

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/235429591>

Survey of reconfigurable architectures for multimedia applications

Article in Proceedings of SPIE - The International Society for Optical Engineering · May 2009

DOI:10.1117/12.821713

CITATIONS

7

READS

160

7 authors, including:



Teresa Cervero

Universidad de Las Palmas de Gran Canaria

14 PUBLICATIONS 45 CITATIONS

SEE PROFILE



Sebastian Lopez

Universidad de Las Palmas de Gran Canaria

127 PUBLICATIONS 782 CITATIONS

SEE PROFILE



Gustavo Marrero Callico

Universidad de Las Palmas de Gran Canaria

146 PUBLICATIONS 771 CITATIONS

SEE PROFILE



Félix B. Tobajas

Universidad de Las Palmas de Gran Canaria

81 PUBLICATIONS 205 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



ENABLE-S3 [View project](#)



Special Issue "Real-Time Processing of Remotely-Sensed Imaging Data" [View project](#)

Survey of reconfigurable architectures for multimedia applications

T. Cervero*, S. López, G. M. Callicó, F. Tobajas, V. de Armas, J. López, R. Sarmiento

Instituto Universitario de Microelectrónica Aplicada

Univ. de Las Palmas de Gran Canaria, Campus de Tafira 35017, Las Palmas de Gran Canaria, Spain

ABSTRACT

In a short period of time, the multimedia sector has quickly progressed trying to overcome the exigencies of the customers in terms of transfer speeds, storage memory, image quality, and functionalities. In order to cope with this stringent situation, different hardware devices have been developed as possible choices. Despite of the fact that not every device is apt for implementing the high computational demands associated to multimedia applications; reconfigurable architectures appear as ideal candidates to achieve these necessities. As a direct consequence, worldwide universities and industries have incremented their research activity into this area, generating an important know-how base. In order to sort all the information generated about this issue, this paper reviews the most recent reconfigurable architectures for multimedia applications. As a result, this paper establishes the benefits and drawbacks of the different dynamically reconfigurable architectures for multimedia applications according to their system-level design.

Keywords: Multimedia application, reconfigurable architecture, ASIC, microprocessor.

1. INTRODUCTION

Modern society demands devices capable of combining several functionalities while maintaining an elevated grade of performance. The technological evolution has allowed and promoted the expansion of several research activities for implementing multimedia applications, such as image processing or video streaming. However, these applications have some drawbacks like their inherent highly repetitive parallel computations that represent a significant fraction of the overall computation required by the system, and their constantly necessity of upgradeability. As a consequence, worldwide research groups are trying to develop flexible and efficient devices not only able to support existing applications, but also able to offer new functionalities and services under the same system with the highest level of performance. Unfortunately, flexibility and efficiency appear as contradictory goals from the designer's point of view, since the improvement of one implies the deterioration of the other. Under these conditions, not every device is apt for implementing these multimedia applications.

Into the range of possible devices, reconfigurable architectures appear as an appropriate solution to meet this simultaneous demand for high computational performance and flexibility [1]. On one hand, simple general purpose processors (GPPs) are not enough to assume the computational requirements of current multimedia applications anymore, since total software execution is too slow and inefficient. On the other hand, traditional application specific integrated circuits (ASICs) are not enough flexible, mainly because they are designed for a specific application. Within the wide range of hardware solutions, reconfigurable architectures have already shown to be a potential solution. They allow configuring the system several times, supporting upgrades and refinements. Furthermore, they almost offer the functional efficiency of ASICs and the programmability of GPPs. These are the main reasons why they have received much attention over the last years.

Talking about reconfiguration means that the functionality of the system can be customized to suit a specific application through post-fabrication. In this sense, there exist two ways to program or configure a system: statically or dynamically. Static reconfiguration demands the interruption of the application execution before applying a newer configuration. On the contrary, dynamic reconfiguration allows activating different configurations without interrupt running application. Obviously, the last technique offers some important advantages to multimedia designers versus the static reconfiguration.

*tcervero@iuma.ulpgc.es

Due to current multimedia applications are submitted to real-time constraints, static reconfiguration is not used to design multimedia devices because it is too slow, inefficient, and not enough adaptive. Definitively, only dynamic reconfigurable architectures appear as solutions to the real-time execution restriction for the more recent multimedia applications. As it was mentioned before, the interest of the research community on reconfigurable architectures has considerably grown along this last decade. In this sense, papers such as [2], [3], [4] and [5] have generated important amounts of information and an improvement into the knowledge on this area. In spite of this ample know-how base, it is necessary to create uniform criteria in order to establish fair comparisons between different reconfigurable designs and also to help designers to take initial decisions according to the constraints of the final application to be implemented. Nowadays, the vast design space makes difficult to find optimum reconfigurable architectures because this process involves satisfying many trade-offs in choosing values for each parameter. As a consequence, this selection is currently heavily based on the designer's experience. However, whether these studies and experiences were grouped within a unique design's guide, the result would provide a detailed knowledge about the effects of the structural design on the performance of the resulting architecture according to the initial conditions and the final application.

