I.J. Intelligent Systems and Applications, **2013, 12, 113-119**
Published Online November 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2013.12.10

Performance Evaluation of Different Memory Components for FPGA based Embedded System Design for Video Processing Application

Sanjay Singh*, Ravi Saini, Anil K. Saini, AS Mandal, Chandra Shekhar
CSIR – Central Electronics Engineering Research Institute (CSIR-CEERI), Pilani – 333031, Rajasthan, India
**Corresponding Author E-mail: sanjaysingh@ceeri.ernet.in*

Anil Vohra
Electronic Science Department, Kurukshetra University, Kurukshetra, Haryana, India

Abstract— Advances in FPGA technology have dramatically increased the use of FPGAs for computer vision applications. Availability of on-chip processor (like PowerPC) made it possible to design embedded systems using FPGAs for video processing applications. The objective of this research is to evaluate the performance of different memory components available on FPGA boards for embedded/platform-based implementations of image/video processing applications. The clustering based change detection algorithm for Ubiquitous Multimedia Environment is selected for evaluating the effect of different memory components (DDR/BRAM) on performance of the system in terms of frame rate (frames per second).

Index Terms— Performance Evaluation, Image/Video Processing, Embedded Systems, FPGA Implementation

I. Introduction

The main challenge for computer vision/image processing applications lies in achieving the real-time performance. For this reason different technologies and design methodologies have been used to build computer vision systems. These technologies go from general purpose processors/special purpose digital signal processors to applications specific integrated circuits (ASICs)/application specific instruction set processor (ASIPs), or even field programmable gate arrays (FPGAs). Recent advances in FPGA technology have increased the use of FPGAs for computer vision applications. Current generation FPGA have their size and performance comparable to those of ASICs and provide the flexibility to perform algorithmic changes in later stages of system development. Moreover, their structure is able to exploit the spatial and temporal parallelism inherent in many image processing applications. These features have increased the interest of researchers toward FPGAs for implementing computer vision/image processing systems [1], [2]. The availability of on-chip embedded processors (like PowerPC and/or Microblaze) made it possible to design whole embedded system for video/image processing applications using hardware/software co-design based methodologies. Recently, many research papers have been presented in the literature related to implementation of video/image processing applications on FPGAs [3]-[7] (a thorough discussion of this body of the work is, however, beyond the scope of this paper).

For any software program running on embedded processor, the two main sections which require storage are program/instructions and intermediate data objects. In this research, we have studied the effect of storing the program/instructions and intermediate data objects in different memory components (Block RAM, DDR, or combination of both) on performance (frame rate) of image processing application. The clustering based change detection scheme [8] is chosen for this study for two reasons. First, the program/instruction memory size is large enough to study the effect of different memory components on program execution time. Second, this algorithm requires the storage of some important background related information for each block of pixels in image. Therefore, a large amount of memory is needed for storing intermediate data objects, which is important for studying the effect of different memory components selected for storage of intermediate data objects.

The rest of the paper is organized in the following way: Section 2 briefly describes the clustering based change detection scheme. In section 3, we show the implemented FPGA based embedded system architecture, and different memory components partitioning and their performance evaluation. Section 4 discusses results and evaluates the performance. Finally, we conclude the paper in section 5.

II. Change Detection Scheme

The clustering based scheme used for change detection for ubiquitous multimedia environment is briefly described in this section, for a more thorough description we refer to [8]. The main steps involved in the algorithm are given below:

1. Each incoming gray frame is partitioned into 4x4 blocks.
2. Each block is represented by a group of clusters and each cluster contains centroid value and the frame number which updated the cluster recently.
3. Initially, the cluster set of each block is initialized with a cluster having its centroid set to average color value of the corresponding block of the first frame.
4. Each block has 3 cluster nodes.
5. For every new frame, each block is compared with the corresponding cluster group. The difference is computed by taking Manhattan distance between average color value of the block and its centroid.
6. If the difference is below a threshold value, it is considered a matching cluster.
7. For a matching cluster, the frame number and centroid associated with the cluster node of the corresponding block is updated.
8. For a given block, if no matching cluster is found then a new cluster node is created by replacing the existing cluster node which has not been updated for the longest period of the time.

III. System Architecture

This section discuses three different embedded/platform-based implementations of clustering based change detection scheme on Virtex-II Pro FPGA board. The basic block diagram of the proposed embedded/platform-based implementation is shown in Fig. 1. There is one embedded processor block PPC405 for executing the software programs, three memory modules (DDR Memory, BRAM Memory, and Flash Memory), one TFT controller module connected to Display monitor, one JTAG controller connected to Host Computer via JTAG cable, one UART controller connected to Host Computer via RS232 cable, two buses (PLB and OPB) for providing interconnection between different modules, and one PLB2OPB Bridge for providing communication between two buses.

