made great progress, and the improvement of processor performance mainly depends on the increase of the number of cores. However, the performance improvement efficiency of the processor is limited by the parallelism of the application itself. In addition, the increase of the number of cores cannot significantly improve the power efficiency of the system.

Processor has the advantages of high flexibility and programmability, but also has the problem of low power efficiency. Application specific integrated circuit (ASIC) gives up the programmable ability, but it can perform thousands of operations in parallel for specific applications, which greatly improves the performance and power efficiency. Compared with ASIC, the power efficiency of processor can be tens of times or even hundreds of times.

Pure processor has essential and insurmountable obstacles in performance and power efficiency, while pure accelerator for specific fields has great limitations in flexibility and programmability. Hennessy and Patterson, winners of Turing prize, have repeatedly emphasized that domain specific processors will be the main trend in the future[66,67]. In the future, the most important mode of chip system will be to integrate general multi-core processor and special accelerator to form a "general core+accelerator" system, so as to obtain the programmability and flexibility of general processor as well as the performance and power efficiency of application specific accelerator[68].

Taking the popular AI processor as an example, this paper introduces the architecture, design principle and implementation method of DSA. Relevant contents mainly come from [69], [70] and [71].

A. Case Study of DSA

In 2010, Temam[72] elaborated the significant influence that neural network may bring to hardware design in various fields of general computing and special computing. Since then, the design of AI chips has entered a period of rapid development, with the emergence of many DSA processors in the field of AI represented by Google's tensor processing unit (TPU)[69] and Huawei's Ascend[70].

1) TPU Architecture

The main modules of TPU include systolic array, vector computing unit, main interface module, queue module, unified buffer and DMA control module.

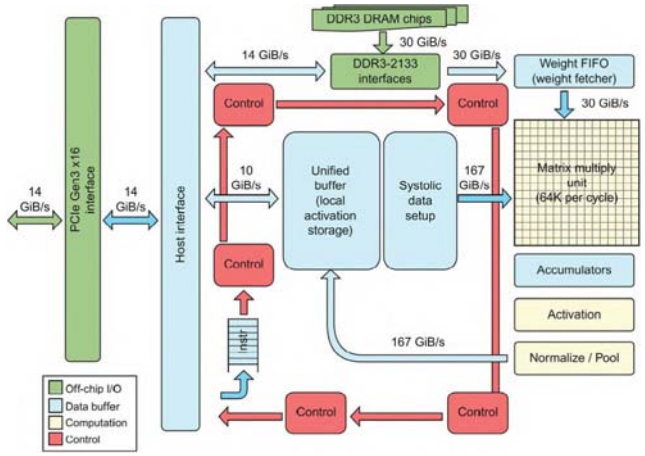

Figure 8. TPU architecture[69].

The main interface is used to obtain the parameters and configuration of neural network, such as the number of network layers, multi-layer weight and activation value. After receiving the read command, DMA control module will read and store the input feature and weight data in the unified on-chip buffer. At the same time, the main interface sends the execution instruction to the queue module. After receiving the instruction, the queue module starts and controls the calculation mode of the whole neural network, such as how the weights and eigenvalues enter into the pulsating array and how to accumulate them in blocks. The main function of the unified buffer is to store the intermediate results of input and output, and also to send the intermediate results to the systolic array again for the next layer calculation. The queue module can send control signals to unified buffer, pulse array and vector computing unit, and can also communicate directly with DMA control module and memory.

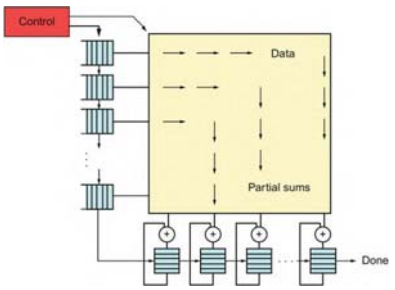

Figure 9. Systolic structure of TPU[69].

