

## Smart-Eyes: a FPGA-based smart camera platform with efficient multi-port memory controller

Kunzhi Xie, Fei Qiao(✉), Wei Qi, Huazhong Yang

Department of Electronic Engineering

Tsinghua University

Beijing, China

xiekunzhi1987@163.com      qiaofei@tsinghua.edu.cn

Li Luo

School of Electronic and Information Engineering

Beijing Jiaotong University

Beijing, China

**Abstract**—In this paper, a FPGA-based smart camera platform is proposed. A Xilinx FPGA has been adopted as the core device. Xilinx ISE Design Suite has been used as the design tool and Verilog as the program language. In order to validate the platform, two real-time demonstrations have been implemented. In order to achieve a better tradeoff between performance and power consumption, a novel Multi-Port Memory Controller (MPMC) has been proposed. With a new memory mapping method and prefetching mechanism, the proposed MPMC outperforms Xilinx's MPMC (at least 25% optimization) in terms of data access. Additionally, the utilization of hardware resources of the FPGA is less than 5%. Thus the proposed Smart-Eyes can be further adopted in many reasonable real-time algorithms and applications.

**Keywords**-Smart Camera; FPGA; Multi-Port Memory Controller; memory mapping; prefetching mechanism

### I. INTRODUCTION

In recent decades, the smart camera has become a hot topic in both academia and industry, which not only strengthened the traditional camera applications, but also motivated new ones [1]. There are two significant motivations for this trend. The first is the extensive market demand. A new type of device is needed to meet the complex and challenging demands of the applications from the various domains such as video surveillance, traffic control, health care and environmental monitoring [2][3]. The second is the development of technology, including CMOS technology, embedded system and computer vision algorithm, especially the research on modeling of biological visual attention mechanism [4][5] and the semantic abstracting from the background environment [6].

Unlike the traditional camera, the smart camera can free staffs from heavy monitoring and searching work and finish some tasks automatically like gesture recognition [7], face recognition [8] and people tracking [9]. However, as an independent embedded visual system, the smart camera is mainly used at the frontend of video sensor networks, which means that the computing efficiency, storage capacity, transmission ability and energy resources are all

limited. How to design a smart camera platform with high performance, low power consumption and low cost with limited resources has become an issue of common concern.

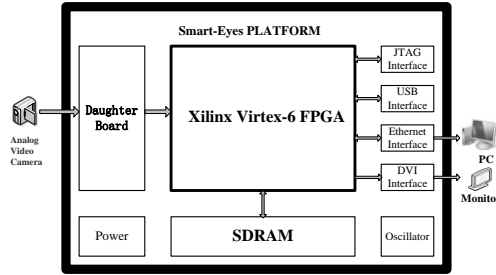
Cyclops [10], a camera module attached to MICA2 Mote, strives to reduce power consumption to allow large scale deployment and extend lifetime at the cost of low resolution. Moreover, its capacity of local processing is limited by the computing power. CMUcam3 platform [11] aims at providing open and flexible embedded visual platform with low power consumption. However, the lack of RAM capacity and computational capability makes it difficult to deal with many complex visual algorithms and real-time applications. MeshEye [12] effectively reduced the power consumption through highly integrated and multi-tier visual sensor mechanism. However, it transmits data through low-throughput SPI, which actually becomes the bottleneck of the platform.

To deal with the disadvantages of previous work, in this paper, we propose a FPGA-based smart camera platform named Smart-Eyes, with high resolution, high performance and low power consumption. Smart-Eyes can capture and process high resolution (720 x 576) video stream at the frame rate of 25fps (PAL). At the same time, it achieves low power consumption through a carefully designed power management module. Smart-Eyes can realize the wired and wireless communications through the USB interface and Ethernet interface respectively. In addition, we also provide two functional demonstrations on real-time video process and display.

The remainder of this paper is organized as follows. Section II outlines the system architecture of Smart-Eyes and discusses the reasons for choosing FPGA-based architecture. Section III describes our implementation of Smart-Eyes in detail, especially the key modules. Section IV presents example applications that illustrate our platform's capability and gives some results. Finally, Section V summarizes the proposed platform.

