Real-time Buffering, Parallel Processing and High-Speed Transmission System for Multi-Channel Video Images Based on ADER

MingWang Chen[a], Jianping Xiong[b], Cheng Ma[c], Qi zhao[d] and Sen Feng[e]

Department of Precision Instuments, Tsinghua University Beijing, China

[a]cmw17@mails.tsinghua.edu.c, [b]xiongjp@tsinghua.edu.cn, [c]macheng@mail.tsinghua.edu.cn,

[d]zhao-q16@mails.tsinghua.edu.cn, [e]fengs17@mails.tsinghua.edu.cn

Abstract. Aiming at the problem that the brain inspired multi- core processing chip processes high-speed images in real time, a real-time data buffer of multi-channel high-definition images is designed, and the image is processed and transmitted at the pixel level. The design uses field-programmable gate array FPGA as the processing platform, and uses Address-Data-Event Representation (ADER) as a model to receive video stream data in real time, and the advantages of FPGA high-speed parallel processing are preliminary. Compression and grayscale processing are performed to continuously buffer and send and receive data in real time. In the data transmission, in order to adapt the multi-core processing chip parallel computing, the frame of data transmission protocol is customized at the bottom layer to realize continuous buffering, parallel processing and real- time transmission and reception of multi-channel high-definition camera data.

Keywords: Image caching and processing, High-speed trans- mission, Address-Data-Event Representation (ADER), Field programmable gate array (FPGA), brain inspired processing chip.

1. Introduction

Recently, with the development of big data and artificial intelligence algorithms, image target detection has been applied in more and more fields. Especially in 2012, the Hinton team participated in the ImageNet[1] image recognition competition for the first time, and through the construction of the CNN network AlexNet Winning the championship, the neural network began to receive widespread attention. With the development of artificial intelligence algorithms, it has become a reality to implement neural network algorithms on traditional architecture computers. However, how to map neural network computations to brain inspired architecture processing chips and use them in practical engineering applications has become a major problem. First of all, the neural network algorithm requires a large amount of parallel computing and processing, which requires large space hardware support, and the power consumption is also large. The traditional von Neumann architecture-based computer has poor parallel computing capability, which is difficult to meet the needs of neural networks. A large number of parallel computing, on the traditional von Neumann architecture computer neural network needs to spend a lot of time to calculate, and it also has limitation in real-time performance, although the current algorithm can have better real-time performance by reducing the number of layers or structure of the network, but its accuracy is reduced while the real-time performance is increasing.

In 2011, IBM first introduced the embedded brain inspired processing chip prototype TrueNorth[2], which broke the traditional von Neumann frame, enabling neural networks to be built on brain-based architecture-based many-core processing chips. Since then, various domestic companies and research units have also developed chips of various architectures, such as Intel’s neuromorphic chip Loihi[3], the Chinese Academy of Sciences’ Cambrian[4], Manchester University’s SpiNNaker chip[5], and Tsinghua University’s TianjiC series chips[6]. brain inspired processing chips can not only improve the computing speed of traditional computers, but also reduce power consumption, support deep learning algorithms, and implement deep learning algorithms for embedded applications. Although the real-time processing speed of the data on the chip is much improved, the current technology still connects the brain inspired processing chip to the CPU, acquires the video data through a conventional computer, processes it, and then sends the data to the brain inspired processing chip. As a result, it still needs to consume a large amount of CPU resources in the data acquisition phase, and the real-time and continuity of its processing is limited. How to improve the massive data collected by the sensor for real-time transmission and parallel processing is a key factor to ensure the processing speed and real-time performance of the brain inspired processing chip processing the image.

Based on the advantages of FPGA parallel processing, this paper proposes an ADER model and continuously acquires and caches multiple video data in real time through the ADER model. The image is cropped and compressed at the pixel level to adapt to the brain inspired processing chip, and the data is sent to the brain inspired processing chip in real time through a high-speed transmission carrier.

2. Ader Principle

Based on the Address-Event Representation (AER) model[7], this paper proposes an Address-Data-Event Rep- resentation (ADER) model. AER is a prototype of pseudo- biological neurology. After an event occurs, it can generate a pulse signal. The pulse signal can trigger the address and attribute of the peripheral address encoder coding pulse, and can pass the pulse signal property through the pulse. The frequency of the signal or the time interval of each pulse is expressed.

