Dedicated Hardware Processors for a Real-Time Image Data Pre-processing Implemented in FPGA Structure

Kazimierz Wiatr
Institute of Electronics, AGH Technical University of Cracow
Reymonta 23, 30 059 Cracow, POLAND, wiatr@uci.agh.edu.pl

Abstract: This paper presents a dedicated hardware processors implemented in FPGA structure for a fast video image data pre-processing to real time application. Author design and made specialised pipelined multiprocessor architecture for specialised hardware processors. This paper presents specialised hardware processors: median filter, logic processor, look-up-table processor, convolution processor and histogrammer. Dedicated hardware processors implementation in the Xilinx FPGA used another chips. This work is supported by Polish Science Comitee.

1. A real time video system

A real time video signal analysis in the classic method require many computation power. This problem generated searching another realisation method for image processing applications in robotics, moveable objects tracking and guidance (road traffic, military applications, industrial TV etc.).

Image processing algorithms can be conventionally divided into several levels. Algorithms of low level image processing are typically performed as a sequence of simple operations such as a look-up-table, binarization, median filtering, convolution, logic operation and others. The middle level image processing makes image segmentation, object localisation, recognizing form and character form indication. High level image processing analyses complicated scene: movement object detection and real-time object control [12]. The high level image processing generates algorithms and parameters for low and middle level image processing.

The fig. 1 presents connexion and feedback between those levels. The situation is very complicated when low level image processing operations are realisated in the hardware processors. High level can require change of operation realised by dedicated hardware. Dedicated hardware processors application in the FPGA structure is a good solution in this situation. Dedicated hardware processors can be programmed by Master Processor in the real-time mode.

2. A specialised pipelined architecture

Conventional approach to the video image processing is connected with many data transfers between video memory buffer and microprocessors units. Therefore, eliminating time consuming transfers and applying the pipelined image processing system seems to be valuable solution (fig. 1). This architecture particulary is a destination for low level image processing [14].

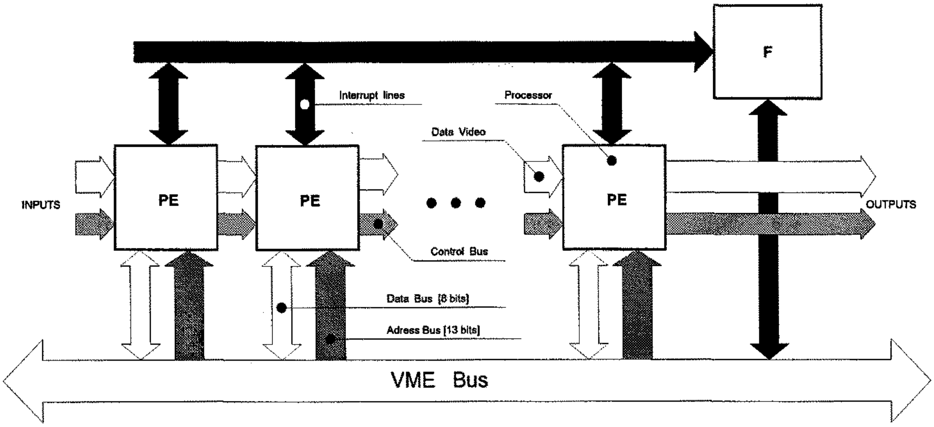

Fig. 1. A specialised hardware processors architecture

A special, custom pipeline bus standard was designed to provide convinceable way of applying pipeline processors. The bus involves a number of slots (6 slots in designed system). Each one provides signals, like: Video Data Bus (daisy-chained) with Horizontal and Vertical Synchro, a number of clock signals (Video Data Latch Clock, Pixel Clock, System Clock), VME Bus interface (Address & Data Bus + proper control signals), VME interrupt interface and power supply.

