

## Real-time Multi-channel Vision Processing Based on DSP&FPGA

Yu-Bin Zhou<sup>1</sup>, Yu-Ning Yang<sup>2</sup>

<sup>1</sup>School of Mechanical Engineering, Shanghai University of Science and Technology, Shanghai, 201620, China

<sup>2</sup>Dept. of Computer Science VICM, University of Mannheim, Mannheim, Germany  
yuning.yang@ziti.uni-heidelberg.de, choybeen@yahoo.com

**Abstract**— In order to realize digital image sequence processing for multi-channel vision in real-time simultaneously, a hardware system with FPGA&DSP is designed. In the system, two ZBT SRAM chips are used as the input and output cache for high data transferring. A FPGA chip is responsible for the core logic controlling and multi-channel video synchronous. Digital videos are sent to the processing module by Camlink bus. Data are exchanged by EMIF and McBSP between FPGA and DSPs. EDMA is used for data transferring between SRAM in FPGA and ZBT SRAM. The QDMA is used for 2D data transferring to 1D into DSP cache. Tasks are assigned to chips by  $\mu$ C/OS on master DSP. All this together, real-time data sampling and processing for multi-channel vision was realized.

**Keywords**— FPGA; DSP; Stereo vision; Image processing.

### I. INTRODUCTION

In the field digital image sequence processing system, a series of procedure such as image mosaicking, image enhancement, target tracking and recognition, etc, need to be processed simultaneously in real-time. As the high-definition and multi-channel requirement in the visual system development over the past years, even high-performance computer can not handle the task in real-time. According to the highly parallel in data processing for image processing, solution with high-speed FPGA and DSP is preferred to overcome the capability constraints in PC computing. There are four main ways at present for real-time image processing: 1) based on a common PC [1-2]; 2) based on general-purpose DSP chips [3]; 3) based on one or more dedicated DSP chips; 4) based on programmable FPGA or DSP+FPGA [4]. The last solution based on general-purpose DSP chip for logic controlling along with the FPGA for highly parallel data processing is very appropriate for real-time processing of high-speed visual application. The advantages are not only taking the advantage of the high-speed DSP for processing control, while taking advantage of FPGA in a highly parallel data processing[5-6].

Data rate in digital image sequence processing systems, such as optical tracking, increases significantly as a result of using multiple cameras with higher resolution and higher frame rate. Suppose there are 3 cameras in a typical optical tracking system, each of which is running at 200 fps with the resolution of 1 Mpixel (1024x1024). These cameras will yield a total system data rate of 600 Mpixel/s. Simple 8-bit monochrome systems feature a data rate of 600 MBytes/s, and color systems, which typically use 24 bits to 48 bits per pixel, have a data rate that quickly

reaches multi-gigabyte-per-second range. Because most machine vision algorithms are computationally expensive, even the up-to-date high-performance PC can not handle such an image data rate in real time. Special hardware resources are needed to overcome the limitations of host CPU.

A real-time digital image processing system is presented in this paper. The system includes 2 FPGAs and 2 high-performance DSPs: C6416, 2 ZBT SRAM chips are data input and output cache. The system works on high-speed data communication between DSPs and FPGA through the EMIF and the McBSP. The EDMA is used for data transfer among chips to meet the multi-channel visual system for real-time computing requirements.

The paper will be structured as follows: The main hardware architecture will be introduced in section 2; Section 3 is about multi-channel video grabber including data sampling and controlling logic of high speed camera by FPGA. We will describe the data communication between DSPs and FPGA in section 4; Section 5 introduces the task management by embedded OS; And the acknowledgment in last section.

### II. HARDWARE SYSTEM DESIGN

The hardware platform built for multi-channel vision processing applications makes use of the state-of-the-art CMOS image sensor, FPGA and DSP technologies. Fig. 1 illustrates a high-level diagram of the hardware system. The designed hardware system consists of three modules: A high speed CMOS camera, a Camera Link image grabber, and a dedicated FPGA + Dual-DSP processing system.