Based on the AER model, this paper proposes an ADER model and uses it for real-time acquisition and caching of video stream data from multiple cameras. As shown in Fig. 1, the video stream data triggers the event encoder. The event encoder generates an event in real time according to the video stream data. The content of the event includes the coordinate position of the pixel and the RGB value of the pixel. The address data encoder will acquire multiple events, and encode the event into a sequence of addresses and data according to the chronological order of the events, send them to the memory decoder, and store the data in the corresponding memory.

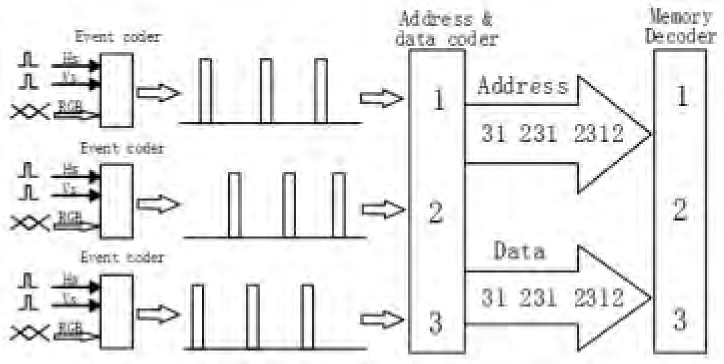


Fig. 1 ADER principle

The ADER method can eliminate redundant data and reduce the amount of data transmitted. In the case that the input events are not seriously competitive, the data of different video data sources can be acquired in real time and stored in the memory. When the speed of storing data is slower than the speed of the data source, the system will compete, and the events will conflict with each other, and the control module needs to arbitrate.

3. System Structure And Algorithm Design

3.1 Event Coding

Event encoding converts the changes produced by the data source into ADER's event structure based on the structure of the data source. Taking video data as an example, an event generated by a video stream includes coordinates of pixel points, and pixel values of pixel points. The video stream data source includes Hsync line sync signal, Vsync field sync signal, DE image enable signal, LLC image clock signal, and RGB pixel signal[8], as in Fig. 2 show.

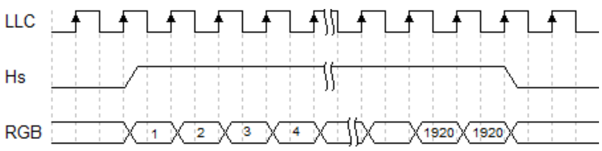


Fig. 2 Timing of vedio data

The event encoding first acquires the LLC clock signal, and the row signal Hsync and the field signal Vsync trigger the event encoding controller[9]. The event encoding controller arbitrates each video data and encodes it to generate a cor- responding event. When there is a line signal, the line pixel is cleared and starts counting. During the period when the line signal is kept high, the rising edge of each clock is incremented by one; when there is a falling edge of the field signal, it indicates a frame signal. At the end, the line and field counters are cleared. Finally, the pixel value of the pixel and the position information of the corresponding pixel are output. Suppose that the row counter is x and the column counter is y at a certain time, then the position of the pixel (x, y) can be calculated according to the value of the row counter and the column counter. So the event output at this moment is $s\,(n, x, y, rgb)$. Where n is the event number.

3.2 Address-Data Coding and Arbitration

The main function of the address data encoding is to generate the address and data of the memory for the event generated by the video data source obtained by the event encoding. Suppose the event generated at a certain moment is $s\,(n, x, y, rgb)$, video clarity is 1080p, That is, its image size is 1920 ×1080, and the corresponding memory address and data can be generated according to its event.

$$\begin{cases} Addr = y \times 1920 + x + 1920 \times 1080 \times (n-1) \\ data = rgb \end{cases} \tag{1}$$

Since the data generation of each video data source is random, when the address data is encoded, each event en- coding output is asynchronous, and the time and frequency of the event occurrence are not fixed, and if the simultaneous event request arrives, it will be generated. In case of conflict, arbitration is required at this point to allow the address- data encoder to prioritize an event. According to different systems and different requirements, the way of arbitration is different, and the way of arbitration will affect the coding time. At present, there are three commonly used arbitration schemes, namely, the right of arbitration, regional priority, and the priority of event concentration[10]. Rotational arbitration method and event concentration priority are usually applied to visual image acquisition systems that require video panorama information and have lower frame rate requirements. For the system of this paper, video data needs to be collected in real time, and the frame rate of video is high. Therefore, regional priority arbitration is adopted. The arbitration controller will acquire events in real time. When multiple events arrive at the same time, the arbitration controller will detect the position of each pixel that generates the event, and preferentially encode the event generated by the pixel near a certain position.