In order to reach the ambitious objective mentioned above, this paper initially presents a review of the most recent reconfigurable architectures for multimedia applications, followed by their general classification according to a high-level criterion. Previously to this paper, some authors like Compton [6], Campi [7] and Abdin [8] have done efforts into this line of work. However, their papers do not explain clearly the advantages of each proposal in general terms. In fact, all of them describe their systems like a detailed set of elemental units; but they forget to mention the general functionality of their structure, that has to be the first criterion to consider when a new design is being planned. In order to solve this inconvenient, this paper discusses the benefits and drawbacks of employing dynamically reconfigurable architectures attending to their system-level structure and their final application issues.

The organization of this paper is divided into six sections, as follows. Section 2 presents the historical evolution of the electronic, focusing the interest on reconfigurable architectures for multimedia applications. Section 3 reviews this last decade of research activity on dynamically reconfigurable architectures, classifying them in three different groups according to the selected criteria. In Section 4, each of the architectures is described in order to know its architectural characteristics in detail. Thus, the benefits and drawbacks about the classified groups of architectures are exposed in Section 5. Finally, some conclusions are summarized in Section 6.

2. REVIEW OF RECONFIGURABLE COMPUTING

This section covers the trajectory of the reconfigurable architectures from their origins to nowadays, showing their diverse stages of evolution throughout their history, and concluding the route into the current state.

2.1 Historical antecedents

Reconfigurable computing concept is young, and it has been in existence for twenty years approximately. Nowadays it is more than a concept; it is a physical reality. It is important to note that this and the rest of the modern electronic devices are possible due to the development of the transistor, which was announced in 1947 by Bell's Lab. This fact is still today one of the main causes of the modernization of our society. Along these decades, the electronic world has suffered numerous transformations to reach the present technological level, from the first integrated circuit (ICs) in 60's years, to the modern chips.

Figure 1 depicts schematically the historical background of reconfigurable architectures, divided into five important stages: the origin of the modern electronic with the development of the transistor, the apparition of first ICs that made possible the development of a wide variety of programmable systems, the development of *Simple Programmable Logic Devices* (SPLDs) and *Complex Programmable Logic Devices* (CPLDs), and the introduction of *Very Large Scale of Integration* (VLSI) technology. This fast evolution has been possible thanks to the integration methodologies advances and the inclusion of new materials into the fabrication process.

First commercial FPGAs were introduced in 1986 by Xilinx Corporation [9]. A big practical difference between FPGAs and CPLDs is that the first one is volatile, while a CPLD is not. That means that an FPGA cannot hold its configuration when powered off. Simultaneously to the FPGAs success, the utilization of GPPs and ASICs, based on full-custom or semi-custom techniques, gained a lot of attraction during the 90's. While early reconfigurable architectures contained small numbers of transistors, new devices have quickly grown their capacity to millions of gates, allowing the utilization of more complex and powerful structures.

Although ASICs and general purpose microprocessors represent fully functional systems, they are not appropriated to assume the stringent demands of multimedia applications. For this reason, research activity has developed useful reconfigurable systems and platforms capable of supporting more powerful applications along these years, not only focusing on FPGAs but also on custom and semi-custom reconfigurable architectures. On Figure 1, shadow boxes represent the active research area centered on FPGAs and semi-custom reconfigurable circuits.