The system has two FPGA, FPGA1 in image grabber module is responsible for image data grabbing and communication control, FPGA2 in data processing module is for high-speed parallel data processing. Two DSPs process the 2D image data from all the three cameras simultaneously.

The logic control chip FPGA1 is XILINX's XC3S1000L-FG456, used as the grab logic controller for video data. It is from Spartan3 low-power series with a 1M logic gates, 24 18x18 multipliers, and a wealth of on-chip storage space. Parallel computing FPGA2 is the member of XILINX's Virtex-4 series designed for parallel data processing: XC4VFX60-FF1152, within 4M on-chip RAM, 56,880 logic gate and 128 logical XtremeDSP modules. It is powerful enough for large number of high-speed data processing. The especial on-chip 18-KRAM module works in 500MHz, supporting the real dual-port simultaneous

read and write operations. The 2 co-processing DSPs C6416, each has eight parallel processing units, working on 600Hz, can finish up to 4.8 thousand trillion instructions / second (MFLOPS). The system also provides rich interface resources, including a 4-lane PCI Express, Gigabit Ethernet, USB2.0, and generous purpose user IO as well as LVDS link for board communication.

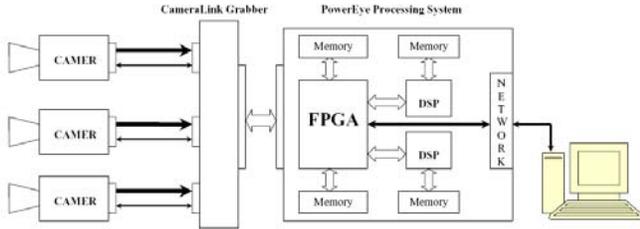


Figure 1. Hardware block diagram.

### III. MULTI-CHANNEL VIDEO GRABBER

As it is necessary for the system to detect fast movements in some applications, image grab rate should high enough to record the history of fast moving objects. Cameras play an essential role in an optical tracking system. With the dramatic improvement of image sensor technology, more and more high-speed image sensors are available. This provides the possibility to build a high-speed camera, which is able to capture fast moving objects.

#### A. Image Grab Module

The MT9M413 image sensor is size of 1280×1024 (1.3Mega pixel) CMOS digital sensor that is capable of 500 frames-per-second operation. This image sensor is available in both monochrome and color mode which has on-chip 10-bit analog-to-digital converters (ADCs). These ADCs require different input reference voltages for bias setting and calibration operation. Two DAC6573 digital-to-analog (DAC) converters are used to generate the reference voltages. The sensor board contains the MT9M413 image sensor is chosen to build our high speed camera, mounted on the top of the PCB and all required external circuitry, including DACs, decoupling capacitors and Samtec connectors on the bottom.

The realized high speed camera consists of three hardware parts:

- 1) The sensor module carries out the photo-electrical signal conversion sensor at a high pixel clock.
- 2) An FPGA device, is programmed to generate the control signals for the sensor module and the interface board, which represents the last hardware module in the camera system.
- 3) The interface board converts the digital image data to high speed LVDS signal pairs and receives control signals from the host. Camera Link standard is chosen as the camera interface.

Different circuit boards in the camera system are connected by high speed Samtec connectors to increase system flexibility. Camera Link is the communication interface for connecting the camera with the main

processing board for our high speed camera, because it supplies the highest data transfer bandwidth. The Camera Link Transmitter (CLinkTx) board implements the Camera Link FULL standard, It can also be configured as Camera Link Base or Camera Link Medium, due to the reprogrammability of the FPGA based control module. The main task of FPGA1 is used to track the video sync-signal, control data acquisition, and inform the FPGA2 to set the parameters such as camera frame rate, exposure time, the video window.

#### B. FPGA Control Module

The system logic controller chip is FG456, whose main task is to control the input/output frame buffer in order to inform the main processor chip to read out the image data from ZBT SRAM in time. It generates the all control signals required for the sensor, and synchronizes the output data stream from the sensor with the interface board. The on-chip programmable PLL generate the clock needed to drive the FPGA different clock. The internal counter could control the image grabbing size to XCFS04 FLASH by changing the carry signal of Adder. Fig. 2 shows the logic control block diagram.