## II. SYSTEM ARCHITECTURE OF SMART-EYES

To select a suitable architecture is the first step of the entire design. Since the concept of the smart camera was put forward, many platforms with different architectures have been developed. In most cases, the system controller and processor is the core of digital system architecture.



**Fig. 1.** The architecture of smart camera platform

Micro Controller Unit (MCU) has been widely used in the research of the smart camera [10] [13]. However, with strict real-time constraint, it is obvious that software implementations on MCU are much slower than hardware implementations on FPGA [14]. Digital Signal Processor (DSP) is another solution. However, DSP has poor performance as

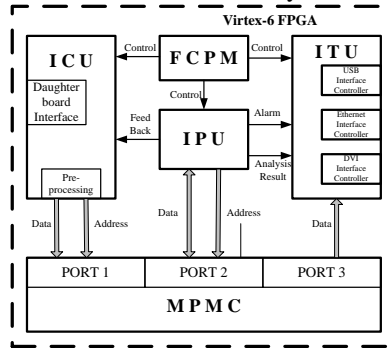
an embedded controller [15]. It is rational to combine MCU and DSP architecture [16]. But this plan means an increase of cost and power consumption.

FPGA, which has been used as the glue logic in the past, now plays an important role in the embedded system. Moreover, FPGA has been proved to be an indispensable step to ultimately implement power efficient ASIC. A lot of smart camera platforms based on FPGA have been developed [17][18]. The reasons why we choose FPGA-based architecture are as follows:

- Tradeoff between performance and power consumption: In recent years, FPGA has more logic elements per device, higher clock frequency and lower power consumption. Obviously, it will achieve better tradeoff between performance and power consumption [19].
- Flexibility and Scalability: Supported by partial and dynamic reconfigurable technology, FPGA can meet the demands of firmware updates and revisions of multimedia standards, while DSP may encounter difficulties in these aspects [20].
- Cost and Time: As is known to all, designers can avoid risks in development, reduce the costs and shorten the time-to-market by using FPGA-based rapid prototyping method to quickly validate the design.

The architecture of our smart camera platform is shown in Fig. 1. We choose Xilinx Virtex-6 FPGA as the core component. The main part of our system is implemented in the FPGA chip as hardware logic. The Power Module offers power supply, while the oscillator provides reference clock. Interface resources include JTAG, USB, Ethernet and DVI. SDRAM guarantees enough external memory space. Daughter-board completes ADC.

In conclusion, we believe that the proposed Smart-Eyes can effectively guarantee the real-time and high performance, and can also take the adaptability and scalability into consideration. The next section will introduce Smart-Eyes in detail.



**Fig. 2.** The block diagram of Smart-Eyes

### III. IMPLEMENTATION OF KEY MODULES

Our Smart-Eyes can obtain the video streaming from the camera and deliver it to the external memory. The communication can be achieved either in a wired manner or a wireless one. The block diagram of Smart-Eyes is shown in Fig. 2. The design mainly consists of six function modules. Image Capture Unit (ICU) serves as a video acquisition module, which buffers the data into External Memory (EM) through port1 of Multi-Port Memory Controller

(MPMC) after some pre-processing. Image Processing Unit (IPU) integrates image processing sub-modules to accomplish the task of specific applications. IPU not only gives feed-back signal to ICU, but also provides alarm and analysis result to Image Transmission Unit (ITU). ITU accepts video data, alarm and analysis result and drives the communication interfaces such as USB, Ethernet and DVI. As the control core, Function Configuration and Power Management (FCPM) is designed to achieve high performance and low power consumption. Next we describe each component respectively.

#### A. Image Capture Unit

As a video acquisition module, the ICU consists of a daughter-board interface controller and a sub-module for data pre-processing. The interface controller is designed to program the video decoder chip (TVP5154) on the daughter-board. The data pre-processing sub-module is designed to extract active video data from the output data of the video decoder chip. All these actions are under the control of FCPM module and adjusted by feed-back signal from IPU.

#### B. Multi-Port Memory Controller

In many mass data systems, the bandwidth of the memory access is often the bottleneck of performance. Therefore, a carefully designed Memory Controller is significant to the smart camera platform design.