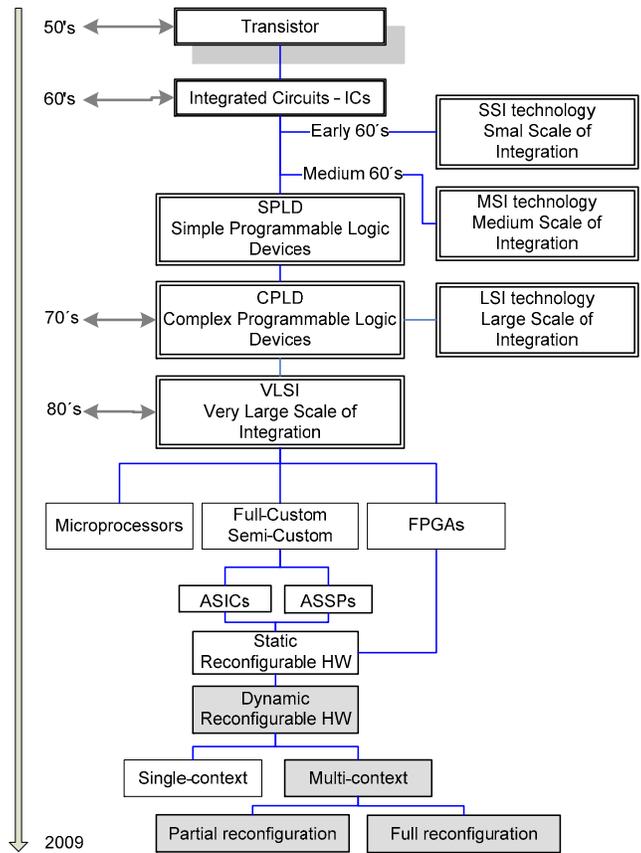


Fig. 1 Historical evolution of reconfigurable architectures

2.2 Multimedia applications

Modern multimedia applications, such as image processing or video streaming operations demand powerful systems to be executed. Within the wide range of programmable devices, dynamically reconfigurable architectures are the best choice. Definitely, they provide similar performance than ASICs, keeping an important level of flexibility, without redesigning the system, neither stopping the execution of the running application, in case of programming their functionality. In addition, because dynamically reconfigurable platforms allow the reuse of their resources for implementing several applications, the fabrication cost is reduced and also the time-to-market.

In order to cover real-time capabilities, dynamically reconfigurable architectures have to use a multi-context framework. This means that different configuration contexts are loaded into the configuration memory before the execution of the application, and then the system switches among contexts according to the environment's conditions. Furthermore, this method can be applied over the whole architecture at the same time (full system reconfiguration) or only to a part of the system while the rest is running (partial reconfiguration). On the other hand, single-context is not useful for current multimedia applications because its behavior is similar to a static reconfiguration.

There exist two design parameters which have an important impact over the final performance and the functionality of a reconfigurable architecture: granularity and reconfigurability. The reconfigurability term has been commented above;

establishing that dynamically reconfiguration is the most appropriate option for implementing real-time multimedia applications. Granularity is a term related to the data manipulation width, being possible to differentiate three classes of architectures attending to their granularity:

1. Fine-grain architectures: the system is programmed at bit-level.
2. Coarse-grain architectures: the system works at word-level.
3. Heterogeneous architectures: the system uses a combination of the previous ones.

Bit-level programming is less efficient than coarse-grain programming in terms of routing area, routability and configuration overhead. However, fine-grain architectures are well suited for application at bit or irregular data width manipulation. On the contrary, coarse-grain architectures reduces the configuration time, as well as it decrements the complexity of the place and route (P&R) process. This time reduction is due to that on coarse-grain architectures the configuration is at level of functional blocks grouping bits, while fine-grain architectures configure bit by bit each element. In this sense, coarse-grain architectures are more suited for multimedia applications.

Despite of the disadvantages of the fine-grain architectures, the scientific community is divided into two lines of actuation about designing dynamic reconfigurable architectures. One part of the current research experts prefer to centre their efforts in FPGAs (fine-grain) as a base of their developments. Conversely, the rest of developers and designers prefer to work in newer coarse-grain semi-custom architectures. In addition, it is worth to mention that there is a group of industries that commercialize dynamically reconfigurable FPGAs, such as Xilinx, Altera Corporation [10], Atmel Corporation [11] and Lattice Corporation [12]. Therefore, the architecture design usually is faster and easier using FPGAs instead of using semi-custom architectures [13]. Moreover, these industries have developed their own sets of designing tools, perfectly adapted to their FPGAs.

In order to be consistent with the previous paragraphs, next sections classify, describe and discuss state-of-the-art dynamically reconfigurable architectures designed for implementing multimedia applications. However, due to the ample activity into this area, next sections are limited to present a review of the most recent systems belonging to this decade.

3. RECONFIGURABLE ARCHITECTURES FOR MULTIMEDIA APPLICATIONS

On this decade many papers have been published describing different dynamically reconfigurable architectures for implementing multimedia applications. More specifically, Figure 2 depicts the most relevant ones presented from 1996 to 2009. This last year and the current one are not shown on this picture because their papers have contributed to develop more energy-saving techniques, models, methodologies and elements related to the design of these architectures instead of presenting novel architectures.

Despite of the important know-how base generated about this issue, data have not been sorted, neither classified. Thus, the goal of this section is to propose a criterion to do a general organization of these dynamically reconfigurable architectures.

Due to space limitations, it is not possible to present the whole group of architectures presented on Figure 2. For this reason, the following sections are focus on 11 of the 23 architectures. This selection contains the most remarkable architectures which projects are still running. Finally the candidates chosen to be described and classified are: MorphoSys [18], XPP [19], UltraSonic [22], DAPDNA-2 [23], DART [24], Montium [25], ADRES [29], XiRISC [30], 3D-SoftChip [31] and ECA [33], chronologically sorted.