Fig. 2 shows the Xilinx Virtex-IIPro XUP FPGA board [9] used in our implementation. The different modules necessary for implementation of clustering based change detection scheme are Flash memory, DDR Memory, and FPGA device. Different modules for interfacing of this platform are VGA connector, JTAG connector, and UART port.

The embedded platform for executing clustering based change detection scheme is implemented using Xilinx Embedded Development Kit (EDK) tool chain. All the blocks are configured using this software. The system architecture produced by EDK is shown in Fig. 3.

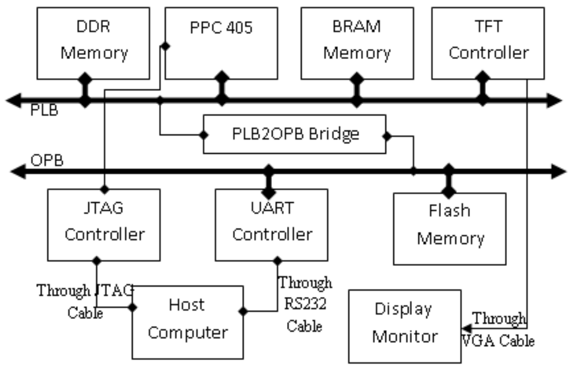


Fig. 1: Block Diagram of FPGA based Embedded/Platform-based system

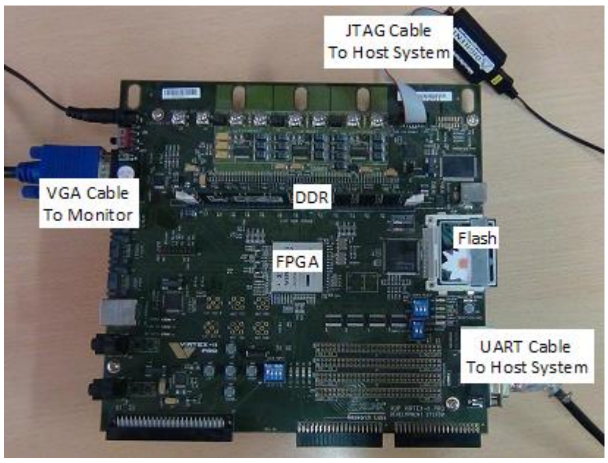


Fig. 2: Xilinx Virtex-IIPro FPGA Platform for Embedded/Platform-based implementation of Change Detection

The embedded processor *ppc405* is connected to processor local bus (*plb*) through two interfaces: data PLB (*DPLB*) and instruction PLB (*IPLB*). JTAG PowerPC controller (jtagppc_cntlr) provides the connection of PowerPC405 with host computer for programming and debugging. DDR and Block RAM memories are connected to PLB through *plb_ddr* and *plb_bram_if_controller* interfaces respectively. TFT display controller is also connected to PLB. Data Control Register (DCR) bus is used to configure the different registers of *plb_tft-cntlr_ref* module for proper functioning. System ACE Compact Flash and RS232 UART modules are connected to on-chip peripheral bus (*opb*). This bus is relatively slower as compared to PLB. The two buses PLB and OPB interact with each other through *plb2opb_bridge*. DCR and OPB bus interact through *opb2dcr_bridge*. In Fig. 3, M is used for Master and S is used for Slave. This embedded system architecture is used to evaluate the performance of different memory components for image/video processing applications using clustering based change detection scheme mentioned in section II.