Most existing FPGA-based smart camera platforms use Xilinx MPMC as their Memory Controller [20][21][22], which can divide each 2D transfer into 32-word transfers. In this way, the MPMC provided by Xilinx has advantages in video processing. However, there are still some drawbacks as follows.

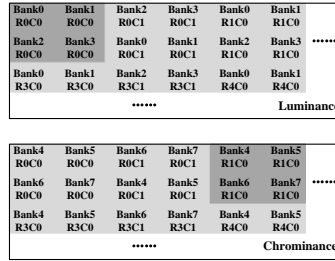


Fig. 3. The proposed memory mapping method

- The data width is 32-bit while the active video is 16-bit (YUV 4:2:2). Therefore, the valid pixel data only accounts for 50%.
- Xilinx MPMC requires that the X-Size, Start Address and Stride should be aligned to a 32-word boundary. This requirement may limit the choice of resolution.

In order to optimize memory access control, this paper designs an optimized MPMC. The optimized MPMC is a three-port memory controller. Compared with previous work, the proposed controller mainly includes two aspects of optimization.

##### 1) Memory Mapping

We use SDRAM as external memory. In video/image processing, it's common to store data based on macro block. In our work, the macro block size is 32 x 32.

Suppose we want to access some data in the macro block. When we meet bank miss status or row miss status, there will be a period of extra time called Row-Activation-Time.

Obviously, if we meet row hit status, which means the data are in the same row of the same bank, we can access the macro block more quickly [23]. Moreover, researchers have proposed a series of optimization method based on the structure of multi-bank [24].

According to the characteristics mentioned above, we follow two principles to reduce row switch delay of memory access as follows.

- Data of the same macro block should be stored in the same row of the same bank as far as possible.
- Data of adjacent macro block should be stored in different banks as far as possible.

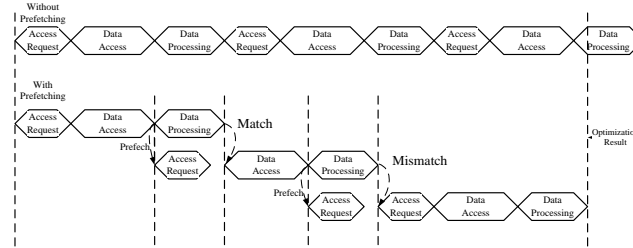
Based on the two principles above, we further put forward an efficient multi-port frame buffer mapping method. At the same time, we store luminance data and chrominance data separately so that we can access gray image conveniently, which brings smaller transmission burden and lower power consumption. The proposed memory mapping method is shown in Fig. 3. The luminance data and chrominance data of each pixel are individually assigned memory address.  $\{Bank(p), R(s), C(t)\}$  denotes the data will be stored in the address of Bank (p), Row (s), Column (t) .

## 2) Prefetching Mechanism

Prefetching technology has been one of the most important thoughts in data storage optimization. The researchers all over the world have put forward a lot of methods to prefetch data from external memory to the cache on-chip, including software way [25][26] and hardware way [27][28].

In this paper, a prefetching mechanism based on windows has been designed in the form of state machine, which can effectively improve memory bandwidth utilization with hardly any increase in costs. The timing of our prefetching mechanism is shown in Fig. 4. The proposed prefetching mechanism uses the idle time between normal memory accesses (the period when data is processed). When the state machine is at the state of prefetching, any access request will lead the system to save and halt the prefetching progress. Then there will be a matching judgment. If the access request matches the prefetching request, the halted prefetching progress goes on. Otherwise, the system just handles the new access request. The two cases are shown in Fig. 4. So the prefetching mechanism has two features as follows.

- Prefetching will not interfere with the original access.
- The cost of prefetching mismatch is small.



**Fig. 4.** Timing of prefetching mechanism

## C. Image Processing Unit

The IPU is open programmed for the designer and user. In this part, image processing algorithm is realized by Hardware Description Language (HDL), usually Verilog or VHDL.

Based on current platform, we can achieve different goals by designing and implementing the IPU in different ways. Firstly, the platform can be used as an image processing algorithm verification tool. Secondly, the platform can also be used as an algorithm analysis and optimization tool. Finally, we can develop and validate some specific video applications. In order to meet the demands above, IPU not only gives feed-back signal to ICU to control the capturing processing, but also provides alarm and analysis result to ITU to fulfill various applications.