3.1 Architectural classification

Deciding a classification is not easy. On one hand, its criterion must be enough general in order to allow, if were necessary, future subdivisions into smaller categories. On the other hand, it must be also enough specific so that its groups are defined by its own characteristics. Actually, there are several possible criteria. For example, according to the type of the programming elements, the topology, the interconnection systems, the communication protocol... But all these choices present an inconvenient; they are too specific and they require an important knowledge of the architectures under consideration. Therefore, if the objective is to generate a high-level organization, those are not proper criteria.

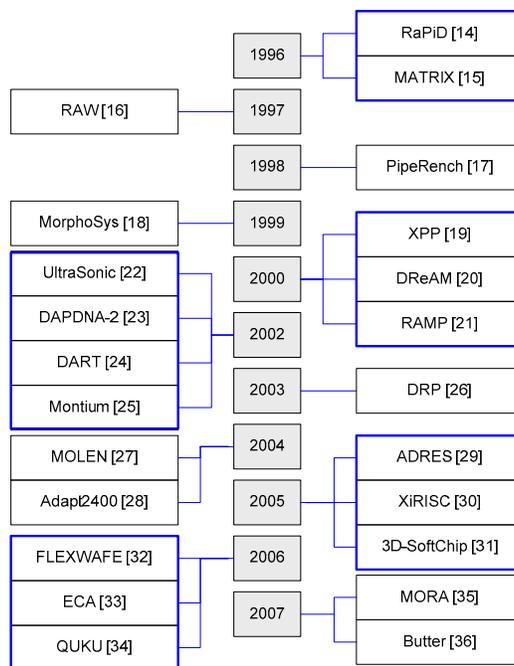


Fig. 2 Dynamically reconfigurable architectures

For instance, granularity could be a possible criterion to classify the architectures. It is general and it also allows defining several groups with different characteristics on it. However, this case is opposite to the previous one. Its problem is that it is too general.

Finally this article has selected an approach widely named in the literature, although in this case the classification is discussed from a practical point of view. The classification depends on the system-level architecture, according to the hardware disposition onto the dynamically reconfigurable architecture. The real contribution of this paper is to pose this classification as an initial reference to the novel designers, helping them to take some decisions according to the multimedia application that they want to implement.

According to the system-level architecture are defined three different categories:

1. Array of functional units (FUs): These architectures are created from chains of interconnected *processing elements* (PEs), which are replicated to create more complex structures, such as 2D-arrays of PEs. This kind of structure is totally dependent on some external resources. For example, it is managed and controlled by a host processor situated off-the-chip, which also supervises the reconfiguration process. Its data or configuration contexts are also stored on external memories. Regarding to this, it is very important to design an efficient communication channel between the host and the array of FUs, in order to accelerate data transfers among them. Consequently, an array of functional units is a useful and flexible system when the running application meets two important conditions: it is very intensive in computation and, simultaneously, it is quite independent of the processor without constant interruptions to the processor.
2. Coprocessor or hybrid architectures: The design complexity is something higher than on the previous category, because the constraints are higher too. The whole system is formed by two main parts: a processor core, which is usually a RISC or a VLIW processor, and a reconfigurable hardware core. Both cores are often tightly coupled and the control of the whole system is handled and managed by the processor core. The communication protocol and the interconnection network among processor-reconfigurable hardware and memory-reconfigurable hardware are critical decisions, since a bad selection could determine a slow data transfer. In this case some internal memories are incorporated to the architectural structure. Reconfigurable hardware core has associated some configuration caches, while the data are stored into SRAM-memories. Generally, these structures allow certain level of autonomy, operating without an external intervention.

3. Array of processors: These systems are formed by a set of hard or soft interconnected processors. The power of this architectural structure is obtained at cost of an important increment of the design complexity. All these processors may operate independently in parallel one to the other, executing different tasks. Although they also may combine their resources to execute the same functions. The major difficulties associated to these systems are: to manage the data dependences among processors running in parallel and assigning the instructions to its proper processor. It is difficult to design these architectural structures because there are a lot of variables implicated into the correct execution of the processors. This architectural structure is useful for implementing parallel applications with low data dependences.