Systolic array is used to accelerate convolution. The main body of systolic array is a two-dimensional sliding array, in which each node is a systolic computing unit, which can complete a multiplication and addition operation in a clock cycle, and realize the right and down sliding transmission of data between the computing units of each row and column through horizontal or vertical data path.

2) Ascend Architecture

The main components of Ascend include control CPU, AI computing engine (including AI core and AI CPU), multi-layer cache or buffer, digital vision pre-processing (DVPP), etc.

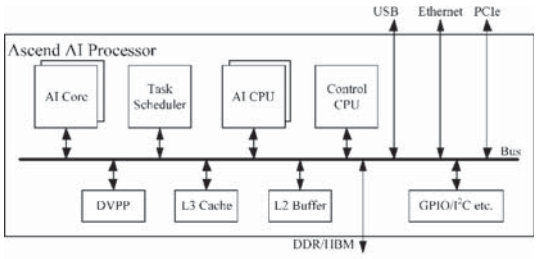

Figure 10. Ascend AI processor architecture[70].

Ascend AI processor integrates multiple CPU cores, each core has its own L1 and L2 cache, and all cores share an on-chip L3 cache. According to the function, the integrated CPU core can be divided into control CPU dedicated to control the overall operation of the processor and AI CPU dedicated to undertake non matrix complex computing. In addition to CPU, the real computing power of the processor is the AI core based on Da Vinci architecture.

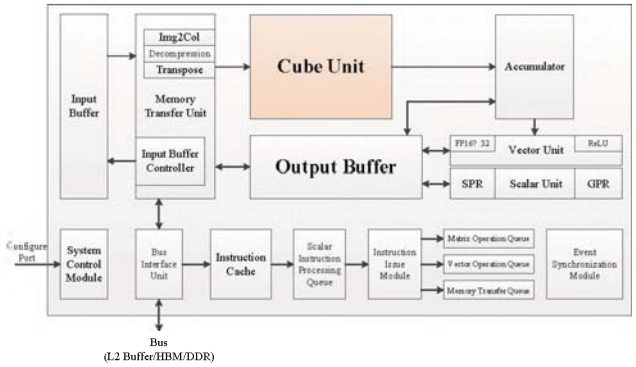

Figure 11. Da Vinci architecture[70].

Da Vinci architecture includes three basic computing resources: cube unit, vector unit and scalar unit. These three computing units correspond to three common computing modes: tensor, vector and scalar, forming three independent execution pipelines.

In order to coordinate the data transmission and transportation in AI core, a series of on-chip buffers are distributed around the three computing resources. Input buffer (IB) and output buffer (OB) are used to place the whole image feature data, network parameters and intermediate results. After the input buffer, a memory transfer unit (MTE) is set to realize data format conversion functions such as transpose with high efficiency. In addition, there are some high-speed register units used to provide temporary variables in each calculation unit.

B. Design Principle of DSA

Hennessy and Patterson systematically summarized the design principles of DSA, including the following five principles[71].

1) Using special memory to minimize the distance of data movement.

The multi-level cache in general-purpose microprocessors uses a lot of area and energy to optimize the data movement of programs. DSA compiler writers and programmers know their domain, so they don't need hardware to move data for them. Instead, software controlled memory is dedicated to specific functions within the domain and tailored to reduce data movement.

- TPU has a 24MB unified buffer, which stores the intermediate matrix and vector of MLP and LSTM, as well as the feature map of CNN. It is optimized for each 256B access. It also has 4MB accumulators, each 32-bit wide, which collect the output of the matrix cells and act as the input of the hardware for calculating the nonlinear functions. The 8-bit weights are stored in a separate off chip weight memory dram and accessed through the on-chip weight FIFO.
- In view of the characteristics of deep neural network, such as large number of parameters and many intermediate values, Ascend has equipped an 8MB on-chip buffer (L2 buffer) for AI computing engine to provide high bandwidth, low latency and efficient data exchange and access.

2) The resources saved from abandoning advanced microarchitecture optimization are put into more computing units or larger memory.