#### D. Image Transmission Unit

Besides the raw video data, the Smart-Eyes can transmit information abstracted from the surrounding background or analysis results of observation object, such as the number of vehicles in a certain lane, facial features of people in a certain area and so on. Sometimes, it even just transmits an alarm. The Smart-Eyes has plentiful interface resources and interface controller IPs developed by ourselves. As a result, it can provide two manners of communication, wired and wireless.



**Fig. 5.** The original object and edge detection result

#### E. Function Configuration and Power Management

The core of the FCPM module is a Control State Machine. In order to achieve both high efficiency and low power consumption, this module monitors working condition of other modules and makes the corresponding response according to the observation results. In the future work, we plan to optimize FCPM using specific strategies in some specific application scenarios. Take the indoor surveillance as an example. We may introduce the idea of wake-up and multi-resolution to this module.

### IV. EXPERIMENTS AND RESULTS

In order to demonstrate the platform proposed, we developed two functional demonstrations. One is the gray image real-time display, the other is real-time edge detection.

The main purpose of gray image display is to validate the basic function of our platform. The raw data captured by ICU is written to SDRAM through port1 of MPMC. At the same time, ITU reads data from SDRAM through port3 of MPMC. MPMC allows user to access luminance data only, so it's easy to get gray image. After data processing, such as format conversion, synchronous and alignment, ITU transmits processed data to a DVI encoding chip. Then we use a monitor to display the encoded data in real time.

Edge detection is a basic algorithm of computer vision and image processing. It can identify the pixel that obviously differs from adjacent ones in luminance. Based on the gray image display mentioned above, we add edge detection using Sobel operator into IPU.

Luminance data read from SDRAM will be processed with edge detection algorithm before transmitted to ITU. Fig. 5 shows the original object and the edge detection result.

Next we give some important results, including hardware resource utilization and memory access optimization.

#### A. Hardware Resource Utilization

The device utilization rate is very critical. Since most components of our platform are implemented on FPGA and some image processing algorithms and specific applications may be implemented on the same FPGA chip, we have to optimize our code to minimize utilization rate of FPGA hardware resource, hoping to leave more resource for development of algorithms and applications. The total device utilization summary of the design is shown in Table 1. From the summary, it is evident that main resources consumed such as registers and LUTs are below five percent (5%). The left-over FPGA resources are sufficient for further implementations of intelligent video/image processing algorithms and applications.

**Table 1.** Device utilization summary

Resource Type	Utilized	Total	Utilization Rate
Slice Register	8648	301440	2%
Slice LUTs	6765	150700	4%
Block RAM/FIFO	25	1248	2%

#### B. Memory Access Optimization

The MPMC provided by Xilinx is so popular that it has been widely used in many FPGA-based smart camera platforms [20][21][22]. We have carried out a series of experiments about MPMC provided by Xilinx and MPMC we proposed. It shows that our optimization work is meaningful.

Firstly, the two MPMCs process the valid data with different efficiency. The data width of Xilinx MPMC is 32-bit and each datum denotes a pixel. The captured active video is in 16-bit according to the YUV 4:2:2 format. Therefore, in the video storage, the valid pixel data only accounts for 50%. In color image access (16-bit) and gray image access (8-bit), valid pixel data accounts for 50% and 25% respectively. The data width of our MPMC is 256-bit and each datum denotes multi pixels. Obviously, no matter storage or access, color or gray, the valid pixel data of our MPMC accounts for 100%, which means our MPMC has made full use of the data width without any waste.

Secondly, Table 2 shows the number of clock cycles when the same color (YUV) image and the same gray image are accessed (320x320) with the same clock period. As shown in Table 2, our memory mapping method has about 9% performance improvements in both color image and gray image access, compared with memory mapping method proposed in [29].

**Table 2.** Results of the color image and gray image access

Image Type	Method in [29](clock)	Proposed Method(clock)	Improvement Rate
YUV	21431	19505	9%
gray	10757	9705	9.78%

Thirdly, prefetching mechanism makes our MPMC faster in accessing data than MPMC provided by Xilinx. Since the prefetching mechanism is to use the period between two memory accesses, its performance depends on the frequency of memory access and clock cycle number required by window data access. We use time interval (number of clock cycles) between two memory access to represent access frequency. Fig. 6 shows the time (measured in clock cycle number) taken to access the color and gray image using two kind of MPMC respectively at different frequency of memory access. Fig. 6 shows the three points that we have obtained.