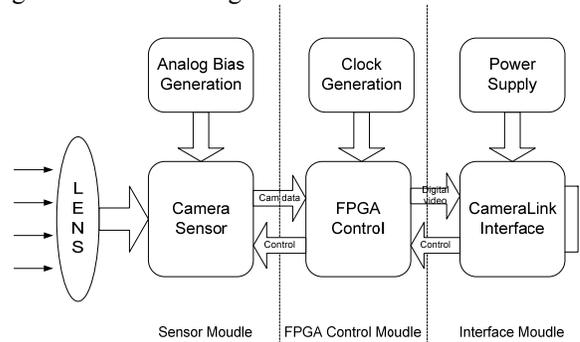


Figure 2. Controlling Logic by FPGA1

The FPGA1 receives the digital video from the sensor, packets the raw image data into Camera Link format, and then sends them to CLinkTx board via the high speed Samtec connector. FPGA1 also receives and executes control commands from the host. These commands include the configuration of the sensor, such as exposure time, gain, frame rate, region of interest (ROI), etc.

#### C. Ping-Pong data buffer in ZBT-RAM

The acquired digital image data are stored for frame buffer through two Zero-bus turn-around(ZBT) SRAM. There is no switching time between read and write cycle, ZBT SRAM can provide the greatest throughput to the system, thus increase maximum system bandwidth, since there is no bus latency (NoBL) during data buffering, not like the DDR which need to be managed through FIFO.[15-16]. A great challenge to implement a high speed ZBT SRAM controller is how to minimize the clock skew. Clock skew potentially reduces the overall design performance by increasing setup times and lengthening clock-to-output delays-both of which increase the clock cycle time. To ensure high performance, 3

Digital Clock Managers (DCMs) inside the FPGA1 are used in the ZBT SRAM controller as can be seen from Fig. 3: one to de-skew and generate a 2x controller clock and two to de-skew and generate a board-level 2x clock for the ZBT SRAM banks. The result is a high-speed, de-skewed clock driving the controller and the ZBT SRAM, which could satisfy all design requirements.

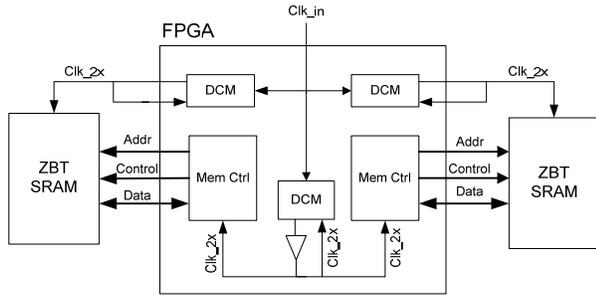


Figure 3. FPGA and ZBT SRAM

ZBT SRAM is well suited for image data buffering or Look-Up-Table applications that experience frequent bus turnarounds. Each ZBT SRAM can simultaneously receive full frame data from two channels. Frame rate of each channel is 60 fps, totally 120Mbytes per second. The operating frequency of ZBT SRAM is 250M, the maximum data throughput is high as 4.5GBytes/s, fully meet the speed requirement of the system data acquisition.

#### IV. DATA COMMUNICATION AMONG CHIPS

The most important aspect of multi-chip co-processing system is the inter-chip data communication efficiency. It directly impacts on the system overall performance. Variety methods for data transmission in the system increase the system efficiency. For this reason, great attention was given to this aspect early on in the hardware architecture design and the realized multi-path scheme shown in Fig. 4 will satisfy the most demanding applications.

##### A. Data communication between FPGA and DSPs

Depending on the chip interface and application characteristics of system demanding, there are two kinds data communication style connecting the FPGA and DSPs: 32bit EMIF-A and McBSP0. The 32bit EMIF-A could run at very high rate for data transmission between DSP and FPGA which takes full advantage of EDMA controller integrated within DSP.