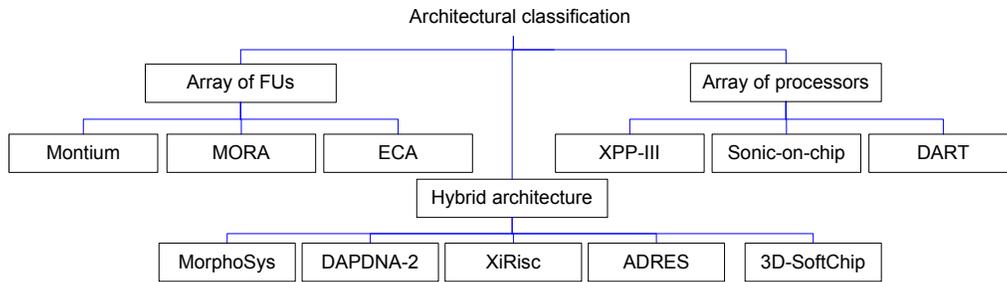


Fig. 3 Architectural classification

The result, of applying this classification to the 11 architectures selected above, is depicted on Figure 3. All these architectures are dynamically reconfigurable, and the majority has been grouped as hybrid architectures.

4. ARCHITECTURAL DESCRIPTION

4.1 Array of functional units

4.1.1 Montium architecture

Montium is a reconfigurable *tile processor* (TP) controlled by a communication and configuration unit (CCU). Typically it is used as an accelerator core in combination with a general purpose processor.

Montium TP is a 1D-array formed by 5 Processing Parts (PP). Each one has a special ALU, which is connected with two local memories with their respective local address generators, using a local bus. In addition to this, each ALU input is associated to a 4x16-bit register file [37].

Five PPs are fully connected with each other and also with the CCU though global busses. CCU unit manages data transfers from/to the external tile processor.

Another important element of this system is the sequencer, which is the responsible of the reconfiguration process, although always under CCU control. This procedure is complex because the sequencer decodes the configuration instruction, decomposing and organizing it according to the type of element (ALU, memory, connection, register file), and among them also depending on the specific one (ALU 1, ALU 2, etc).

4.1.2 MORA architecture

Multimedia Oriented Reconfigurable Array is a coarse-grain dynamically reconfigurable architecture, ordered like 2D-mesh topology, divided in 4 quadrants with 4 identical reconfigurable cells (RCs) each one.

Interconnection is created as a hierarchical reconfigurable network organized on two levels: the first one communicates the intra-quadrant RCs using horizontal and vertical 8-bit buses configuring a mesh topology. The second level connects its four quadrants among them via unidirectional buses and programmable bus switches [38]. Only the quadrants peripheral cells of can access to global busses.

4.1.3 ECA architecture

Elemental Computing Array (ECA) is a commercial chip developed in 2007 by *Element CXI* [39], which contains 7 different types of elements grouped in 5 computation elements, one memory element, and one signaling element. All these heterogeneous elements are grouped hierarchically into zones (that is a set of 4 elements), clusters (4 zones grouped), super-clusters, and so on.

This structure has been designed to be scalable and flexible. However, nowadays its scalability is limited to four clusters (64 elements), although in a early future the number of clusters could grow.

Interconnection network has a hierarchically structure. Connectivity among elements located into the same zone is done using a Cross Point Switch (CPS). On the other hand, different zones are connected using Through Queues (TQs) which are similar to FIFO memories. Finally, clusters communications are determined by a hierarchical bus structure. It is important to remark that communication protocol is packet-based.

Dynamic partial reconfiguration allows configure each element independently in one clock cycle, but also it is possible to reconfigure several elements simultaneously.

4.2 Hybrid architectures

4.2.1 MorphoSys architecture

The most recent contribution to this research project is the architecture named M2 [40], which is an improved version of the M1 architecture. M1[41] is structured in several elements: a 32-bit Tiny RISC processor, an 8x8 coarse-grain array of reconfigurable cells (RCs) and a frame buffer (FB). M2 introduces significant improvements with respect to M1, being the main ones [42]:

Tiny RISC processor modifications: Core processor allows the simultaneous execution of the RC Array, it incorporates subroutines, an interrupt unit. It also includes new instructions over the initial instruction set.

RC Array: In this case, MAC unit is pipelined with shorter critical-path. Now, each RC incorporates a local SRAM accelerating the data capturing, moreover, it is possible to idle some RCs,

Frame buffer: it has accelerated the data transfers between RC array and FB.

In this structure dynamic reconfiguration is done by using a double context memory in collaboration with the frame buffer, and it may affect to the functionality of each RC as well as the interconnections between the different cells. This procedure is specified by the processor through the DMA, and it is done without interrupting RC array operation by loading the context data into a non-active parts of context memory.

4.2.2 DAPDNA-2 architecture

Digital Application Processor (DAP) on Distributed Network Architecture (DNA) was presented in 2003 by IPFlex as a commercial chip [43]. It is composed by a 32-bit RISC processor and a coarse-grain dynamically reconfigurable array interconnected through a high speed bus switch.

DNA is a 2D-array of 376 processing elements (PEs) organized in 6 segments. Every PE is associated to two independent 32-bit buses, one for input and one for output data. These elements connect each PE with all the PEs located in the same segment.