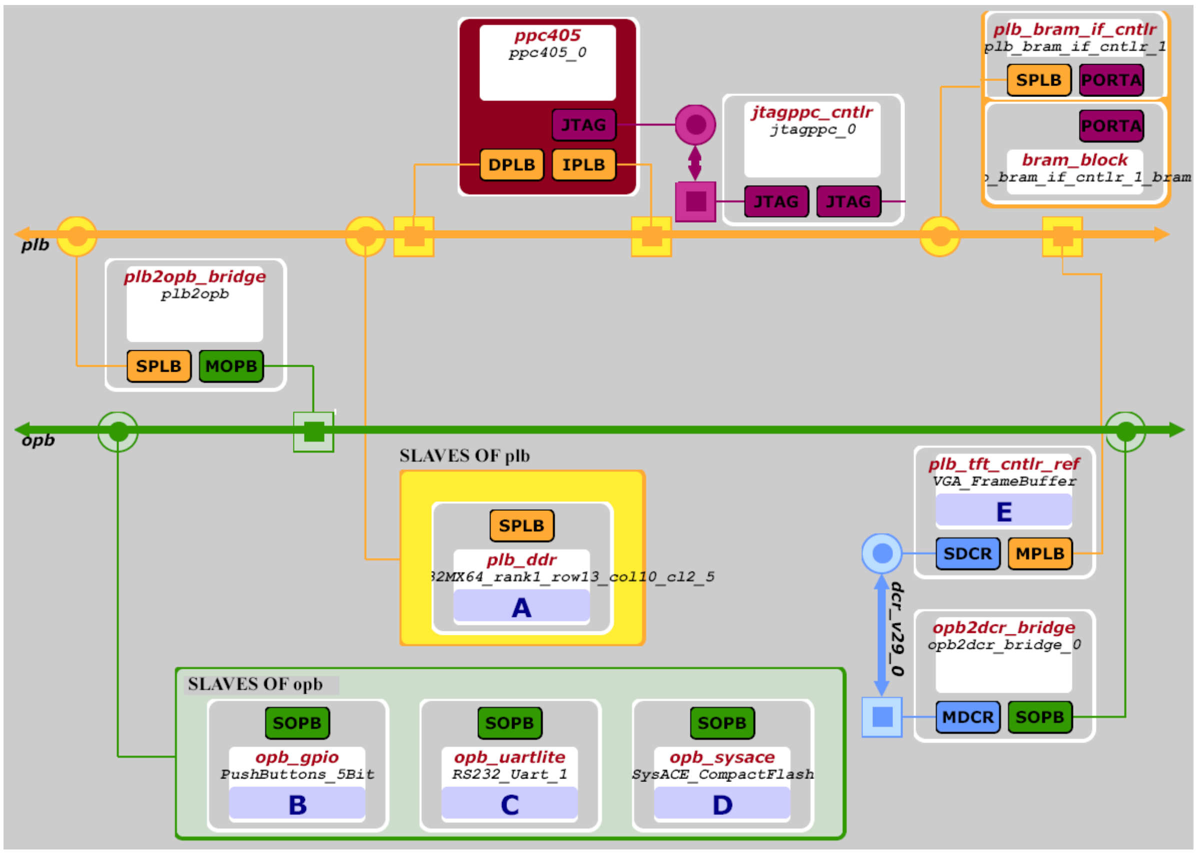


Fig. 3: Embedded/Platform-based system for Change Detection generated by EDK

For evaluating the performance of different memory components available on FPGA board, the following procedure is used.

1. Input video frames are stored in Flash memory in *.bmp image format.
2. C/C++ program running on PowerPC405 is used to read the image files from Flash memory and to store them into DDR memory.
3. The video frames stored in DDR memory are converted in gray scale and processed by change detection C/C++ code running on PowerPC405 and the results are stored back in DDR memory.
4. The processed video frames stored in DDR memory are displayed on Monitor by executing the C/C++ program on PowerPC405. This program provides the addresses of processed video frames to TFT controller.
5. The complete process is monitored using hyper terminal on host computer through RS232 UART port.
6. The executable program files from host system are downloaded on FPGA PowerPC405 using JTAG interface and Xilinx Microprocessor Debugger (xmd) tool.
7. The time taken by change detection C/C++ program running on FPGA PowerPC405 is measured and frame rate is computed.

Three implementations are considered and executed on PowerPC405 embedded processor for this evaluation. These are discussed below.

3.1 Implementation I

In this implementation, only video frames are stored in DDR Memory. The all program execution related sections (Boot Section, Program/Instructions, Intermediate Data Objects, Stack, and Heap) are stored in Block RAMs. The memory partitioning for this implementation is shown in Fig. 4. The snapshot of downloading of *executable.elf* file for this implementation is shown in Fig. 5. It is clear that all sections reside in Block RAMs (in address range of 0xfffe0000 to 0xffffffff).

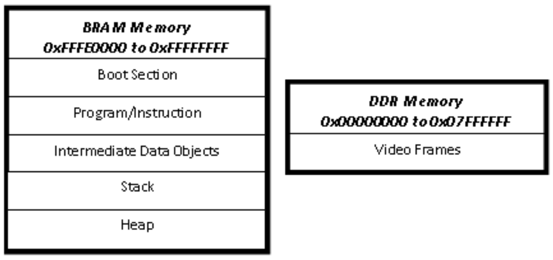
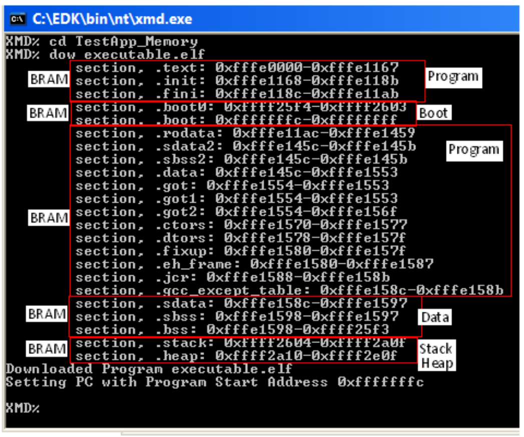