The FPGA internal FIFO is used as data buffer for transferring to the receiving data from DSPs. The FPGA computing core began to get data from the FIFO to receive and process as soon as receiving data over a threshold amount. On the other hand, the calculation core sends the result to the transmit FIFO buffer if it has a threshold amount of free space. DSP will start the EDMA interrupt to transfer data to the DSP on-chip Cache as soon as the data transmit FIFO data reaches the threshold amount

without interrupting the CPU core. The peak data rate is up to 532 MBytes/s since the EMIF-A is working in a 32bit bus, 133 MHz.

The second method of communication between DSP and FPGA is by McBSP. Two DSPs' McBSP0 is connected to the FPGA. McBSP is a full-multiplexed serial port, working at 125Mbps. It has independent frame sync signal: FSX, FSR and bit sync clock CLKX, CLKR, and the system clock signal CLKS. In addition, an external CLKS input allows the transmitter and/or receiver to run from an externally provided clock. Data are communicated to the McBSP port via the data transmit (DX) pin for transmission and via the data for receive (DR) pin for reception. Data receiving and transmitting is also very simple in this way. For both the transmitter and receiver, the each clock or frame may be configured independently to be driven from an external source (slave) or from the McBSP's sample rate generator (master). This sample-rate generator can program both the width and active period of internally generated frame synchronization. Because of the simplicity of the McBSP transmission protocol, this communication interface costs minimum FPGA resources.

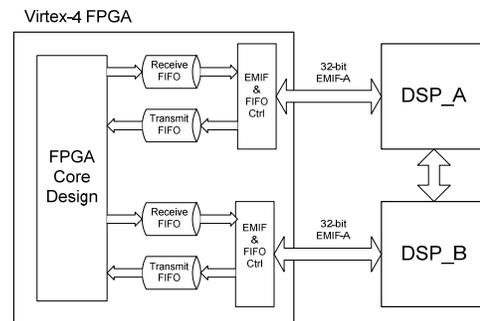


Figure 4. Inter-processor communication between FPGA and DSPs

##### B. Data communication between DSPs

There are two ways are designed for data communication between the two DSPs. In the first way, two DSPs are linked together through the two other McBSP interfaces. In order to obtain maximum data transfer rate, the two serial port of DSP: McBSP1 and McBSP2 are linked so that each DSP can be used as clock master and frame master. In other words, when one DSP's McBSP is used as a master in the communication interface generating data transmission clock, it can also generate the frame synchronization signals. In the same time, the other interface is waiting the control signals from the controller as a slave.

When the McBSP1 of DSP-A acts as the master to send data, the McBSP1 of DSP-B acts as the slave for receiving. At the same time, McBSP2 of DSP-B transmitter is configured to acts as the master to McBSP2 of DSP-A to generate the synchronization signal like clock and frame.

In addition to McBSP, the two DSPs can also be linked through EMIF-A for high-speed data exchange. This

interface is implemented by the FPGA2's internal 32bit bi-directional interface through the FIFO buffer. The programmable FIFO threshold interrupts support inter-DSP data movement via EDMA. This FIFO based link architecture support higher data transfer rate and lower latency for complex exchanges of bulk data and control messages between the DSPs. The data transfer between DSPs can be achieved as high as 512 MBytes/s when the DSP's EMIF-A working on 133 MHz.

### C. Data communication with PC

In order to control from PC or remote terminator, the system has access interface of Ethernet, USB2.0 and PCI-Express for communication purposes. Network access is provided by a 10/100/1000 Mbps Ethernet PHY, which is connected to the Virtex-4 FPGA via a standard Gigabit Multi Independent Interface (GMII). The PHY connects to the outside world with a standard RJ45 connector. General purpose I/O transfers are supported by way of the USB2.0 port. The 4 lane PCI-Express edge fingers are connected to the Multi-Gigabit Transceiver (MGT) blocks of the Virtex-4 FPGA, allowing the system to be used as a PCI-Express device.