- It makes no difference in the total time when color image and gray image are accessed by the using of MPMC provided by Xilinx, while it is faster to access gray image than color image by the using of our MPMC.

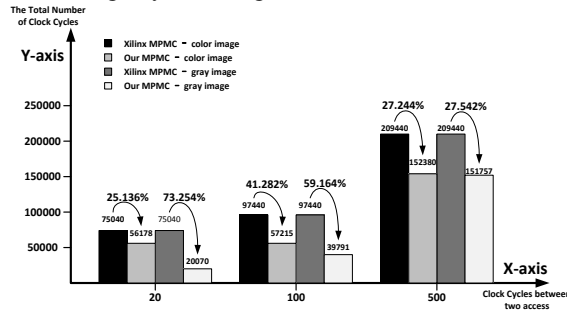


Fig. 6. The original object and edge detection result

- Our MPMC is faster (at least 25%) than Xilinx MPMC when color image and gray image are accessed.
- When the time interval between two memory accesses is approximately equal to the time required by window data access, our MPMC can get the best performance. When the time interval is shorter ( $X=20$  for example), only part of the data can be accessed, which will actually lower the performance. When the time interval is longer ( $X=500$  for example), the increasing idle time will not make any improvement of performance.

## V. CONCLUSIONS

In this paper, we have proposed a FPGA-based smart camera platform named Smart-Eyes. The platform utilizes a little device resources to achieve high resolution, high performance and low energy consumption. Smart-Eyes can capture and process high resolution (720 x 576) video stream at the frame rate of 25fps. Most components of Smart-Eyes are implemented on a Xilinx Virtex-6 FPGA. By designing and optimizing MPMC, we actually improve the performance of memory access phase. At the same time, we also adopt FCPM to achieve the goal of low power consumption. Smart-Eyes can meet the need of both academic research and commercial development. It can be used in the area of smart surveillance, intelligent traffic control and so on. An Optimized FCPM module and some more algorithms and applications will be our future work.