With out-of-order execution, multi-thread, multi-core, prefetch and address coalescing, architects translate the benefits of Moore's Law into resource intensive optimization of CPU and GPU. Considering the deeper understanding of program execution in these narrow areas, it is better to spend these resources on more processing units or larger on-chip storage.

- TPU provides 28MB of dedicated storage and 65536 8-bit ALUs, which means it has about 60% of the storage and 250 times the ALU of the server level CPU, although its size and power consumption are only half of the server level CPU. Compared with the server level GPU, the storage on chip of TPU is 3.5 times that of GPU, and Alu is 25 times that of GPU.
- 256 matrix calculation sub circuits are integrated in the matrix calculation unit of Acsend AI core, and each sub circuit realizes two 16 element vector dot products (each element is a 16 bit floating-point number). Two 16×16 matrices can be multiplied by one instruction, which is equivalent to 16^3=4096 multiplication and addition operations in a very short time.

3) Use the simplest parallel form that matches the domain.

The target domain of DSA almost always has inherent parallelism. The key decision of DSA is how to use this parallelism and how to open it to software. It is necessary to design DSA around the natural granularity of parallelism and simply expose the parallelism in the programming model.

- The performance of TPU is provided by a two-dimensional SIMD parallel processing unit. Its 256×256 matrix multiplication unit adopts pulsating organization and a simple instruction overlapping pipeline.
- In view of the fact that the demand of large computing power in AI applications can often be transformed into matrix operation, Ascend provides its performance through a fixed 16×16 matrix operation unit. This is similar to NVIDIA's Tensor Core[73].

4) Reduce data size and type to the simplest size and type required by the domain.

Applications in many fields are usually limited in storage. Therefore, by using narrower data types, effective storage bandwidth and on-chip storage utilization can be improved. Narrower and simpler data also allows designers to place more computing units in the same chip area.

- TPU mainly computes 8-bit integers, although it supports 16 bit integers and accumulates them in 32-bit integers.
- The matrix computing unit in Ascend AI core supports 8-bit integer and 16 bit floating-point computation, while the vector computing unit supports 16 bit and 32-bit floating-point computation as well as a variety of integer computation.

5) Using domain specific programming language to port code to DSA.

A typical challenge for DSA is to make applications run on a new architecture. In fact, domain specific programming languages have long been popular, such as halide for video processing and tensorflow for deep learning. Such a language makes it more feasible to port applications to DSA. In some areas, only a small number of applications need to run on DSA, which also simplifies the migration.

- TPU is programmed with TensorFlow programming framework. TensorFlow supports running on CPU, GPU and TPU, and its programming style is declarative programming.
- Acsend adopts MindSpore programming framework. MindSpore uses functional differentiable programming architecture to achieve dynamic static combination of development and debugging mode, automatically complete model segmentation and tuning, and provide consistent development, on-demand collaboration and flexible deployment functions.

In the field of AI, major companies develop programming frameworks for their own hardware. The work of various programming frameworks is basically similar. By defining a set of intermediate representations dedicated to deep learning, we can get through a process of DSL→Deep Learning IR→LLVM IR→Target, and add various optimizations in the middle.

C. Implementation Method of DSA

1) Implementation Method Based on IP Block

Amdahl's Law reminds us that the performance of the accelerator is limited by the rate of data transmission between the core and the accelerator. Integrating the core and accelerator into the same SOC will benefit the application.

This design is called IP block, which is usually specified by hardware description language to integrate into SOC. Many companies produce IP blocks, and other companies can buy these IP blocks to build SOC for their own applications without having to design everything themselves.

IP block must be scalable in area, power consumption and performance. For a new IP block, it is particularly important to provide a small resource version, because it may not have a good foothold in the SOC ecosystem. If resource requests are moderate, adoption is much easier.

2) Open Source, Extensible Instruction Set Architecture

For DSA designers, an open source and extensible instruction set architecture is needed.