The pipeline system claims very high marks from its elements. Each processor in the pipe should process a full frame before the next one will come. The time is a direct function of: frames per second count, horizontal lines number and finally pixel per line number. In television standards (SECAM, PAL) it is 25 frames per second (50 fields per second) with 625 lines per one frame. Number of pixels in the analysed line comes from A/D converter sampling frequency. In image processing systems a square field (equal horizontal and vertical dimensions) with square pixels is commonly used. It is also easy to interface a memory chips when resolution 512 x 512 of the frame window is fixed. To preserve the square pixel shape, the sampling frequency is valuated as follows 14,7 MHz.

3. Dedicated hardware processors

Author design and made some dedicated hardware processors. For example was realization the median filtering processor (fig. 2). This processor realisated by using FPGA structure [15] and two FIFO buffers. The processor allows to make a median filter based on reduced 9 or 5-elements kernel mask (the fig. 2 presents 9-elements kernel mask). In the FPGA structure are implementation 45 comparators by 8 bits everyone (for 5-elements kernel mask only 10 comparators), multiplexer and 9 registers by 8 bits. This processor was realised by involving Xilinx FPGA (XC4003-5PG120C) and two FIFO buffers (IDT72210L15TP).

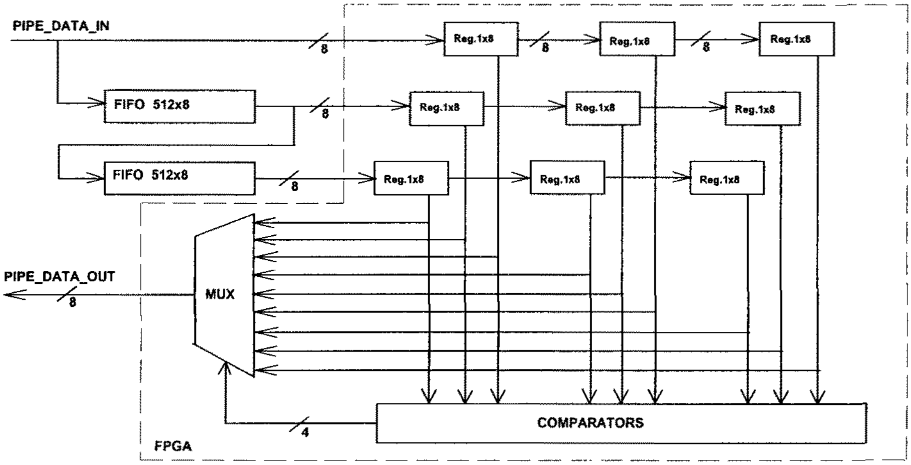

Fig. 2. Median processor

Other example (fig. 3) presents subtraction processor. In this realisation are used Triple Port RAM and FPGA structure. The unit performes a subtraction of two images: first one is a currently anaysed image; the second one is an image stored in TPRAM memory buffer. The second image may be stored in the buffer through the VME bus or may be stored as a one of the preceeding images. The TPRAM (Triple Port RAM) contains, besides the standard parallel port, 2 independent 8bit-wide and 512bytes-long serial ports. Than one of them can be used as an output port providing data for subtraction (as a background) and the second one can be used as an input port for storing current image for the next frame subtraction. This processor was realised by involving Xilinx FPGA (XC4004-5PG120) and two Micron 4-bits Triple Port RAM (MT43C4257).