3.3 Image Scaling Algorithm and Grayscale Processing

Since the current brain inspired computing chip processes the image data, the neural network structure needs to be built first, which makes the brain inspired computing chip have a constant requirement on the image data size when processing the image data, so it is necessary to carry out the preliminary image of the high-definition image. Scale to match the image interface of the brain inspired computing chip. For pixel-level image scaling, the pixels of the image are actually reduced, thereby adjusting the size of the digital image. For the image scaling algorithm, the algorithm structure based on image pyramid is generally used[11]. As shown in Fig. 3, the schematic diagram of the image pyramid, the bottom layer is the original image, the higher the number of upper layers, the smaller the pixel resolution and the smaller the image.

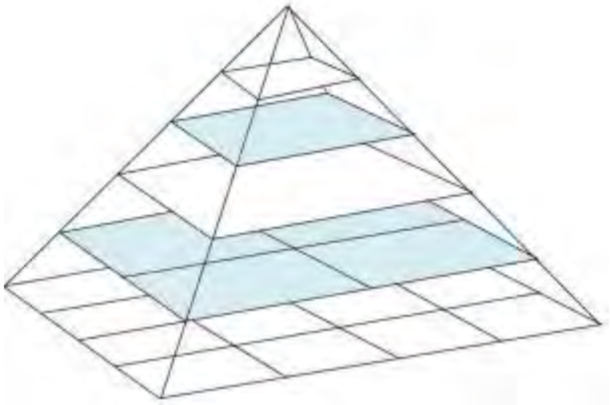


Fig. 3 Image pyramid structure

In order to reduce the time required for data processing and increase the real-time performance of the system, this paper uses the method of sampling to reduce the image. Assuming that the size of original image is $x \times y$, and the size needs to be reduced to $m \times n$, according to the pyramid principle,

$$g_1(i,j) = g_0(\frac{x}{m}(i-\frac{1}{2}), \frac{y}{n}(j-\frac{1}{2})) \tag{2}$$

By reducing the image by means of snapshots, the process- ing time of the image by the system can be minimized, and the method of sampling is compared to other methods such as block averaging, although to some extent it is obtained. The image is less smooth, but it can retain the characteristics of the image to the greatest extent, which has certain advantages for brain inspired computing chips.

The RGB image is converted to an 8-bit grayscale image with a famous mental formula:

$$Gray = R \times 0.299 + G \times 0.587 + B \times 0.114 \tag{3}$$

For FPGAs to operate with logic, it is necessary to avoid floating-point operations, and to convert to a shift algorithm, which is convenient for FPGA calculation. Therefore, the mental formula is scaled to the 8th power of 2 to calculate, and the formula is:

$$Gray = (R \times 76 + G \times 150 + B \times 30) >> 8 \tag{4}$$

4. System Hardware Implementation

The design receives one HDMI multiplexed video data through each FPGA through the FPGA, and converts it into 24- bit true color video data with 1080P (1920 x1080) and refresh rate of 60Hz through HDMI receiving chip[12]. The FPGA receives the video stream data and controls the two memory memories through the ADER model to implement ping-pong storage and reading of the video data. When performing pixel reading, the image of 1920 x1080 is split into nine

28×28 images by drawing points, and then the RGB image is converted into an 8-bit grayscale image to adapt the brain inspired processing chip. The data is input, and then the data is transmitted and received through the high-speed transmission module, and finally sent to the brain inspired processing chip. The overall system design is shown in Fig. 4.