On the one hand, a challenge for DSA designers is to determine how to work with the CPU to run the rest of the application, which means choosing the instruction set architecture of the CPU.

On the other hand, in order to cover as many applications as possible, universal instruction set architecture often needs to support thousands of instructions, which leads to the complexity of pipeline front-end design (finger fetching, decoding, branch prediction, etc.), which has a negative impact on performance and power consumption. Domain specific instructions can greatly reduce the number of instructions, increase operation granularity, integrate memory access optimization, and achieve energy efficiency improvement[74].

RISC-V is a free and open instruction set architecture, which reserves a lot of opcode space for domain specific coprocessor to add instructions, so as to achieve closer integration between core and accelerator. Due to the openness of RISC-V, designers can obtain the IP block of RISC-V for free and get the support of open source software.

3) Microarchitecture Design[27]

The microarchitecture design of DSA processor is a process of software and hardware co design. The design steps include: ①design and analyze the target application algorithm, locate the hot area of operation and data access storage in the target application algorithm, and identify the bottleneck of acceleration; ②transfer the software code of calculation time from the benchmark processor to the special functional unit module, and at the same time, update the data in the benchmark processor On the basis of processor, add pipeline, special register, local register, special operation unit and corresponding acceleration instructions to form a new hardware model; ③ design software development tools, mainly including instruction set simulator, software compiler, assembler and linker; ④simulate target application and evaluate acceleration performance based on new hardware model and software development tools; ⑤ according to the evaluation results, iterative optimization is carried out repeatedly to meet the preset target requirements; ⑥DSA processor is implemented based on FPGA or ASIC to further evaluate the speed, area and power consumption.

4) Software Stack

In order to make DSA processor play an excellent performance, it is very important to design a perfect software solution. A complete software stack includes a framework for computing resources and performance tuning, as well as various supporting tools.

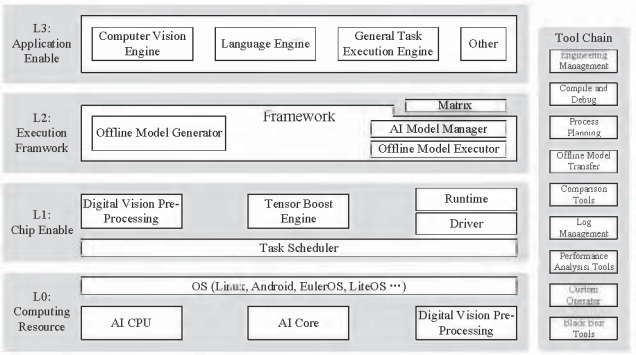

Figure 12. Software stack of Ascend AI processor[70].

Fig. 12 shows the software stack of Acsend AI processor, which is divided into two parts: neural network software flow and tool chain. Neural network software flow mainly includes matrix, framework, runtime, DVPP, tensor boost engine (TBE) and so on. Neural network software flow is mainly used to complete the generation, loading and execution of neural network model. The tool chain includes engineering management, compilation and debugging, process planning, offline model conversion, comparison tools, log management, performance analysis tools, custom operators and black box tools. Tool chain mainly provides auxiliary convenience for the realization of neural network.

VI. Conclusion

In the post exascale computing era, the development of CMOS process is difficult to maintain the original speed of energy efficiency progress. In order to achieve the goal of zettascale computing around 2035, we need to fill the energy efficiency gap of nearly three orders of magnitude in processor architecture and design. The full application of traditional low-power consumption technology and the development of NTC technology are expected to improve the energy efficiency by about one order of magnitude. NDP technology minimizes data mobility, while the interconnection centered design method minimizes the overhead of data mobility. The combination of the two technologies is expected to further improve the energy efficiency by about one order of magnitude. In recent years, DSA processor has developed rapidly. From the existing results, it has been proved that it can improve the energy efficiency by more than one order of magnitude. In summary, architecture and design innovation can support the sustainable development of processor technology in the post exascale era.

The development of technology will not stop, and the human pursuit of computing peak will continue. It can be predicted that in the post exascale computing era, the evolution of microprocessor architecture and design will be more exciting.