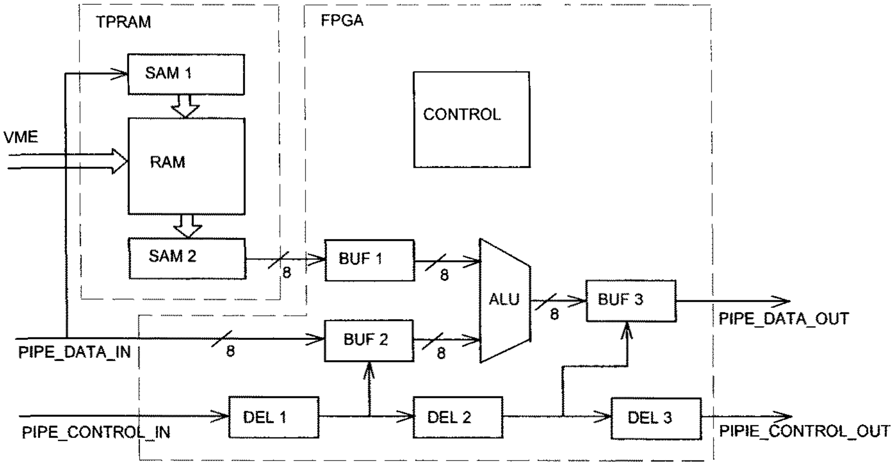

Fig. 3. Subtraction processor

Histogram computing is very important operation for data image preprocessing. This processor counting in the local memory cell pixels at equal grey level. Memory cell (word) must be 20 bits because all image perhaps be one grey level. In the practical realisation (fig. 4) was application two multiple counter in the RAM memory. This realisation result from high speed image data flow. This processor was realised by involving Xilinx EPLD chips (XC72108-16WC84C).

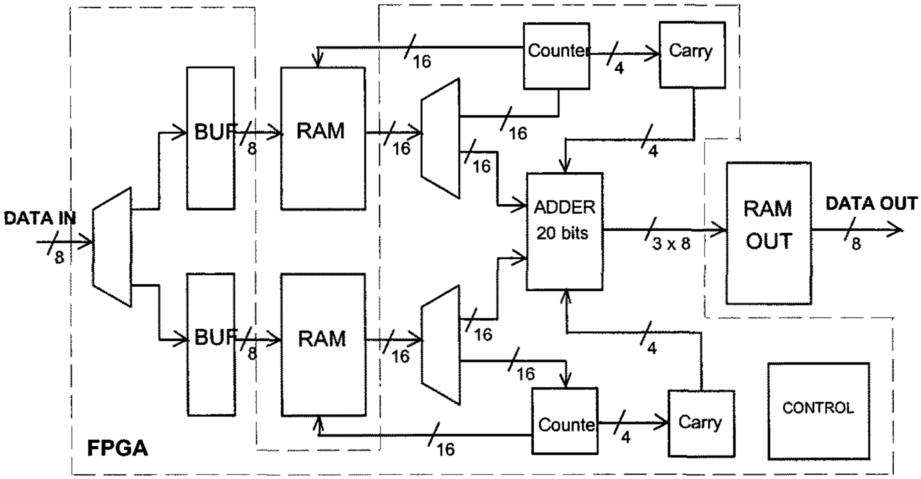

Fig. 4. Histogram processor

The next worked processor is a look-up-table module (fig. 5). The module was built using only FPGA device. There were 2 designs made. First one as a LUT array uses a ROM memory implemented in the FPGA device. Than alteration of the memory contents (changing LUT function) requires reconfiguring the whole device. The second design uses a RAM memory implemented in the FPGA device. The memory contents can be accessed throug the VME bus. This design requires more hardware resources. and is slower (2 pixel period delay in pipeline data flow). This processor was realised by involving Xilinx FPGA chips (XC4005-5PC84).

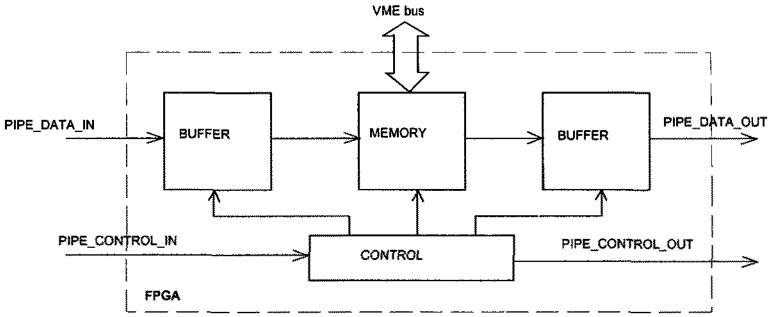

Fig. 5. Look-up-table processor

4. Reconfigurable pipelined hardware processors

Author designed an universal reconfigurable pipelined processor module (fig. 6). This module is composed of Triple Port RAM, two FIFO buffers and Xilinx FPGA [15]. Such universal module enable realisation many dedicated hardware processors for image processing.

