The International Workshop
on Discrete-Event System Design, DESDes'01,
June 27÷29, 2001; Przytok near Zielona Gora, Poland

FLEXIBLE RESOURCE ARBITER FOR HETEROGENOUS IMAGE PROCESSING SYSTEM

Jaromir PRZYBYLO, Marek GORGON

Biocybernetic Laboratory, Institute of Automatics, AGH University, al. Mickiewicza 30,
30-059 Krakow, Poland, <*phoenix@biocyb.ia.agh.edu.pl; mago@biocyb.ia.agh.edu.pl*>

Abstract. *In the present paper realisation of resource arbiter for RETINA image processing module has been described. The 32-bit RETINA module is used for image acquisition, processing and analysis. The module's resources include A/C converter, Virtex FPGA device, Motorola 96002 floating-point DSP device and PCI Master interface and they allow real-time realisation of those operations. Resource arbiter is an important control block of the module. It is responsible for the resource allocation for the main control elements of the module, it arbitrates the allocation of internal and external buses, and keeps the information concerning the system state.*

Key Words. *arbiter, real-time, FPGA, re-configurable computing, image processing*

1. INTRODUCTION

The technological progress in the field of production of FPGA devices in the last two years, oriented towards tailoring of the resources of FPGA devices for the needs of digital image processing, makes possible realisations of hardware or software-hardware vision systems. The mostly widespread way of realisation of digital image processing algorithms is the software method [1][2][3][4]. The software's basic advantage is the possibility of convenient and flexible realisation of the algorithm. Because of the high calculational complexity of the image processing algorithms their software realisation is not always efficient enough to allow the algorithm's work in real time. Therefore search continues for multiprocessor architectures [5], hardware methods [6][7] and software-hardware methods [8][9] speeding up the execution of calculations. Great popularity has been achieved by the realisation of image processing and analysis algorithms in DSP [10][11] and specialised or dedicated hardware processors [12][13]. Many attempts have been made of realisation in complex, multiprocessor architecture [5][7][8]. Solution earning a steadily growing popularity is the implementation of processing algorithms, and recently also image analysis algorithms, in reprogrammable devices [12][13][14][15].
Essential advantages of the FPGA-based architectures for image processing are their flexibility, efficiency and structural adaptation to tasks consisting of multiple and parallel execution of algorithms for relatively simple data like image pixels.
In the present paper heterogeneous architecture has been shown, in which a single FPGA chip of high densities has been used both for realisation of the image processing and controlling the resources of the device itself. The paper contains the discussion of architecture and

working modes of that part of the implemented FPGA chip, which realises the function of the resource arbiter for the constructed image processing system.

2. THE ARCHITECTURE OF THE RETINA HETEROGENEOUS IMAGE PROCESSING SYSTEM.

The 32-bit architecture of heterogeneous image processing system is based on Virtex device working in co-operation with 96002 floating point DSP. The board is equipped with high-speed analogue to digital converter, several memory blocks, real-time clock and 32-bit PCI Master interface (AMCC). The Virtex chip combines two functions. It contains all the necessary control logic (FPGA Controller) and is used for performing the image processing operations (Video Processor) - see Fig.1.

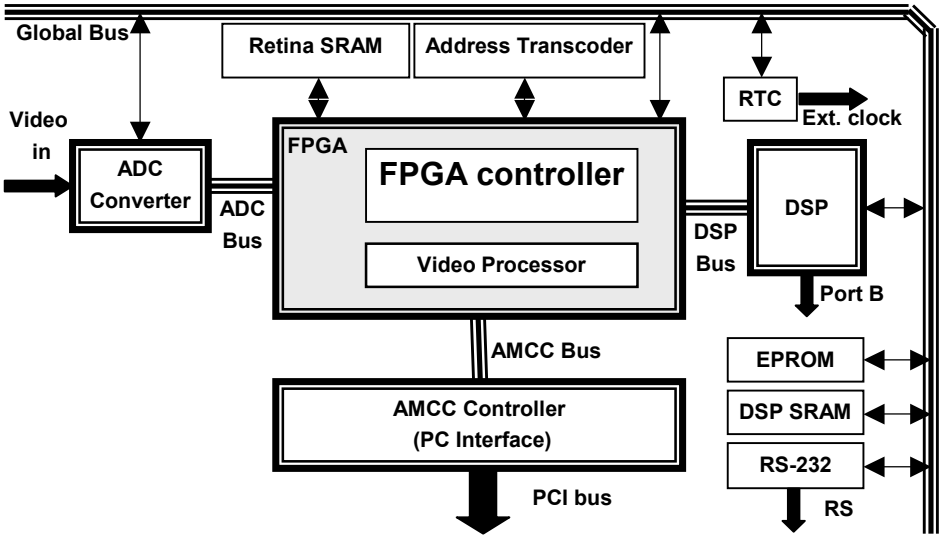

Figure 1. The architecture of heterogeneous image processing system.