Regarding to its dynamic reconfiguration methodology, it contains a configuration memory which stores multi-context configurations (four contexts), although additional banks can be loaded from the external memory on processor demand. One of these banks remains foreground while the other three are background, until the processor is signaling a new reconfiguration.

4.2.3 XiRisc architecture

XiRISC architecture was developed in 2004 as a hybrid architecture formed by a 32-bit RISC processor and a 2D fine-grain reconfigurable functional unit, PiCoGA (Pipelined Reconfigurable Gate Array), with two 32-bit pipelined data-

paths. Processor and PiCoGA can operate concurrently, although it is the processor who controls the activation and the data transfer to the PiCoGA. This reconfigurable array has 24 rows, each one with 16 reconfigurable cells (RCs).

Dynamic full or partial reconfiguration is activated by special instructions, and controlled by the XiRISC core.

Based on XiRisc architecture, at the end of 2004, surged XiSys [44]. The most significant difference between both systems is the inclusion of a reconfigurable I/O unit based on FPGA (eFPGA).

In 2005 the same research group developed XiMP [45], a dual processor architecture composed by a VLIW core and a XiRISC core, with an eFPGA (reconfigurable I/O unit), a DMA, two banks of memory, and a multi-layer AHB Bus.

4.2.4 ADRES architecture

Architecture for Dynamically Reconfigurable Embedded Systems [46] is a flexible template that allows parameterization of a set of architectural features. It is constituted by a VLIW processor and a coarse-grained reconfigurable array (CGRA). The VLIW processor controls CGRA configurations and operations, although they share some resources, such as data register file and special functional units. In this architecture the CGRA acts like a coprocessor.

4.2.5 3D-SoftChip architecture

This chip is composed by a coarse-grain dynamically configurable array processor (CAP) of heterogeneous processing elements (grouped on 4 quadrants), an intelligent configurable switch (ICS) and an Indium bump interconnection array (IBIA).

This architecture permits programming multiple computational models (SISD, SIMD, and MISD).

Interconnections are organized hierarchically, where the PEs within the same quadrant form a mesh topology, in the sense that each PE is connected with its direct horizontal and vertical neighbors. On the next level of connectivity is a switch block array communicating all quadrants. Finally the Indium bump interconnection array communicates the processor with the CAP.

Dynamic fully or partially reconfiguration is controlled by the ICS chip. Every PE contains a small local SRAM memory which store configuration contexts in order to reduce the reconfiguration overhead

4.3 Architectural structure based on an array of processors

4.3.1 Ultrasonic architecture

Ultrasonic is based on Sonic-on-chip [47] architecture. In both cases the objective is to create a flexible and scalable architecture at different levels. For example, at system level it allows the use of several video coding standards, at application level it supports different applications and at processor level it is capable of storing various contexts for an application.

The basic computational elements are denoted as PIPEs (Pipelined Processing Elements), connected by buses. These PIPEs are subdivided into smaller units: PIPEs engine is the main processing element that executes the computation, PIPE router is a routing element that controls the input/output data and the transferring from/to the external chip, PIPEbus interconnects PIPEs data from/to the host processor and PIPEflow allows synchronous transfer of video between PIPEs. Reconfiguration is controlled through software plug-ins.

4.3.2 XPP-III architecture

PACT Corp. commercialized its first chip, XPP (eXtreme Processing Platform) [48], in 2002. Nowadays XPP-III is replacing it, being an improved version available on the market in three sizes in function of the requirements of the multimedia application to handle.

XPP-III chip is constituted by several sequential processors kernels (FNC-PAEs) connected with a hierarchical coarse-grain reconfigurable array (XPP-Array), which elements (ALU-PAE, RAM-PAE) are tightly coupled with a self-synchronizing network and crossbars.

PAEs (Processing Array Elements) integrate horizontal routing buses for point to point connections between XPP objects, and their communication protocol is data packet oriented.

4.3.3 DART architecture

The architecture is formed by a task controller unit, a group of clusters (processing elements), a shared data memory, a configuration memory, an instruction memory and an I/O unit. The power of this structure resides on its clusters, since they are in charge of executing the applications [49].

These clusters can work independently, and also together. In addition to this, one or more cluster can act as a hardware accelerator. They are formed by 6 Reconfigurable DataPaths (RDP), interconnected each other through a segmented network. A memory controller, a data memory, a configuration memory, a DMA Controller and a FPGA complete the units included in a cluster. The principal manager into this substructure is a cluster controller.

Dynamic reconfiguration is managed by a cluster controller. It only can handle one RDP configuration per cycle which is done using special instructions. The cluster's controller handles the serial FPGA reconfiguration, specifying an address to DMA controller where the serial configuration information is. The DMA controller then manages the data transfer to the FPGA reconfiguration memory.