Fig. 4: Memory Partitioning for Implementation I



```
C:\EDK\bin\nt\xmd.exe
XMD% cd TestApp_Memory
XMD% dow executable.elf
        section, .text: 0xfffe0000-0xfffe1167
        section, .init: 0xfffe1168-0xfffe118b
        section, .fini: 0xfffe118c-0xfffe11ab
        section, .boot0: 0xffff25f4-0xffff2603
        section, .boot: 0xfffffffc-0xffffffff
        section, .rodata: 0xfffe11ac-0xfffe1459
        section, .sdata2: 0xfffe145c-0xfffe145b
        section, .sbss2: 0xfffe145c-0xfffe145b
        section, .data: 0xfffe145c-0xfffe1553
        section, .got: 0xfffe1554-0xfffe1553
        section, .got1: 0xfffe1554-0xfffe1553
        section, .got2: 0xfffe1554-0xfffe156f
        section, .ctors: 0xfffe1570-0xfffe1577
        section, .dtors: 0xfffe1578-0xfffe157f
        section, .fixup: 0xfffe1580-0xfffe157f
        section, .eh_frame: 0xfffe1580-0xfffe1587
        section, .jcr: 0xfffe1588-0xfffe158b
        section, .gcc_except_table: 0xfffe158c-0xfffe158b
        section, .sdata: 0xfffe158c-0xfffe1597
        section, .sbss: 0xfffe1598-0xfffe1597
        section, .bss: 0xfffe1598-0xffff25f3
        section, .stack: 0xffff2604-0xffff2a0f
        section, .heap: 0xffff2a10-0xffff2e0f
Downloaded Program executable.elf
Setting PC with Program Start Address 0xfffffffc

XMD%
```

Fig. 5: Different sections and their addresses for Implementation I

3.2 Implementation II

For this implementation, the intermediate data objects section is moved to DDR Memory. Therefore, DDR memory contains video frames and intermediate data objects. The all remaining program execution related sections (Boot Section, Program/Instructions, Stack, and Heap) are stored in Block RAMs. The memory partitioning for this implementation is shown in Fig. 6. The snapshot of downloading of *executable.elf* file for this implementation is shown in Fig. 7. It is clear that intermediate data objects section resides in DDR memory (in address range of 0x00000000 to 0x00011067) and all other remaining sections reside in Block RAMs (in address range of 0xfffe0000 to 0xffffffff).

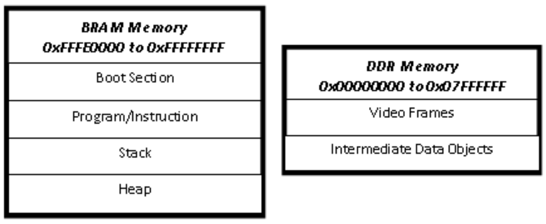
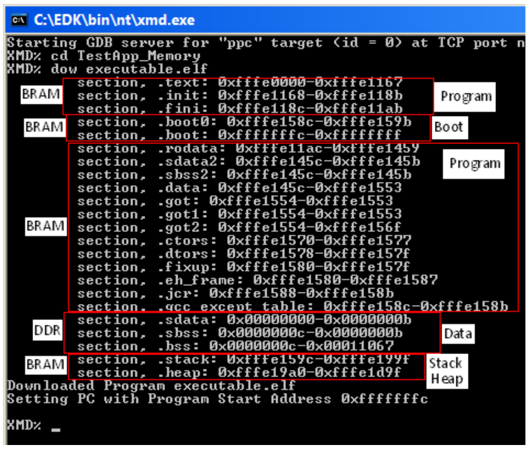


Fig. 6: Memory Partitioning for Implementation II

3.3 Implementation III

In this implementation, all the program execution related sections except Boot Section are moved to DDR memory. Therefore, DDR memory contains video frames and all program execution related sections (Program/Instructions, Intermediate Data Objects, Stack, and Heap). It is necessary to store the Boot section in Block RAMs for successful execution of the program. The memory partitioning for this implementation is shown in Fig. 8. The snapshot of downloading of *executable.elf* file for this implementation is shown in Fig. 9. It is clear that intermediate data objects section resides in DDR memory (in address range of 0x00000000 to 0x0005da1f) and all other remaining sections reside in Block RAMs (in address range of 0xfffe0000 to 0xffffffff).



```
C:\EDK\bin\nt\xmd.exe
Starting GDB server for "ppc" target (id = 0) at TCP port n
XMD% cd TestApp_Memory
XMD% dow executable.elf
        section, .text: 0xfffe0000-0xfffe1167
        section, .init: 0xfffe1168-0xfffe118b
        section, .fini: 0xfffe118c-0xfffe11ab
        section, .boot0: 0xfffe158c-0xfffe159b
        section, .boot: 0xfffffffc-0xffffffff
        section, .rodata: 0xfffe11ac-0xfffe1459
        section, .sdata2: 0xfffe145c-0xfffe145b
        section, .sbss2: 0xfffe145c-0xfffe145b
        section, .data: 0xfffe145c-0xfffe1553
        section, .got: 0xfffe1554-0xfffe1553
        section, .got1: 0xfffe1554-0xfffe1553
        section, .got2: 0xfffe1554-0xfffe156f
        section, .ctors: 0xfffe1570-0xfffe1577
        section, .dtors: 0xfffe1578-0xfffe157f
        section, .fixup: 0xfffe1580-0xfffe157f
        section, .eh_frame: 0xfffe1580-0xfffe1587
        section, .jcr: 0xfffe1588-0xfffe158b
        section, .gcc_except_table: 0xfffe158c-0xfffe158b
        section, .sdata: 0x00000000-0x0000000b
        section, .sbss: 0x0000000c-0x0000000b
        section, .bss: 0x0000000c-0x00011067
        section, .stack: 0xfffe159c-0xfffe199f
        section, .heap: 0xfffe19a0-0xfffe1d9f
Downloaded Program executable.elf
Setting PC with Program Start Address 0xfffffffc

XMD% _
```