The board contains three that can be treated as master devices - DSP96002, Video Processor and PC (through AMCC chip). Optimising the data transfer between the master devices and memory blocks was an essential goal of implementation of the FPGA Controller.
The FPGA Controller (Fig.2) consists of four modules - three dedicated local sub-controllers (Video Processor Controller, DSP Controller and AMCC Controller) and the resource arbiter module. The local sub-controllers are responsible for local arbitration and matching signals between devices. Therefore parallel work is enabled. Resource arbiter controls the data transfers between master devices.

3. RESOURCE ARBITER MODULE

The proposed resource arbiter module enables data exchange between master devices and synchronises their access to memory resources of the module. It enables various arbitration schemes (e.g. token ring), and due to that the possibility of conflict occurrence between devices with "master" privileges. It also enables flexible management of the data transfer by application of two types of device priorities - global and local. If necessary the arbiter's configuration can be changed by using the possibility of reconfiguration of the FPGA Virtex device.
The system bus arbiter module (see Fig.2 - module D) consists of the following elements:

- the resource arbiter (RA) itself
- configuration registers (FPGA Registers)

- buses: FPGA Bus1,2,3 and Internal Bus
- additional arbitration signals

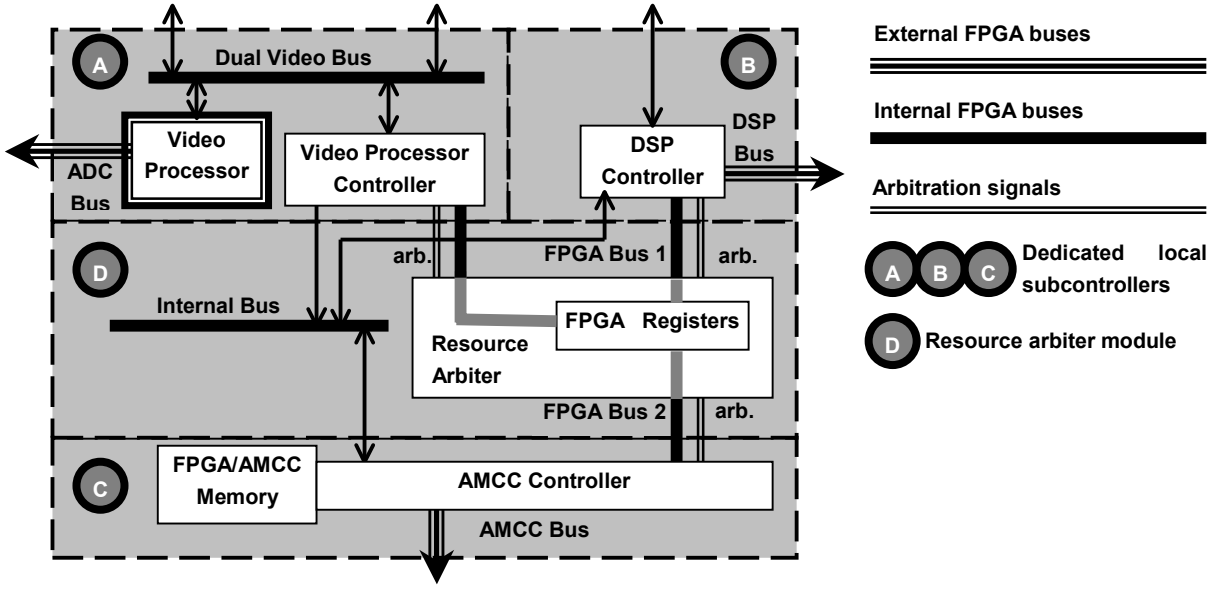

Figure 2. General layout of the controller.