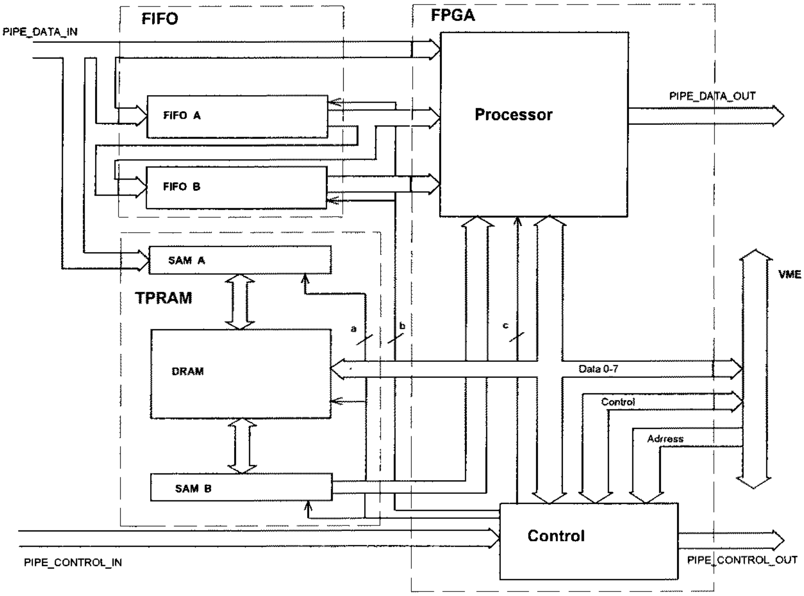

Fig. 6. Reconfigurable pipelined processor

In this reconfigurable pipelined processor structure is possibility implemented median processor, convolution processor, logic processor, histogramm processor and other. In the practical realisation it can be used one, two or all composition elements on this module. For example to implemented in this reconfigurable universal processor the median filtering processor is realisated by using FPGA structure and two FIFO buffers. For implemented logic processor must used FPGA structure and Triple Port RAM. Histogramm processor and look-up-table processor used only FPGA structure.

Reconfigurable universal hardware processor implemented in the FPGA structure (for example XC4000) is a good solution in this application. Reconfigurable hardware processor can be programmed by Master Processor in the real-time mode.

5. Conclusion

Presented in this paper dedicated hardware processors implemented in FPGA structure for specialised pipelined architecture is very competitive to conventional computation. Pipelined multiprocessor architecture eliminated many time consuming for data video transfer. Reconfigurable hardware processor lets for a realization of different algorithms in the universal hardware.

Time operation's in that hardware processor is even:

- look-up-table (ROM)	- 66 nsec.,
- look-up-table (RAM)	- 132 nsec.,
- median filtering (5 pixels)	- 68 μsec.,
- median filtering (9 pixels)	- 68 μsec.,
- subtraction two frame	- 66 nsec.,
- convolution 3x3	- 102 μsec.

For compare in the Data Translation dedicated modul DT2878 (DSP32C) time operation's for convolution 3x3 is even 1180 msec. and for adding two frames is 250 msec. Other modul FT200 made by Alacron (two i860 processors) is realizated operation of convolution 3x3 by 51,9 msec. [1].