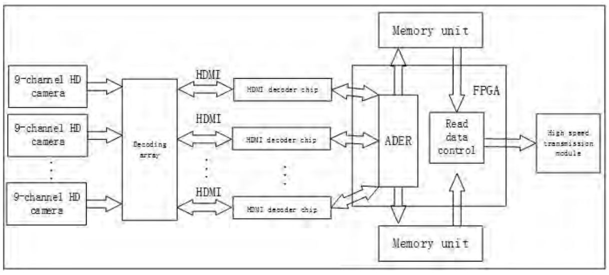


Fig. 4 Overall design block diagram

The entire system includes multiple HD cameras, a decoding array, an FPGA processor, two memory units and a high- speed transmission module. The decoding array captures the HD camera data in real time and converts it to an HDMI transmission signal. The HDMI decoder chip decodes the

HDMI stream in real time, converts it to video stream data, and sends the data to the FPGA. The FPGA includes an ADER- driven video stream control module, a memory control module, and a high-speed transmission control module.

For the transmission module, communication speed has be- come its biggest bottleneck in the field of traditional commu- nication. Especially in high-speed communication, traditional protocols such as RS232 are difficult to meet the requirements of current communication speed[13]. At present, the high- speed communication field is faster in optical fiber communi- cation, and a GBIC communication technology is developed. GBIC (Gigabit Interface Converter) is an interface device that converts gigabit electrical signals into optical signals[14]. SFP (Small Form-factor Pluggable) can be easily understood as an upgraded version of GBIC. It is small in size and supports hot swapping, and the corresponding manufacturers also provide simple module operation and parameter monitoring means, so SFP optical modules become high-speed communication. The main choice of field researchers, the latest optical module rate has reached 10 Gbit / s and above, and is moving towards low noise, low bit error rate, long-distance transmission[15]. So this article uses SFP as data transmission.

For the data frame transmission of SFP, in order to adapt the transmission and reception of high-speed video stream data as much as possible, this paper customizes a frame transmission protocol inside the FPGA and builds a data frame protocol control module. When there is data for transmission, the data frame protocol control module will formulate the address rules of the port and the destination port, and then detect the size of the transmitted data. The data frame protocol module will group the packets according to the size of the transmitted data, and the contents of the package include 64. The bit width of the frame header, the 32-bit source address and the 32-bit destination address, the packet size, the data, and the frame data check bits are then sent to the data transmission module, as shown in Table 1.

Table 1 Frame transmission data format

Flag	Addr	Length	Data	Crc
0x1A2B3C4 D5E6F7A8B	0xFFFFFFF FFFFFFFFF	98	64bit×98	CRC64

The source address and destination address here are used when a one-to-many transmission requires a switch. This article uses 1-to-1 transmission, so both the source address and the destination address are set to 0xFFFFFFFF.

When the FPGA needs to receive, the data frame protocol module detects the frame header, and then detects the source address and the destination address. When the destination address matches the local address, the length of the data is detected, and the data is received, and the data is unpacked. Read and save to the cache.

5. Experimental Data And Results

In this paper, based on the MNIST handwritten digits, the data is produced and placed in front of the camera, and the HDMI data output from the decoding array is obtained as shown in Fig. 5.

The image of the nine-channel camera input through HDMI is as shown in Fig. 6, and Fig. 7. The image is cut, and the image data received by the SFP is processed by matlab to obtain a 28 × 28 image.

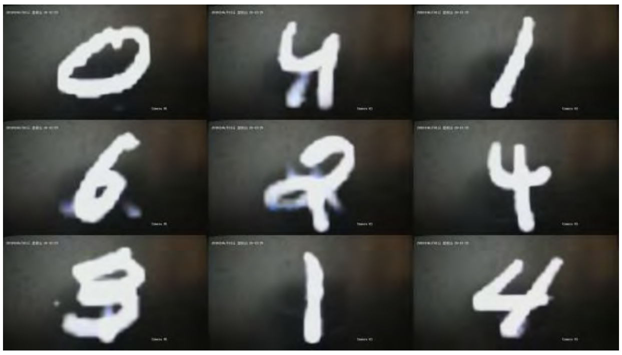
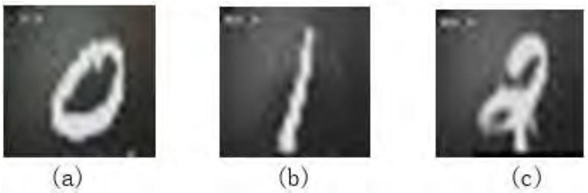