Fig. 7: Different sections and their addresses for Implementation II

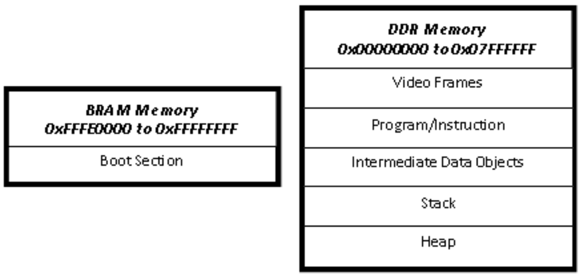
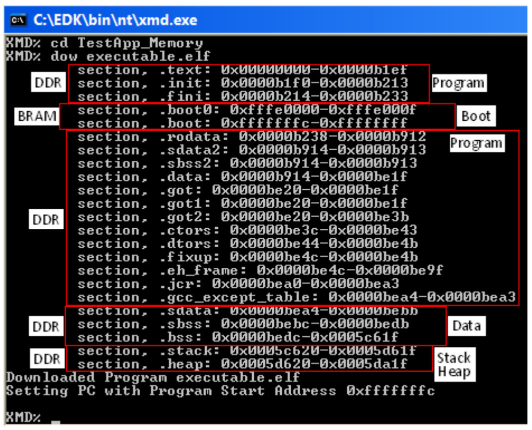