## V. PARALLEL IMAGE PROCESSING

During object tracking, the huge amount of data throughput and very large volume of calculation need high performance efficiency for real-time data processing. Tracking by features is implemented in our system by the following steps: 1) The current frame is firstly divided into some small blocks, features are extracted from each block with an adaptive threshold. And features are extracted from blocks of the reference frame using the same way. 2) Predicted the inter-frame motion, and find the correspondent feature in a small searching window of reference frame. 3) MVs of those features are classified from background or from moving objects by their velocity and acceleration using K-Medoids cluster. Images are stabilized by the global MVs from the feature set being identified from the background. And the motion from objects is introduced by the rest feature points. The small difference of motion among different channels implies the real motion direction in 3D world.

Since the object tracking application on the system requires the software had to support images of multi-channel been processed simultaneously, the processing procedures are also complicated, including tasks for pre- and post- processing except the feature tracking algorithm for optical tracking. The  $\mu$ C/OS is a kind of low-cost priority-based pre-emptive real time multitasking operating system kernel for microprocessors which could run on efficiently on C6416[7]. New version of  $\mu$ C/OS-II supporting C6416 could be downloaded from Micrium homepage. In our system, one of the two DSPs runs in master mode and assigns the tasks to FPGA2 and another DSP.

Object tracking involving lots of parallel data processing is assigned by  $\mu$ C/OS to FPGA2. The two DSPs could work simultaneously on different task phase of

tracking for different frames to implement the whole tracking algorithm as shown in Fig.5.

The experiment result shows the method could deal with multi-object moving in the scene along with stabilization on our system in 60fps. FPGA2 takes off much parallel data processing such as a number of the total features and clusters which make the processing speed fast enough for real-time application even though multi-channel videos are processed in the same time.

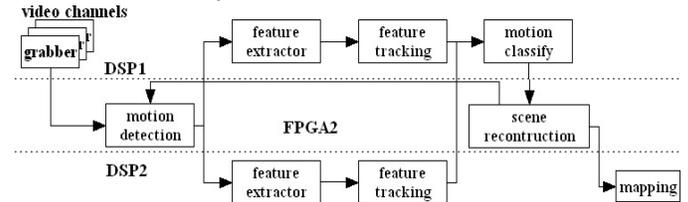


Figure 5. Parallel tasks processing

## VI. ACKNOWLEDGMENT

Thanks the Shanghai Education Committee for supporting the project by Science Fund of Excellent Yong Teacher (GJD08017: 2008-2010).

## REFERENCES

- [1] R. C. Chaubey. Computerized Image Analysis Software for the Comet Assay, Molecular Toxicology Protocols, ISSN 1064-3745,2008
- [2] JinSeon Youn, JunRim Choi, Seung-Soo Han, Parallel Integer Motion Estimation Method by Using Reference Blocks Shared for HD Video Encoding, International Conference on Electronic Computer Technology, ISBN: 978-0-7695-3559-3, IEEE Computer Soc, CA, USA, Feb.2009: p577-581
- [3] M. Fikret Ercan. Parallel Image Understanding on a Multi-DSP System. O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4706, Part II, pp. 1-C12, 2007. Springer-Verlag Berlin Heidelberg 2007
- [4] Shirvaikar Mukull ; Bushnaq, Tariq1 , A comparison between DSP and FPGA platforms for real-time imaging applications, Proceedings of SPIE-IS and T Electronic Imaging - Real-Time Image and Video Processing, 2009
- [5] Davis, L.S.: Foundations of Image Understanding. Springer, Heidelberg (2001)
- [6] Kumar, V.P., Wang, C.L.: Parallelism for Image Understanding. In: Zomaya, A.D. (ed.) Parallel and Distributed Computing Handbook, pp. 1042-C1070. McGraw-Hill, New York (1996)
- [7] LI Ji-kui, XIANG Huai-kun, HU Hong. Research on video image processing system based on uCOS - II. Journal of Harbin Institut of Technology. Vol.39, No.2, 2007