Fig. 5 Input data for the camera



(a) (b) (c)

Fig. 6 Image received after image processing

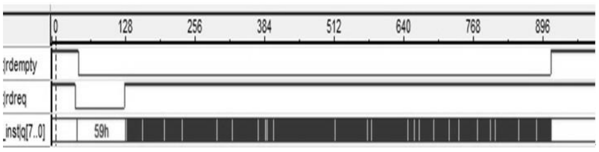
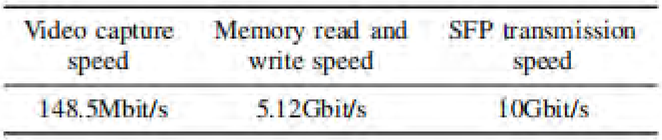


Fig. 7 Timing chart sampled by actual hardware using Signal Tap

Through analysis, the data obtained is not much different from the results obtained by the CPU processing data, and the speed of the system in storage and transmission is shown in Table 2.

Table 2 Transfer speed



Video capture speed	Memory read and write speed	SFP transmission speed
148.5Mbit/s	5.12Gbit/s	10Gbit/s

According to the chart, the data transfer speed is faster than the memory read and writes speed, and the memory read and write speed is faster than the video capture speed. Moreover, the speed bottleneck of the system is read and written in the memory, and when the memory read/write channel is fully loaded, the system can simultaneously process and transmit about 40 1080p HD video data at the same time. For more channels of video, the system can arbitrate the video stream in an arbitration manner, and can store only the more important images in part of the image area according to requirements.

Using python programming, using CPU processing and image segmentation, it takes about 0.12s to process one frame of image and segmentation, and the data speed of one frame of HD 1080p@60Hz image is 0.017s, so real-time image segmentation processing cannot be realized by cpu. However, the system processes only one frame of image and divides it by about 0.1 ms, and the processing speed can fully satisfy the real-time image processing.

As shown in Fig. 8, the antenna array is configured by the client, and the image data obtained above is input into the antenna chip through the SFP[16], and the calculation result of the antenna array is obtained.

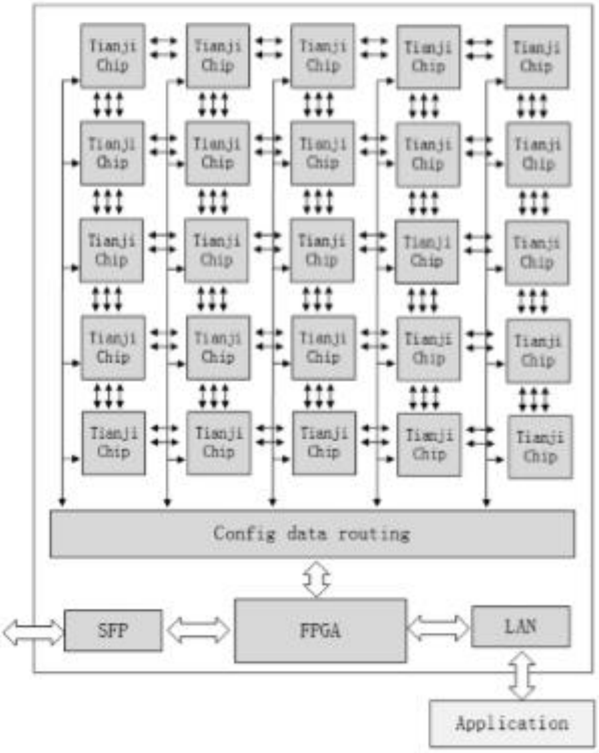


Fig. 8 TianjiC array structure

By analyzing the calculation results, the celestial chip can recognize and output the correct result well, and the calculation result is consistent with the calculation result sent to the celestial chip after cpu processing.