Fig. 8: Memory Partitioning for Implementation III



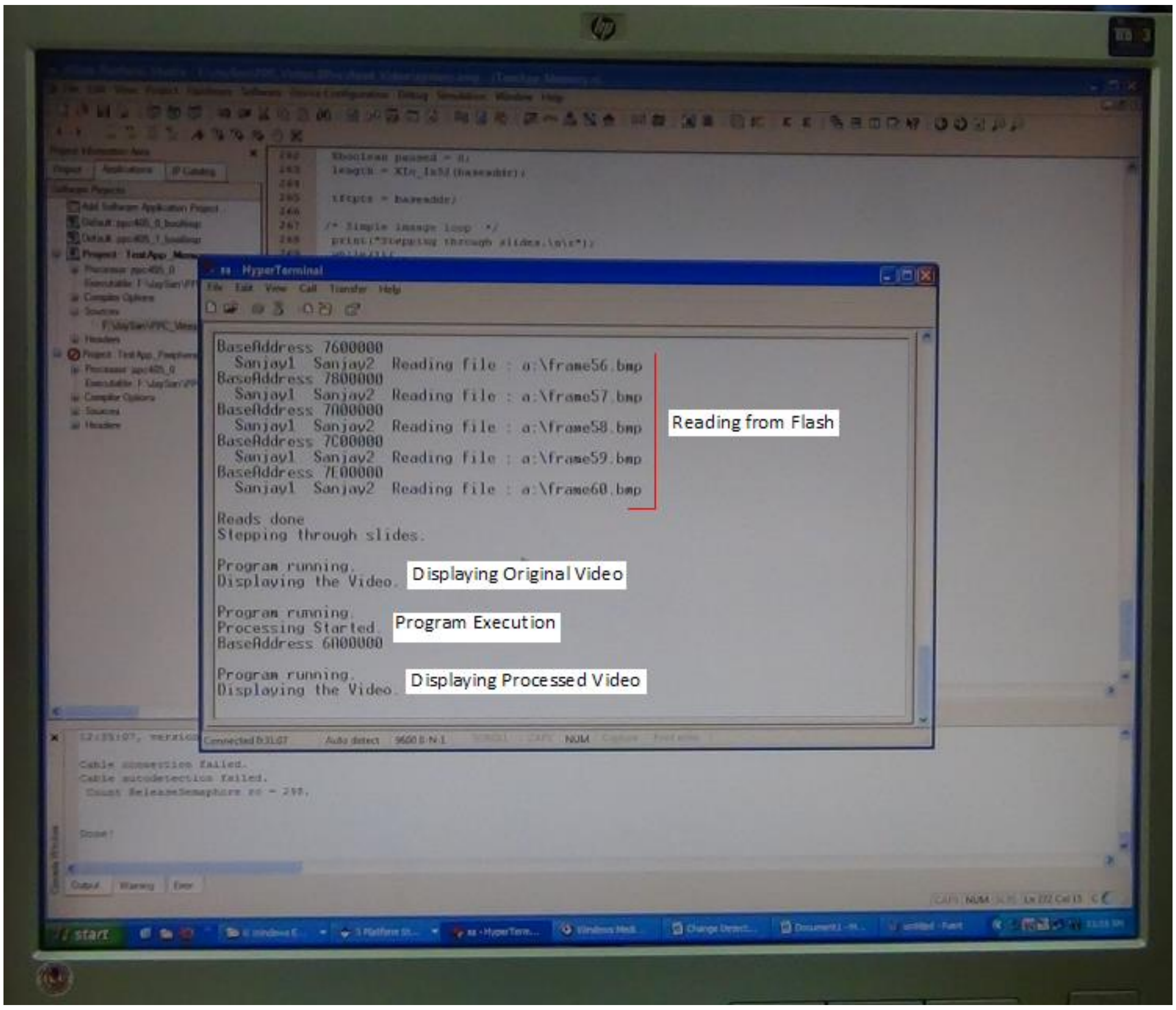
```
C:\EDK\bin\nt\xmd.exe
XMD% cd TestApp_Memory
XMD% dow executable.elf
        section, .text: 0x00000000-0x0000b1ef
        section, .init: 0x0000b1f0-0x0000b213
        section, .fini: 0x0000b214-0x0000b233
        section, .boot0: 0xfffe0000-0xfffe000f
        section, .boot: 0xfffffffc-0xffffffff
        section, .rodata: 0x0000b238-0x0000b912
        section, .sdata2: 0x0000b914-0x0000b913
        section, .sbss2: 0x0000b914-0x0000b913
        section, .data: 0x0000b914-0x0000be1f
        section, .got: 0x0000be20-0x0000be1f
        section, .got1: 0x0000be20-0x0000be1f
        section, .got2: 0x0000be20-0x0000be3b
        section, .ctors: 0x0000be3c-0x0000be43
        section, .dtors: 0x0000be44-0x0000be4b
        section, .fixup: 0x0000be4c-0x0000be4b
        section, .eh_frame: 0x0000be4c-0x0000be9f
        section, .jcr: 0x0000bea0-0x0000bea3
        section, .gcc_except_table: 0x0000bea4-0x0000bea3
        section, .sdata: 0x0000bea4-0x0000bebb
        section, .sbss: 0x0000bebc-0x0000bedb
        section, .bss: 0x0000bedc-0x0005c61f
        section, .stack: 0x0005c620-0x0005d61f
        section, .heap: 0x0005d620-0x0005da1f
Downloaded Program executable.elf
Setting PC with Program Start Address 0xfffffffc

XMD% _
```

Fig. 9: Different sections and their addresses for Implementation III

IV. Results

The embedded system/platform for change detectionalgorithm is designed using Xilinx Embedded Development Kit (EDK). For performance evaluation, the CIF (352x288) size gray video frames are taken. The video frames are stored in Flash memory. Storing the program/instruction as well as intermediate data objects in Block RAMs (implementation 1) processed 50 video frames in 10 seconds (frame rate of 5 frames per second). Storing the intermediate data objects of algorithm in DDR and program/instructions in Block RAMs (implementation 2) have resulted in a frame rate of 4.25 fps (processed 50 frames in 12 seconds). Storing of both intermediate data objects as well as program/instruction in DDR (implementation 3) has resulted in slowest processing. It takes 17 seconds for processing 50 frames (approximate frame rate of 3 frames per second). Fig. 10 shows the snapshot of host system monitor screen where the program execution is monitored on hyper terminal (connected to FPGA through UART port). It shows the successful reading process of video frames from Flash memory (and storing them into DDR memory), display of original video, program execution, and display of processed video.


```
BaseAddress 7600000
  Sanjay1  Sanjay2  Reading file : a:\frame56.bmp
BaseAddress 7800000
  Sanjay1  Sanjay2  Reading file : a:\frame57.bmp
BaseAddress 7A00000
  Sanjay1  Sanjay2  Reading file : a:\frame58.bmp
BaseAddress 7C00000
  Sanjay1  Sanjay2  Reading file : a:\frame59.bmp
BaseAddress 7E00000
  Sanjay1  Sanjay2  Reading file : a:\frame60.bmp

Reads done
Stepping through slides.

Program running.
Displaying the Video.

Program running.
Processing Started.
BaseAddress 6A00000

Program running.
Displaying the Video.
```

Fig. 10: Monitoring of Program Execution through Hyper Terminal

Fig. 11 shows the initial frame 1 when no object is present in the scene, intermediate frame 23 when cow started entering in the scene, and change detection results produced by all three implementation for frame number 23. Fig. 12 shows the initial frame 1 when no object is present in the scene, intermediate frame 38 when cow entered and is in middle of the scene, and change detection results produced by all three implementation for frame number 38.