The resource arbiter makes use of the configuration registers, containing the information on the priorities controlling the interrupts and control semaphores governing the arbitration, attributed to particular modules. Their state is mapped to local address spaces, what allows an independent access in any moment - with an essential restriction for the DSP block, when the external global bus is being used by another device. The configuration registers are directly connected with local controllers via FPGA Buses 1,2,3. The resource arbiter also obtains, due to dedicated controllers, various control signals (e.g. interrupts) and the remaining arbitration signals - allowing supervision over the functioning of every "master" device (e.g. reclaiming some resources).

4. THE ARCHITECTURE OF RESOURCE ARBITER.

The architecture of the Resource Arbiter provides the possibility of taking full advantage of module's resources. The introduction of co-operation of three devices provided with potential possibility of work in "master" mode leads to the necessity of ensuring sufficient resources, allowing the device's work in various configurations. The application of the FPGA device allows changes of the control module's infrastructure to be realised fully in hardware, in order to provide the possibility of the device's work in various modes and configurations.
Resource Arbiter consists of the following blocks (see Fig.3):

- Arbitration Unit (AU) - the main control unit of the arbiter, responsible for conflict arbitration and management of the modules resources, containing the Grant Register storing information about intermodular transfers currently taking place;
- Global Arbitration Logic (GAL) - the block co-operating with local arbitration systems;
- Interrupt Controller (IC);
- Control Unit (CU) - the unit generating the clock signals and RESET signals;
- Watch Dog (WD) - the block containing timers, used for supervision of the correctness of particular module blocks functioning;
- Additional Logic - additional auxiliary chips, not included in the block diagram.

Local Arbitration Logic systems have been related with every master devices. The LAL's take over the local arbitration tasks from RA, while the Resource Arbiter synchronises the data transfer between the modules. The arbitration, because of the specific features of the DSP, has been realised in software-hardware manner and it is based on solutions applied in PCI and DSP96002 processor.

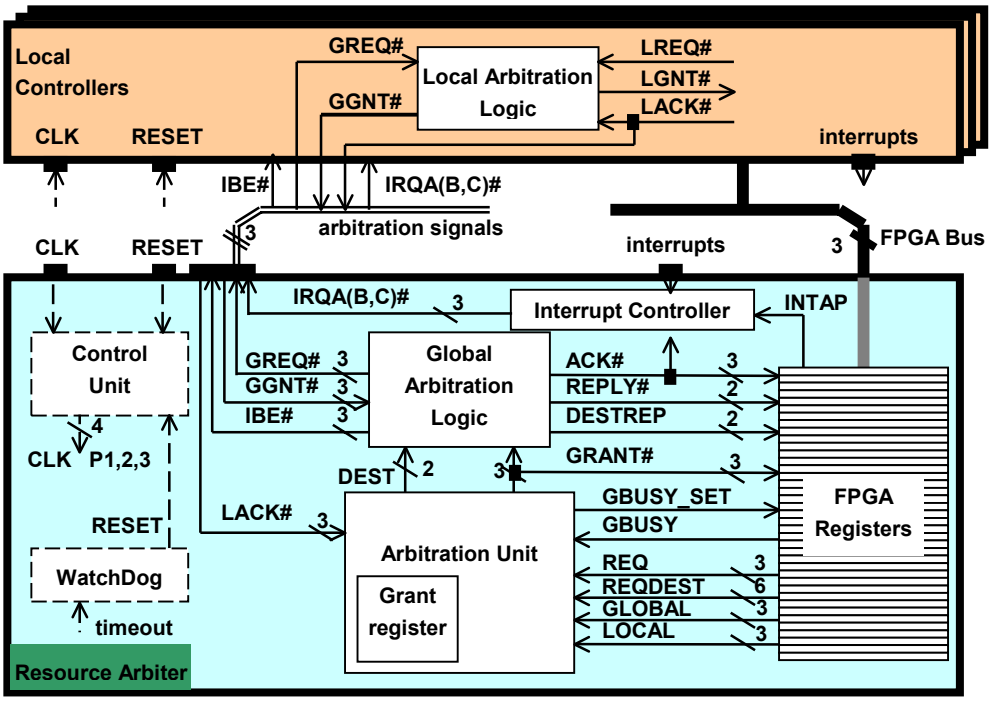

Figure 3. Layout of the Resource Arbiter.