## REFERENCES

- [1] Belbachir, A. N. (2010). Smart cameras. Springer-Verlag US.
- [2] Hampapur, A., Brown, L., Connell, J., Pankanti, S., Senior, A., & Tian, Y. (2003, December). Smart surveillance: applications, technologies and implications. In *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on* (Vol. 2, pp. 1133-1138). IEEE.
- [3] Akyildiz, I. F., Melodia, T., & Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. *Computer networks*, 51(4), 921-960.
- [4] Borji, A., & Itti, L. (2013). State-of-the-art in visual attention modeling.
- [5] Frintrop, S., Rome, E., & Christensen, H. I. (2010). Computational visual attention systems and their cognitive foundations: A survey. *ACM Transactions on Applied Perception (TAP)*, 7(1), 6.
- [6] Zhang, C., Liu, Z., Zhang, Z., & Zhao, Q. (2008, October). Semantic saliency driven camera control for personal remote collaboration. In *multimedia signal processing, 2008 IEEE 10th workshop on* (pp. 28-33). IEEE.
- [7] Mitra, S., & Acharya, T. (2007). Gesture recognition: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(3), 311-324.
- [8] Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4), 399-458.
- [9] Hu, W., Tan, T., Wang, L., & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(3), 334-352.
- [10] Rahimi, M., Baer, R., Iroez, O. I., Garcia, J. C., Warrior, J., Estrin, D., & Srivastava, M. (2005, November). Cyclops: in situ image sensing and interpretation in wireless sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems* (pp. 192-204). ACM.
- [11] Rowe, A., Goode, A., Goel, D., & Nourbakhsh, I. (2007, May). CMUcam3: an open programmable embedded vision sensor. In *International Conferences on Intelligent Robots and Systems*.
- [12] Hengstler, S., Prashanth, D., Fong, S., & Aghajan, H. (2007, April). MeshEye: a hybrid-resolution smart camera mote for applications in distributed intelligent surveillance. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on* (pp. 360-369). IEEE.
- [13] Chen, P., Ahammad, P., Boyer, C., Huang, S. I., Lin, L., Lobaton, E., ... & Sastry, S. S. (2008, September). CITRIC: A low-bandwidth wireless camera network platform. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on* (pp. 1-10). IEEE.
- [14] Leiser, M., Miller, S., & Yu, H. (2004, April). Smart camera based on reconfigurable hardware enables diverse real-time applications. In *Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on* (pp. 147-155). IEEE.
- [15] Kleihorst, R., Reuvers, M., Krose, B., & Broers, H. (2004, October). A smart camera for face recognition. In *Image Processing, 2004. ICIP'04. 2004 International Conference on* (Vol. 5, pp. 2849-2852). IEEE.
- [16] O'Rourke, D., Jurdak, R., Liu, J., Moore, D., & Wark, T. (2009, October). On the feasibility of using servo-mechanisms in wireless multimedia sensor network deployments. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on* (pp. 826-833). IEEE.
- [17] Mosqueron, R., Dubois, J., & Paindavoine, M. (2007). High-speed smart camera with high resolution. *EURASIP Journal on Embedded Systems*, 2007(1), 23-23.
- [18] Bravo, I., Mazo, M., Lázaro, J. L., Gardel, A., Jiménez, P., & Pizarro, D. (2010). An intelligent architecture based on field programmable gate arrays designed to detect moving objects by using principal component analysis. *Sensors*, 10(10), 9232-9251.
- [19] Pellizzoni, R., & Caccamo, M. (2007). Real-time management of hardware and software tasks for FPGA-based embedded systems. *Computers, IEEE Transactions on*, 56(12), 1666-1680.
- [20] Desmoulières, C., Aslan, S., Oruklu, E., Sanie, J., & Vallina, F. M. (2010, May). HW/SW co-design platform for image and video processing applications on Virtex-5 FPGA using PICO. In *Electro/Information Technology (EIT), 2010 IEEE International Conference on* (pp. 1-6). IEEE.

- [21] Salem, M. M., Appel, M., Winkler, F., & Meffert, B. (2008, September). FPGA-based smart camera for 3D wavelet-based image segmentation. In *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on* (pp. 1-8). IEEE.
- [22] Salem, M. M., Klaus, K., Winkler, F., & Meffert, B. (2009, August). Resolution mosaic-based smart camera for video surveillance. In *Distributed Smart Cameras, 2009. ICDSC 2009. Third ACM/IEEE International Conference on* (pp. 1-7). IEEE.
- [23] Lee, K. B., Lin, T. C., & Jen, C. W. (2005). An efficient quality-aware memory controller for multimedia platform SoC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(5), 620-633.
- [24] Zhu, J., Hou, L., Wu, W., Wang, R., Huang, C., & Li, J. (2004, October). High performance synchronous DRAMs controller in H. 264 HDTV decoder. In *Solid-State and Integrated Circuits Technology, 2004. Proceedings. 7th International Conference on* (Vol. 3, pp. 1621-1624). IEEE.
- [25] Roth, A., & Sohi, G. S. (1999). Effective jump-pointer prefetching for linked data structures. In *Computer Architecture, 1999. Proceedings of the 26th International Symposium on* (pp. 111-121). IEEE.
- [26] Luk, C. K. (2001). Tolerating memory latency through software-controlled pre-execution in simultaneous multithreading processors. In *Computer Architecture, 2001. Proceedings. 28th Annual International Symposium on* (pp. 40-51). IEEE.
- [27] Nesbit, K. J., & Smith, J. E. (2004, February). Data cache prefetching using a global history buffer. In *Software, IEE Proceedings-* (pp. 96-96). IEEE.
- [28] Hur, I., & Lin, C. (2006, December). Memory prefetching using adaptive stream detection. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture* (pp. 397-408). IEEE Computer Society.
- [29] Zhang, P., Gao, W., Wu, D., & Xie, D. (2006, July). An efficient reference frame storage scheme for H. 264 HDTV decoder. In *Multimedia and Expo, 2006 IEEE International Conference on* (pp. 361-364). IEEE.