References

[1] H. Meuer, H. Simon, E. Strohmaier, et al., (2020, Nov 16). *TOP500 super-computer sites* [Online]. Available: http://www.top500.org.

[2] G. E. Moore, "Cramming more components onto integrated circuits," Electronics, Vol. 86, No. 1, pp. 82-85, 1965.

[3] R. H. Dennard, F. H. Gaensslen, H. Yu, et al., "Design of ion-implanted MOSFET's with very small physical dimensions," IEEE Journal of Solid-State Circuits, Vol. 9, No. 5, pp. 256-268, 1974.

[4] H. H. Fu, J. F. Liao, J. Z. Yang, et al., "The sunway taihulight supercomputer: system and applications," Science China. Information Sciences, Vol. 59, No. 7, pp. 1-16, 2016.

[5] M. M. Waldrop, "The chips are down for Moore's law," Nature, Vol. 530, pp. 144-147, 2016.

[6] IRDS, (2017). *International roadmap for devices and systems* [Online]. Available: https://irds.ieee.org/roadmap-2017.

[7] W. Guo, "Research on rtl-level and architecture-level key technique of low power for high performance processor design," Master Thesis, National University of Defense Technology, Changsha, China, 2011. (in Chinese)

[8] D. Wang, S. Shi and H. L. Li, "Three-dimensional technology and low power design: the prospect and challenge of integrated circuits," The 26th Annual Conference of the National Anti-Harsh Environment Computer, Chongqing, China, pp. 24-33, 2016. (in Chinese)

[9] A. Chandrakasan, W. J. Bowhill and F. Fox, Design of High-Performance Microprocessor Circuits. Wiley-IEEE Press, New York, USA, 2000.

[10] Rabaey, A. Chandrakasan and B. Nilolić, Digital Integrated Circuits: A Design Perspective, 2nd Edition, Pearson Education, Inc., New Jersey, USA, 2003.

[11] J. H. Wang, "Low-power on-chip networks in high-performance multi-core processors," Doctor Thesis, National University of Defense Technology, Changsha, China, 2014. (in Chinese)

[12] D. Wang and M. J. Wang, "High performance many-core processor: a design method based on ratio of performance and power estimation," High Performance Computing Technology, No. 233, pp. 27-33, 2015. (in Chinese)

[13] Z. Y. Yu, X. Y. Zeng and S. J. Wei, Microprocessor Design: Architecture, Circuit and Implementation. Science Press, Beijing, China, 2019. (in Chinese)

[14] I. C. Wey, P. J. Lin, B. C. Wu, et al., "Near-threshold-voltage circuit design: The design challenges and chances," International SoC Design Conference (ISOCC), Jeju, Korea (South), pp. 138-141, 2014.

[15] R. G. Dreslinski, M. Wieckowski, D. Blaauw, et al. "Near-threshold computing: reclaiming moore's law through energy efficient integrated circuits," Proceedings of the IEEE, Vol. 98, No. 2, pp. 253-266, 2010.

[16] S. Jain, S. Khare, S. Yada, et al. "A 280mv-to-1.2v wide-operating-range ia-32 processor in 32nm cmos," IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, USA, pp. 19-23, 2012.

[17] H. Yoo, Y. Kim and I. Hong, "A 0.5V 54μW ultra-low-power recognition processor with 93.5% accuracy geometric vocabulary tree and 47.5% database compression," IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, USA, pp.330-331, 2015.