For configuration and setting of the arbiter's working modes control flags semaphores have been used, located in configuration registers (FPGA Registers):

- Interrupt Register (8-bit) - the respective bits of the register are responsible for masking the interrupts, which inform the local "master" devices about allocation of the required resources;
- Priority Register (8-bit) - store the information about local and global priorities;
- Request/Acknowledge Register (3x3-bity + 1 bit) - contains the respective semaphores controlling the arbitration;
- Auxiliary Registers;

4. FUNCTIONING OF THE RESOURCE ARBITER

The operation of the Resource Arbiter should be analysed taking into account the working algorithms of the Local Arbitration Logic systems (LAL's), which take over the local arbitration task from the RA.

The work cycle of the arbitration systems can be divided into several stages, which are supervised by the Control Unit (CU), responsible also for initialisation (RESET) of the module and generation of the clock signals. Division of the work cycle into phases allows the elimination of changes of the FPGA Registers contents during the Resource Arbiter's work, what might lead to irregularities of its functioning. During the arbitration cycle the following phases can be distinguished (see Fig.4):

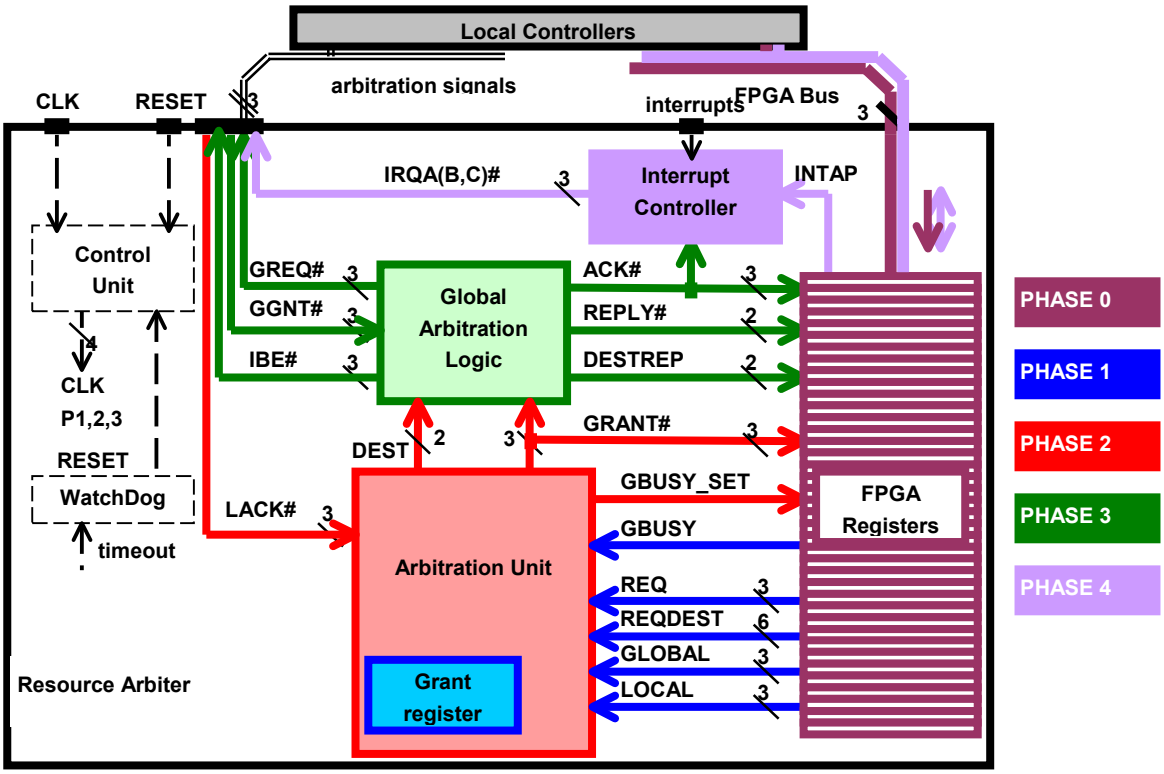

Figure 4. Phases of the arbitration cycle.