5. DISCUSSION

Real-time multimedia applications demand systems capable of achieving a balance between flexibility and efficiency. As it has been exposed along this paper, the dynamically reconfigurable architectures are the best choice to cover these requirements, since they allow obtaining a better global performance than ASICs and GPPs.

These architectures should be organized according to common criteria, in order to create design's references for helping to the novel developers. For coping this objective, previous sections have reviewed, classified and described the most recent dynamically reconfigurable architectures within this last decade. Finally, this section presents some benefits and drawbacks associated to each one of the established categories.

Array of FUs: these systems are totally dependent of an external host processor, and they behave like normal peripherals of the processor. This situation involves a high communication cost, due to the slow data transfers. However, these structures are useful for implementing applications with a high computational cost and a low dependence with the processor.

Hybrid architecture: the benefit that is introduced having an inherent processor as part of the system affect at the final performance of the system in terms of power consumption. As a consequence, this kind of architectures has higher energy waste than the arrays of FUs, because the first one has to assume the processor execution cost. Moreover, decoding instructions process and their memory readings are responsible of most power consumption into the whole system.

Array of processors: they are the faster systems in comparison with the previous ones. This structure allows the parallel execution of its independent processors. These systems require a high level of power consumption and also an important number of clock cycles to decoding the instructions.

Multimedia applications demand a high data manipulation, and this situation often generates bottlenecks into the memory accesses. Regarding to this, a designer has to pay special attention when selecting the memory size, its number of write or read ports, its location and its controller.

6. CONCLUSIONS

Reconfigurable architectures represent an excellent alternative in order to meet flexibility and efficiency into the same piece of hardware. For this reason, several research groups have designed very different reconfigurable architectures in order to efficiently implement multimedia applications onto them, generating an enormous amount of papers during the last years. However, the authors do not clearly identify the pros and the cons of the architecture proposed when compared with other state-of-the-art works, and thus, they do not provide a very valuable information for future designers.

This paper has proposed a classification for multimedia-based reconfigurable architectures attending to a system-level criterion. Although the process followed in this paper is suitable for any reconfigurable architecture, this work has focused on coarse and fine-grained dynamically reconfigurable circuits that have proven their efficiency for video processing applications. As a result, 11 of the most significant works published within this research field have been grouped in three different categories, uncovering common advantages and disadvantages between them.

REFERENCES

- [1] Garcia, P., Compton, K., Schulte, M., Blem, E., and Fu, W., "An Overview of Reconfigurable Hardware in Embedded Systems," *EURASIP Journal on Embedded Systems*, 1-19 (2006).
- [2] Hartenstein, R., "A decade of reconfigurable computing: a visionary retrospective," *Proceedings of International conference on design automation and testing in Europe and exhibit (DATE)*, (2000).
- [3] Tessier, R. and Burleson, W., "Reconfigurable Computing Signal Processing: A Survey," *Journal of VLSI Signal Processing*, 28, 213-235 (2005).
- [4] Shoa, A., Shirani, S., "Run-time reconfigurable systems for digital signal processing applications: a survey," *Journal of VLSI Signal Processing*, 39, 213-235 (2005).
- [5] Todman, T., Constantinides, G., Wilton, S., Mencer, O., Luk, W., and Cheung, P., "Reconfigurable computing: architectures and design methods," *IEEE Proceedings - Computing on Digital Technology*, 152(2), (2005).
- [6] Compton, K., and Hauck, S., "Reconfigurable computing: a survey of systems and software," *ACM Computing Surveys*, 34(2), 171-210 (2002).
- [7] Campi, F., and Mucci, C., "Run-Time reconfigurable processors," J. Nurmi (ed.), *Processor Design: System-on-chip computing for ASICs and FPGAs*, Springer, 177-208 (2007).
- [8] Abdin, Z., and Svensson, B., "Evolution in Architectures and Programming Methodologies of Coarse-grained reconfigurable computing," *Embedded Hardware Design, Journal of Microprocessors and Microsystems*, (2007).
- [9] Xilinx Corporation, www.xilinx.com (2009).
- [10] Altera Corporation, www.altera.com (2009).
- [11] Atmel Corporation, www.atmel.com (2009).
- [12] Lattice Semiconductor Corporation, www.latticesemi.com (2009).
- [13] Hauck, S., "The Roles of FPGAs in Reprogrammable systems," *Proceedings Of the IEEE*, 86(4), 615-638 (1998).
- [14] Ebeling, C., Cronquist, D., Franklin, P., Secosky, J., and Berg, S., "Mapping applications to the RaPiD configurable architecture," *IEEE Symposium on FPGAs for custom computing machines*, 106-115 (1997).
- [15] Mirsky, E., and DeHon, A., "MATRIX: A reconfigurable computing architecture with configurable instruction distribution and deployable resources," *Proceedings Of IEEE FCCM*, (1996).
- [16] [raw] Bedford, M., Lee, W., Miller, J., Wentzlaff, D., Bratt, I., Greenwald, B., Hoffmann, H., Johnson, P., Kim, J., Psota, J., Saraf, A., Shnidman, N., Strumpfen, V., Frank, M., Amarasinghe, S., and Agarwal, A., "Evaluation of the RAW microprocessor: an exposed-wire-delay architecture for ILP and Streams," *Proceedings Of 31st ISCA*, 2-13 (2004).
- [17] Schmit, H., Cadambi, S., and Moe, M., "Pipeline Reconfigurable FPGAs," *Journal of VLSI signal processing systems*, 24, 129-146 (2000).
- [18] Singh, H., Lee, M., Lu, G., Kurdahi, F., and Bagherzadeh, N., "MorphoSys: a reconfigurable architecture for multimedia applications," *Proceedings Of the XI Brazilian Symposium on Integrated Circuit Design*, 134-139 (1998).