V. Conclusion

This paper has presented the performance evaluation of different memory components available on Virtex-IIPro FPGA board. By using this study, a designer should be able to find an optimal memory system for designing embedded/platform-based system for given image/video processing application. It is found that storing program/instructions in Block RAMs and intermediate data objects in DDR makes the best tradeoffs between memory and performance for image/video processing. The frame rates could be improved to meet the real-time video processing requirements (25 to 30 frames per second) by moving some time consuming and computationally intensive section of algorithms in pure hardware.

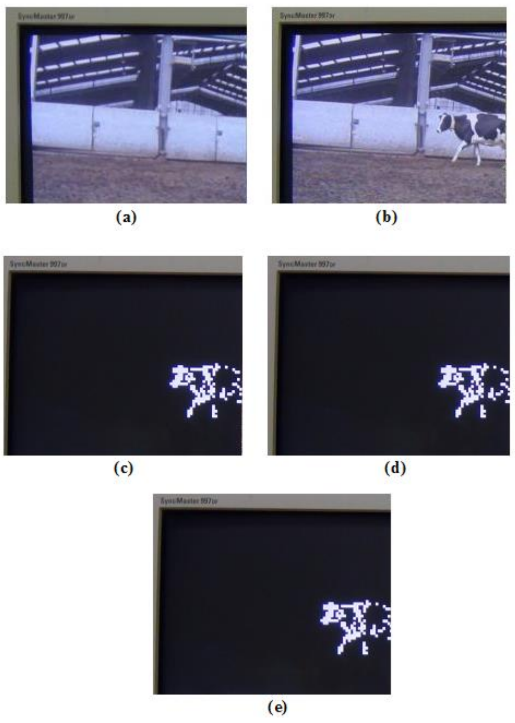


Fig. 11: Initial frame 1 (a), frame no. 23 (b), change detection result of Implementation I (c), Implementation II (d), and Implementation III (e)

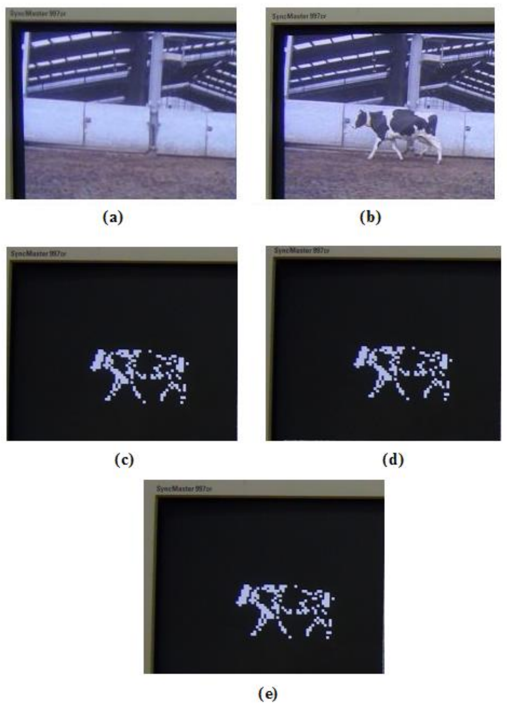


Fig. 12: Initial frame 1 (a), frame no. 38 (b), change detection result of Implementation I (c), Implementation II (d), and Implementation III (e)

Acknowledgements

Sanjay Singh would like to thanks Mr. Raj Singh, Group Leader, IC Design Group for his constant guidance, suggestions, support and encouragement.

References

[1] Meng H, Pears N E, and Bailey C. FPGA Based Video Processing System for Ubiquitous Applications[C]. Proceedings of Perspectives in Pervasive Computing. London, October 25, 2005.

[2] Video and Image Processing Design Using FPGAs. White Paper from Altera, 2007.

[3] Said Y, Saidani T, Smach F, Atri M, and Snoussi H. Embedded Real-time Video Processing System on FPGA[C]. Proceedings of 5th International Conference on Image and Signal Processing. Agadir, Morocco. June 28-30, 2012.

[4] Zhang B F, Yang Y, Zhu J C, and Li C. Embedded Real-time Image Processing System Based on DM6446+FPGA Architecture[J]. Advanced Engineering Forum, 2012, 6-7: 542-546.