6. Conclusion

Based on the AER model, this paper proposes a new ADER model and uses it for real-time acquisition, storage, parallel processing and transmission and reception of multi-channel video data. Taking FPGA as the control core, SFP hot-swap small package module high-speed optical module is used as the transmission carrier, and the interface circuit design is carried out. Through the advantages of high-speed parallel processing of FPGA, the data of 9-channel camera will be collected with handwritten data as an example. The initial reduction is carried out to obtain a 28 x28 grayscale image, which is transmitted and received through SFP, and it is verified that the scheme can perform real-time video image processing and transmission. The whole system is highly portable and driven by ADER event. The model can be used not only for image processing, but also for multi-modal fusion, etc. It provides a high-speed solution for pre-processing image data of deep learning algorithms for brain inspired computing chips, and has a broad application prospect.

References

[1]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in International Conference on Neural Information Processing Systems, 2012.

[2]. F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Naka mura, P. Datta, and G. J. Nam, “Truenorth: Design and tool flow of a 65 mw 1 million neuron pr ogrammable neurosynaptic chip,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 10, pp. 1537–1557, 2015.

[3]. M. Davies, N. Srinivasa, T. H. Lin, G. Chinya, P. Joshi, A. Lines, A. Wild, and W. Hong, “Loihi: A neuromorphic manycore processor with on-chip learning,” IEEE Micro, vol. 38, no. 1, pp. 82 –99, 2018.

[4]. S. Liu, Z. Du, J. Tao, H. Dong, L. Tao, X. Yuan, Y. Chen, and T. Chen, “Cambricon: An instruction set architecture for neural networks,” in Acm/ieee International Symposium on Computer Architecture, 2016.

[5]. E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, and S. B. Furber, “Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation,” IEEE Journal of Solid-State Circuits, vol. 48, no. 8, pp. 1943–1953, 2013.

[6]. L. Shi, P. Jing, D. Ning, W. Dong, D. Lei, W. Yu, Y. Zhang, C. Feng, M. Zhao, and S. Song, “Development of a neuromorphic computing system,” in Electron Devices Meeting, 2016.

[7]. K. A. Boahen, “Point-to-point connectivity between neuromorphic chips using address events,” IEEE Transactions on Circuits & Systems II Analog & Digital Signal Processing, vol. 47, no. 5, pp. 416–434, 2000.

[8]. S. Liu, C. Lyu, Y. Liu, W. Zhou, X. Jiang, P. Li, H. Chen, and Y. Li, “Real-time implementation of harris corner detection system based on fpga,” IEEE International Conference on Real-time Computing and Robotics (RCAR), pp. 339–343, 2017.

[9]. S. Liu, C. Lyu, Y. Liu, W. Zhou, J. Xin, L. Peng, H. Chen, and Y. Li, “Real-time implementation of harris corner detection system based on fpga,” in IEEE International Conference on Real-time Computing & Robotics, 2018.

[10]. D. G. Chen, A. Bermak, and Y. T. Chi, “A low-complexity image compression algorithm for address-event representation (aer) pwm image sensors,” in IEEE International Symposium on Circuits & Systems, 2011.

[11]. T. M. Lehmann, C. Gonner, and K. Spitzer, “Survey: interpolation meth- ods in medical image processing,” Medical Imaging IEEE Transactions on, vol. 18, no. 11, pp. 1049–1075, 1999.

[12]. C. M. Maunder, R. E. Tulloss, D. K. Bhavsar, V. Gerousis, G. L. Giles, C. L. Hudson, C. Jensen, D. V. D. Lagemaat, P. F. Mchugh, and J. Maierhofer, “Ieee standard test access port and boundary-scan architecture,” IEEE Transactions on Medical Imaging, 1989.

[13]. Y. F. Zhu and X. E. Ye, “Design of data transmission platform based on lvds and fpga technology,” Advanced Materials Research, vol. 403-408, pp. 2111–2114, 2012.

[14]. L. Shen and S. X. Zheng, “Comment of receiving chip hdmi 1.3 in digital television,” Video Engineering, 2007.

[15]. N. Parkin, M. Bartur, D. Nesset, and D. Jenkins, “Gigabit sfp transceiver with integrated optical time domain reflectometer for ethernet access services,” in European Conference & Exhibition on Optical Communication, 2013.

[16]. P. Chi, S. Li, C. Xu, T. Zhang, and Y. Xie, “Prime: A novel processing- in-memory architecture for neural network computation in reram-based main memory,” in Acm/ieee International Symposium on Computer Architecture, pp. 27–39, 2016.