[18] K. Yang, "Low power sram research and design under near-threshold voltage supply," Master Thesis, Shanghai Jiao Tong University, Shanghai, China, 2011. (in Chinese)

[19] S. Wang, "Research on near-threshold energy-efficient processor design based on timing error resilience," Doctor Thesis, Zhejiang University, Hangzhou, China, 2017. (in Chinese)

[20] W. Jin, "Research on ultra-low power pvt tolerant circuits design techniques," Doctor Thesis, Shanghai Jiao Tong University, Shanghai, China, 2017. (in Chinese)

[21] Z. M. Sun, "Near-threshold sram failure estimation and error-tolerant design for convolutional neural networks," Master Thesis, Southeast University, Nanjing, China, 2019. (in Chinese)

[22] Z. G. Wei, "Research on fault tolerance of cache at near-threshold voltage," Master Thesis, Wuhan University of Technology, Wuhan, China, 2018. (in Chinese)

[23] S. Borkar, "Exascale computing - a fact or a fiction?," IEEE International Parallel and Distributed Processing Symposium, Cambridge, USA, pp. 3-3, 2013.

[24] S. Jain, S. Sapatnekar, J. P. Wang, et al., "Computing-in-memory with spintronics," Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, pp. 1640-1645, 2018.

[25] D. Patterson, T. Anderson, N. Cardwell, et al., "Intelligent ram (iram): chips that remember and compute," IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, USA, pp. 224-225, 1997.

[26] Y. Kang, W. Huang, S. M. Yoo, et al., "Flexram: toward an advanced intelligent memory system," IEEE International Conference on Computer Design, Montreal, Canada, pp. 192-201, 1999.

[27] Z. Y. Yu, X. Y. Zeng and S. J. Wei, Microprocessor Design: Architecture, Circuit and Implementation. Science Press, Beijing, China, 2019. (in Chinese)

[28] P. A. Tsai, C. P. Chen and D. Sanchez, "Adaptive scheduling for systems with asymmetric memory hierachies," IEEE/ACM International Symposium on Microarchitecture (MICRO), Fukuoka, Japan, pp. 641-654, 2018.

[29] F. Y. Chen and H. B. Tan, "Research on the architecture of processing in memory towards computation," High Performance Computing Technology, No. 260, pp. 1-9, 2019. (in Chinese)

[30] S. Jeloka, N. B. Akesh, D. Sylvester, et al., "A 28nm configurable memory (tcam/bcam/sram) using push-rule 6t bit cell enabling logic-in-memory," IEEE Journal of Solid-State Circuits, Vol. 51, No. 4, pp. 1009-1021, 2016.

[31] S. Aga, S. Jeloka, A. Subramaniyan, et al., "Compute caches," IEEE International Symposium on High Performance Computer Architecture (HPCA), Austin, USA, pp. 481-492, 2017.

[32] C. Eckert, X. W. Wang, J. C. Wang, et al., "Neural cache: bit-serial in-cache acceleration of deep neural networks," ACM/IEEE International Symposium on Computer Architecture (ISCA), Los Angeles, USA, pp. 383-396, 2018.

[33] M. Kang, S. K. Gonugondla, A. Patil, et al., "A multi-functional in-memory inference processor using a standard 6t sram array," IEEE Journal of Solid-State Circuits, Vol. 53, No. 2, pp. 642-655, 2018.

[34] S. C. Li, D. M. Niu, K. T. Malladi, et al., "Drisa: a dram-based reconfigurable in-situ accelerator," IEEE/ACM International Symposium on Microarchitecture (MICRO), Boston, USA, pp. 303-314, 2017.

[35] Q. Li, J. Zhong and X. Li, "Memory management mechanism for hybrid memory architecture based on new non-volatile memory," Acta Electronica Sinaca, Vol. 47, No. 3, pp. 664-670, 2019.

[36] NVIDIA, (2020). *What is in-network computing?* [Online]. Available: https://zhuanlan.zhihu.com/p/166266347. (in Chinese)

[37] F. Zheng, H. L. Li, H. Lv, et al. "Cooperative computing techniques for a deeply fused and heterogeneous many-core processor architecture," Journal of Computer Science and Technology, Vol. 30, No. 1, pp. 145-162, 2015.

[38] J. Y. Huang, R. R. Puli, P. Majumder, et al., "Active-routing: compute on the way for near-data processing," IEEE Symposium on High-Performance Computer Architecture (HPCA), Washington, DC, USA, pp. 674-686, 2019.