[5] Said Y, Saidani T, Smach F, and Atri M. Real-time Hardware Co-Simulation of Edge Detection for Video Processing System[C]. Proceedings of 16th IEEE Mediterranean Electrotechnical Conference. Monastir, Tunisia. March 25-28, 2012.

[6] Hiraiwa J, Vargas E, and Toral S. An FPGA based Embedded Vision System for Real-time Motion Segmenetation[C]. Proceedings of 17th International Conference on Systems, Signals and Image Processing. Brazil. June 17-19, 2010.

[7] Samanta S, Paik S, Gangopadhyay S, and Chakrabarti A. Processing of Image Data using FPGA-based MicroBlaze Core. Proceedings of International Conference on High Performance Architecture and Grid Computing. Chandigarh, India. July 19-20, 2012.

[8] Chutani E R and Chaudhury S. Video Trans-coding in Smart Camera for Ubiquitous Multimedia Environment[C]. Proceedings of International Symposium on Ubiquitous Multimedia Computing. Hobart, Australia. October 13-15, 2008.

[9] http://www.xilinx.com/univ/xupv2p.html

Authors' Profiles

Sanjay Singh is working as Scientist in CSIR-Central Electronics Engineering Research Institute, Pilani, Rajasthan, India. He is Member of IEEE - USA, IACSIT - Singapore, and IAENG - Hong Kong. Currently, he is involved in various projects sponsored by Indian Government on Computer Vision and Smart

Cameras. His research interests are VLSI architectures for image & video processing algorithms, FPGA Prototyping, and Computer Vision. Prior to joining this research lab, he received his Master in Technology, Master in Electronics, and Bachelor in Science in 2007, 2005, and 2003 respectively from Kurukshetra University, Kurukshetra, Haryana, India. He earned Gold Medal (First Position in University) during his Master in Technology and Master in Electronics. He topped college during his Bachelor in Science. He received more than 20 Merit Certificates and Scholarships during his academic career.

Ravi Saini is working as Scientist in CSIR-Central Electronics Engineering Research Institute, Pilani, Rajasthan, India. He is Member of IEEE – USA. His research interests include VLSI Architectures, ASIC and ASIP Design, HDLs, and FPGA Prototyping. He received his Master in Technology in 2002 from Punjab University, India and Master in Electronics in 2000 from DAVV, Indore, India.

Anil Kumar Saini is working as Scientist in CSIR-Central Electronics Engineering Research Institute, Pilani, Rajasthan, India. He is Member of IEEE – USA and IACSIT – Singapore. His research interests include Analog and Mixed Signal Design, Embedded System Design, and CMOS RF IC design. Prior to joining this research lab, he worked in Cadence, India. He received his Master in Technology and Master in Science in 2003 and 2000 respectively from IIT Roorkee, India.

Dr. A.S. Mandal is working as Chief Scientist in CSIR-Central Electronics Engineering Research Institute, Pilani, Rajasthan, India. His research interests include Embedded VLSI Design, Computational Nuero-Vision, and Perception and Cognition Engineering. He is project leader for various projects in these areas. He delivered many invited talks in Indian Universities and Institutes on topics related to his research areas. He received his M. Tech. and Ph. D. from IIT Delhi, India.

Dr. Chandra Shekhar is Director of CSIR-Central Electronics Engineering Research Institute, Pilani, Rajasthan, India. His research interests include VLSI Design and Design Methodologies, Analog IC Design and Mixed Signal Design, Processors and Application Specific Processors (Architecture and Design), CAD for VLSI, Physics and Modeling of MOS Devices, VLSI System Applications. He received M.Sc. degree in Physics in 1971 and Ph.D. degree in 1975, from BITS, Pilani, India. He is a Fellow of IETE and Life Member of Indian Physics Association, Semiconductor Society of India and Indo-French Technical Association.

Dr. Anil Vohra is Professor in Electronic Science Department, Kurukshetra University, Kurukshetra, Haryana, India. Currently, he is holding the position of Dean, Faculty of Engineering and Faculty of Sciences. He is member of various committees of university. He has published several research papers in various reputed international/national journals and conferences. He leaded various projects sponsored by DST and AICTE, India. He received UK Govt. Commonwealth Research Fellowship in 1998, Japanese Govt. Research Fellowship in 1991, Dr. K.S. Krishnan, DAE Research Fellowship in 1985 and CSIR Research Fellowship in 1985 & 1987. He received his M.Sc. and Ph.D. from Panjab University, Chandigarh, India.

How to cite this paper: Sanjay Singh, Ravi Saini, Anil K. Saini, AS Mandal, Chandra Shekhar, Anil Vohra,"Performance Evaluation of Different Memory Components for FPGA based Embedded System Design for Video Processing Application", International Journal of Intelligent Systems and Applications(IJISA), vol.5, no.12, pp.113-119, 2013. DOI: 10.5815/ijisa.2013.12.10