- Phase 0 - Writing to FPGA Register by the "master" devices (setting the semaphores) - requests for intermodular transfers (optional phase).
- Phase 1 - Start of the arbitration in the Arbiter system - collection of the information concerning requests, priorities from the FPGA Registers. In absence of Phase 0 initialisation of the Grant Register
- Phase 2 - Resolving the conflicts in the AU unit. Setting of appropriate resource allocation signals for the GAL system and reservation of the internal bus.
- Phase 3 - Negotiation of the resources reclaiming from the current user (its GAL block and local LAL system). After regaining control over the resources - setting the confirmation signal for the initiator. Possible storage of local resources reclaim requests, generated by the previous user
- Phase 4 - Realisation of the intermodular transfer (optional phase) - reception of information concerns the allocation of Internal Bus. Bus reclaiming after finishing the transfer.

5. Conclusions.

The only possibility of fulfilling all the requirements that should be met by the Resource Arbiter of the RETINA image processing module, was its implementation in the reprogrammable FPGA device. Such a solution allows the realisation of flexible and changeable arbiter structure fully in hardware. It provides a possibility of module adaptation for realisation of various arbitration procedures. The important thing is the possibility of configuration of the FPGA's internal memory resources as the module's configuration registers. The hardware implementation ensures great operation speed and high integration level of the arbiter's structure. There is also a possibility to adapt the device's structure to specific algorithms implemented in the system during its usage by the end-user.

The possibility of gradual development of the arbiter's structure during the prototype's testing should be also highly appreciated. The resources of the FPGA device allow a construction of additional modules, supporting the developing process of the arbiter and the FPGA module as a whole. Flexible FPGA structure opens a series of possibilities of RETINA module testing and monitoring of its co-operation with external systems via the PCI bus and serial port.
The implementation has been done in Xilinx Virtex XCV 300BG432-6 device. A prototype of the RETINA card is supplied with BGA432 socket, so XCV800E is considered as a final implementation platform.

ACKNOWLEDGEMENTS

We wish to thank the Xilinx University Program for software donation. The research has been performed under University of Mining and Metallurgy Research Grant no.: 10.10.120.39.

REFERENCES.

1. PRATT W.K., Digital Image Processing, John Wiley & Sons, Inc., 1991.
2. TADEUSIEWICZ, R., Komputerowa analiza i przetwarzanie obrazów, Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków 1997.
3. ULLMAN, S., High-level Vision, The MIT Press 1997
4. UMBAUGH, S. E., *Computer Vision and Image Processing*, Prentice Hall International, Inc., 1998.
5. TADEUSIEIWICZ, R. The multiprocessors architectures for image processing, *Lecture Notes on Computer Vision and Artificial Intelligence*, Warszawa, pp. 226-269, 1989.
6. KUNG, S.Y., VLSI Array Processors, Prentice Hall, New Jersey 1989.
7. JONKER, P.P., Morphological Image Processing: Architecture and VLSI design, Delft University of Technology, 1992.
8. LANGE, A.A.J., *Design and Implementation of Highly Parallel Pipelined VLSI System*, Delft University of Technology, 1991.
9. DAGLESS, E. at al., Image Processing Hardware, *Proc. of the 5-th School Computer Vision and Graphics Zakopane*, Wyd. Format Wrocław 1994.
10. SOREL, Y., Real - Time Embedded Image Processing Applications using the A^3 Methodology, Proc. of ICIP – 96, IEEE 1996.
11. BRUDŁO, P., System przetwarzania i analizy sekwencji obrazów video w oparciu o sieć procesorów sygnałowych serii ADSP-21060, *Konferencja Systemy Czasu Rzeczywistego 2000*, Katedra Automatyki Akademii Górniczo-Hutniczej 2000.
12. WIATR, K. *Architektura potokowa specjalizowanych procesorów sprzętowych do wstępnego przetwarzania obrazów*, AGH Uczelniane wydawnictwa Naukowo-Techniczne, Kraków 1999.
13. GORGOŃ, M. 1995. Evaluation of reliability of dedicated image processors in low level image processing, *Ph.D. Thesis: AGH Kraków, 25 September 1995.*
14. GORGOŃ, M. 1997. Universal Reprogrammable Architecture for Implementation Dedicated Image Processors Based on FPGA. *Proc. of Sixth International Conference on Image Processing and its Applications IPA 97, Trinity College, Dublin, Ireland 14-17 July 1997*, IEE Conference Publication no.: 443, vol.2, pp. 556-560.
15. XUE, J, at al., Approach to constructing reconfigurable computer vision system, Proc. Of SPIE Vol. 4212, SPIE Photonics East Conference, Boston 2000.
16. XILINX 1999. AppLINX Rev.9 First Quarter 1999.