[39] B. Abali, B. Blaner, J. Reilly, et al., "Data compression accelerator on ibm power9 and z15 processors," ACM/IEEE International Symposium on Computer Architecture (ISCA), Valencia, Spain, pp. 1-4, 2020.

[40] E. Choukse, M. B. Sullivan, M. O'Connor, et al., "Buddy compression: enabling larger memory for deep learning and hpc workloads on gpus," ACM/IEEE International Symposium on Computer Architecture (ISCA), Valencia, Spain, pp. 926-939, 2020.

[41] Z. Y. Wang, S. Ma, L. B. Huang, et al., The Principle and Design of Networks on Chip. China Machine Press, Beijing, China, 2016. (in Chinese)

[42] W. J. Dally and B. Towles, "Route packets, not wires: on-chip interconnection networks," Proceedings of the 38th Design Automation Conference (DAC), Las Vegas, USA, pp. 684-689, 2001.

[43] A. Hemani, A. Jantsch, A. Postula, et al., "Network on chip: an architecture for billon transistor era," IEEE NorChip, pp. 1-8, 2000.

[44] L. Benini and G. De Micheli, "Powering networks on chips: energy-efficient and reliable interconnect design for socs," International Symposium on Systems Synthesis (ISSS), Montreal, Canada, pp. 33-38, 2001.

[45] C. C. Feng, "Research on key techniques of bufferless router for network-on-chip", Doctor Thesis, National University of Defense Technology, Changsha, China, 2012. (in Chinese)

[46] S. L. Liu, T. S. Chen, X. X. Feng, et al., "Imr: high-performance low-cost multi-ring nocs", IEEE Transactions on Parallel and Distributed Systems, Vol. 27, No. 6, pp. 1700-1712, 2016.

[47] F. Alazemi, A. Azizimazreach, B. Bose, et al., "Routerless network-on-chip," IEEE International Symposium on High Performance Computer Architecture (HPCA), Vienna, Austria, pp. 492-503, 2018.

[48] J. W. Wang, "Research of key issues on three dementional network on chip", Doctor Thesis, Nanjing University, Nanjing, China, 2012. (in Chinese)

[49] D. Masoud, E. Masoumeh, L. Pasi, et al., "Cmit-a novel cluster-based topology for 3d stacked architectures," IEEE International Conference on 3D System Integration, Munich, Germany, 2010.

[50] F. Li, C. Nicopoulos, Richardson, et al., "Design and management of 3d chip multiprocessors using network-in-memory," IEEE/ACM International Symposium on Computer Architecture (ISCA), Boston, USA, pp. 130-141, 2006.

[51] D. Park, S. Eachempati, R. Das, et al., "Mira: a multi-layered on-chip interconnect router architecture," IEEE/ACM International Symposium on Computer Architecture (ISCA), Beijing, China, pp. 251-261, 2008.

[52] J. Kim, C. Nicopoulos, D. Park, et al., "A novel dimensionally-decomposed router for on-chip communication in 3d architecture," IEEE/ACM International Symposium on Computer Architecture (ISCA), San Diego, CA, USA, pp. 138-149, 2007.

[53] C. C. Feng, Z. H. Lu, A. Jantsch, et al., "A 1-cycle 1.25 ghz bufferless router for 3d network-on-chip," IEICE Transactions, Vol. 95-D, No. 5, pp. 1519-1522, 2012.

[54] M. Briere, B. Girodias, Y. Bouchebaba, et al., "System level assessment of an optical noc in an mpsoc platform," Design, Automation & Test in Europe Conference & Exhibition (DATE), Nice, France, pp. 1-6, 2007.

[55] C. Batten, "Building manycore processor-to-dram networks using monolithic silicon photonics," IEEE Micro, Vol. 29, No. 4, pp. 8-21, 2009.

[56] A. Shacham, K. Bergman and L. P. Carloni, "On the design of a photonic network-on-chip," International Symposium on Networks-on-Chip (NOCS), Princeton, USA, pp. 53-64, 2007.