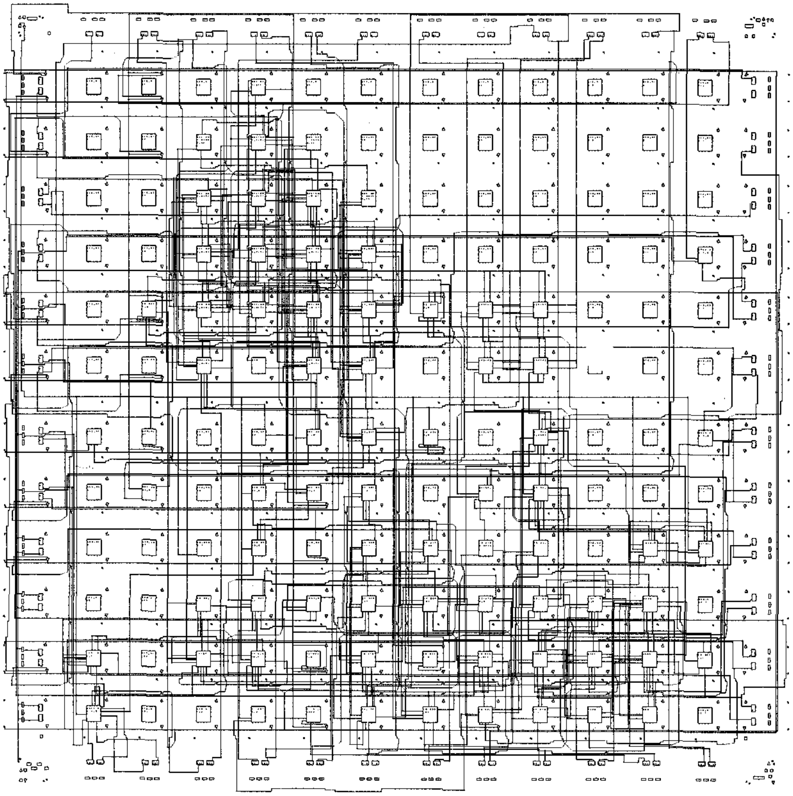

Fig. 7. Logic processor implemented in FPGA structure (XC4004-5PG120)

Full realisation made in used XACT ver. 4.2 development system with OrCAD interface as a schematic entry and simulation tool (ver. 4.1 and 386). Fig. 7 present logic processor implemented in FPGA structure print from level XDE (XACT Design Editor). On the finish this text place PPR Results Report (exceptions). In particullary this report present timing analysis summary (delay pad to pad, pad to setup, clock to pad etc.).

Made pipelined multiprocessor unit for a fast image data pre-processing in real-time application was realised in the VME bus standard. It works with autonomus VME modules (CPU MC68030, local RAM, EPROM, I/O modules) runing MICROWARE's real-time OS-9 operating system. Under development are system functions as well as grafucar user interface with MGR window graphic package.

I would like to thank my collaborators and my graduate students who have been making many of the routines discussed in this paper. This work is supported by Polish Science Comitee. Their help and support are greatly appreciated.

References

[1] Advanced Imaging, PTN Publishing Co, New York June 1995.

[2] N. A. Andresen, O. Ravn: Vision as a Multipurpose Sensor in Real-Time Systems, Proc. of the ISMM International Symposium: Mini and Microcomputers and Their Applications, Lugano Switzerland 1990.

[3] G. W. Awcock, R. Thomas: Applied Image Processing, Mc Graw Hill 1995.

[4] S. C. Chan, H. O. Ngai, K. L. Ho: A Programmable Image Processing System using FPGA, Proc. of the IEEE Int. Symposium on Circuits and Systems, London 1994.

[5] A. L. Decegama: The Technology of Parallel Processing - Parallel Processing Architectures and VLSI Hardware. Prentice Hall, New Jersey 1989.

[6] M. Gorgoń: FPGA Implementation of Dedicated Image Processor, Proc. of the 7-th School: VLSI and ASIC Design, Format Wrocław Poland 1995.

[7] K. Hwang, F. Briggs: Computer Architecture & Parallel Processing, Mc Graw-Hill 1984.

[8] Y. Iwata, M. Kawamata, T. Higuchi: Design of fine VLSI arraay processor for real-time 2-D digital filtering. IEEE Symposium on Circuits and Systems, Chicago 1993.

[9] P. P. Jonker: Morphological Image Processing: Architecture and VLSI design. Kluwer Holland 1992.

[10] J. Kittler, M. J. B. Duff: Image Processing System Architecture, John Wiley & Sons, New York 1985.

[11] B. Liebowitz, J. Carson: Multiple Processor Systems for Real Time Applications, Prentice Hall 1985.

[12] R. Tadeusiewicz: Computer System Dedicated to Image Processing, Proc. of the 5-th School: Computer Vision and Graphics, Format Wrocław Poland 1994.

[13] A. Tsutsui, T. Miyazaki, K. Yamada, N. Ohta: Special Purpose FPGA for High-speed Digital Telecomunication Systems, Proc. of the IEEE Intern. Conf. on Computer Design: VLSI in Computers & Processors, Austin - Texas 1995.

[14] K. Wiatr: A Pipelined Multiprocessor Architecture for a Fast Image Data Pre-Processing in Real-Time Applications. Proc. of the IASTED Int. Conference: Signal and Image Processing, Las Vegas - Nevada 20-23.11.1995.

[15] Xilinx: The Programmable Logic Data Book, San Jose 1996.

PPR - Xilinx Results for Design Logic Processor Implementation (exceptions)

```
*****************************
XMAKE: begin command 'sdt2xnf -x -
p4004APG120-5 gr_out.inf'
SDT2XNF Ver. 4.20
(c) Copyright 1987-1992 Xilinx Inc.
*****************************
XMAKE: begin command 'ppr xi.xnf
ppr[1.30] Xilinx Automatic CAE Tools

+ ppr  @ 1996/11/20 13:03:42
+ Parameters
   design        = xi.xnf
   parttype      = 4004APG120-5
   justflatten   = FALSE
   estimate      = FALSE
   logfile       = ppr.log

+ Additional Specified or Non-
Default Parameters
seed          = 848495022

Speedfile version 9, revision 8.
+ Converting XNF to MXN  @ 13:03:45
+ Checking Netlist  @ 13:03:50
+ Partitioning the Netlist  @ 13:03:52

Preliminary evaluation of your selected
part, 4004APG120:
89% utilization of io pins
( 85 of  95)
40% utilization of function generators
(116 of 288)
37% utilization of clb flip-flops.
(106 of 288)
25% utilization of bus resources.
( 12 of  48)

+ Generating .LCA File  @ 13:05:45

Begin work on a 19.8ns path with 1 pins
(slack  13.5ns).
Cycle detected...
--------------------------------------------
Timing analysis summary
--------------------------------------------
Deadline  Actual(*)  label: [qualifier]
-------------  ---------  ----------------
pad to pad          <auto>   56.2ns
<default>
----------------------
Selector net: Default
----------------------
clock to setup       <auto>   40.5ns
(rising edges)
clock to setup       <auto>    8.5ns
(falling edges)
clock to setup       <auto>    7.9ns
(rising to falling edge)
pad to setup        <auto>   42.0ns
clock to pad        <auto>   64.8ns
----------------------
Selector net: ODE/CLK_OUT
----------------------
clock to pad     33.3ns   19.8ns    TS1:
----------------------
Selector net: PR/N_ADR_P
----------------------
 clock to setup    33.3ns    19.1ns    TS2:
 clock to setup    33.3ns     9.1ns    TS2:
            (from other)
clock to pad     <auto>   19.0ns

Resource_usage: Pips 1398, Feeds 22,
Locals 418, Doubles 234, Longs 149,
Globals 9, Dec_lines 2
# of unrouted connections: 0.

+ Making Report File  @ 13:07:52
[00:04:20]

*****************************
Xilinx LCA MAKEBITS Ver. 4.31
ended normally
```