[57] K. H. Mo, Y. Y. Ye, X. W. Wu,et al. "A hierarchical hybrid optical-electronic network-on-chip," IEEE Computer Society Annual Symposium on VLSI, Lixouri, Greece, pp. 327-332, 2010.

[58] S. Ma, "Research on the Key Techniques of Routing Algorithm and Flow Control Optimizations for Cache-Coherent Networks-on-Chip", Doctor Thesis, National University of Defense Technology, Changsha, China, 2012. (in Chinese)

[59] L. Q. Cheng, N. Muralimanohar, K. Ramani, et al., "Interconnect-aware coherence protocols for chip multiprocessors," IEEE/ACM International Symposium on Computer Architecture (ISCA), Boston, USA, pp. 339-351, 2006.

[60] N. Eisley, L. Peh and L. Shang, "In-network cache coherence," IEEE/ACM International Symposium on Microarchitecture (MICRO), Orlando, USA, pp. 321-332, 2006.

[61] N. Agarwal, L. Peh and N. K. Jha, "In-network snoop ordering (inso): snoopy coherence on unordered interconnects," IEEE International Symposium on High Performance Computer Architecture (HPCA), Raleigh, USA, pp. 67-78, 2009.

[62] N. Agarwal, L. Peh and N. K. Jha, "In-Network Coherence Filtering: Snoopy coherence without broadcasts," IEEE/ACM International Symposium on Microarchitecture (MICRO), New York, USA, pp. 232-243, 2009.

[63] N. Enright Jerger, "SigNet: Network-on-chip filtering for coarse vector directories," Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, pp. 1378-1383, 2010.

[64] B. K. Daya, C. H. O. Chen, S. Subramanian, et al. "SCORPIO: A 36-Core Research Chip Demonstrating Snoopy Coherence on a Scalable Mesh NoC with In-Network Ordering," International Symposium on Computer Architecture (ISCA), Minneapolis, USA, pp. 25-36, 2014.

[65] Q. Hu, P. Liu, M. C. Huang, et al., "Exploiting transmission lines on heterogeneous networks-on-chip to improve the adaptivity and efficiency of cache coherence", International Symposium on Networks-on-Chip (NOCS), Vancouver, Canada, pp. 1-8, 2015.

[66] J. Hennessy, D. Patterson, "A new golden age for computer architecture: domain-specific hardware/software co-design, enhanced security, open instruction sets, and agile chip development," Turing Lecture in International Symposium on Computer Architecture (ISCA), Los Angeles, 2018.

[67] D. Patterson, "50 years of computer architecture: from then mainframe cpu to the domain-specific tpu and the open risc-v instruction set," IEEE International Solid-State Circuits Conference (ISSCC), San Francisco, USA, pp. 27-31, 2018.

[68] R. I. D. Tullsen, "Heterogeneous computing," IEEE Micro, Vol. 35, No. 4, pp. 4-5, 2015.

[69] N. P. Jouppi, C. Young, N. Patil, et al., "In-datacenter performance analysis of a tensor processing unit," ACM/IEEE International Symposium on Computer Architecture (ISCA), Toronto, Canada, pp. 1-12, 2017.

[70] X. Y. Liang, Ascend AI Processor Architecture and Programming Principles and Applications of CANN. Tsinghua University Press, Beijing, China, 2019. (in Chinese)

[71] J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, Sixth Edition. Morgan Kaufmann Publishers, Cambridge, USA, 2019.

[72] O. Temam, "The rebirth of neural networks," ACM/IEEE International Symposium on Computer Architecture (ISCA). Saint-Malo, France, pp. 349-349, 2010.

[73] NVDIA, (2018). *NVIDIA Turing GPU Architecture* [Online]. Available: http://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf.

[74] Y. G. Bao, (2021). *After multi-core, what is the development direction of CPU?* [Online]. Available: https://www.zhihu.com/question/20809971. (in Chinese)