- [19] Baumgarte, V., Ehlers, G., May, F., Nüchel, A., Vorbach, M., and Weinhardt, M., "PACT XPP- A self-reconfigurable data processing architecture," *Journal of Supercomputing*, 26, 167-184 (2003).
- [20] Becker, J., and Glesner, M., "A parallel dynamically reconfigurable architecture designed for flexible application-tailored hardware/software systems in future mobile communication," *Journal of supercomputing*, 19, 105-127 (2001).
- [21] Rath, K., Tangirala, S., Friel, P., Balsara, P., Flores, J., and Wadley, J., "Reconfigurable Array Media Processor (RAMP)," *IEEE Symposium on Field-Programmable Custom Computing Machines*, 287-288 (2000).
- [22] Haynes, S., Epsom, H., Cooper, R., and McAlpine, P., "UltraSONIC: A reconfigurable architecture for video image processing," *Field-Programmable logic and applications (FPL02)*, 25-45 (2002).
- [23] IPFlex, <http://www.ipflex.com> (2009).
- [24] David, R., Chillet, D., Pillement, S., and Sentieys, O., "DART: a dynamically reconfigurable architecture dealing with future mobile telecommunications constraints," *International Parallel and distributed Processing Symposium (IPDPS)*, (2002).
- [25] Heysters, P., Smit, G., and Molenkamp, E., "A Flexible and Energy-Efficient Coarse-Grained Reconfigurable Architecture for Mobile Systems," *The Journal of Supercomputing*, 26, 283-308 (2003).
- [26] Toi, T., Nakamura, N., Kato, Y., Awashima, T., and Wakabayashi, K., "High-level synthesis challenges and solutions for a dynamically reconfigurable processor," *Proceedings Of ICCAD*, 702-708 (2006).
- [27] Vassiliadis, S., Wong, S., Gaydadjiev, G. N., Bertels, K.L.M., Kuzmanov, G. K., and Moscu Panainte, E., "The Molen Polymorphic Processor," *IEEE Transactions on Computers*, 53(11), 1363- 1375 (2004).
- [28] QuickSilver Technology, <http://www.qstech.com/products.htm> (2009).
- [29] Mei, B., Lambrechts, A., Mignolet, J., Verkest, D., and Lauwereins, R., "Architecture exploration for a reconfigurable architecture template," *Configurable Computing: Fabrics and Systems*, 90-101 (2005).
- [30] La Rosa, A., Lavagno, L., and Passerone, C., "Software Development for high-performance, reconfigurable, embedded multimedia Systems," *IEEE Design & Test of computers – Embedded systems for real-time multimedia*, 28-38 (2005).
- [31] Kim, C., Rassay, A., Lachowicz, S., Lee, M., and Eshraghian, K., "3D-ShoftChip: A novel architecture for next-generation adaptive computing systems," *EURASIP Journal on applied signal processing*, 1-13 (2006).
- [32] Heitchecker, S., Carmo, A., and Ernst, R., "A high-end real-time digital film processing reconfigurable platform," *Journal on Embedded Systems*, (2007).
- [33] Master, P., "Reconfigurable hardware and software architectural constructs for the enablement of resilient computing systems," *Application-specific Systems Architectures and Processors (ASAP'06)*, IEEE Computer Society, (2006).
- [34] Shukla, S., Bergmann, N., and Becker, J., "QUKU: a two-level reconfigurable architecture," *Proceedings Of the emerging VLSI tech. and architectures*, (2006).
- [35] Lanuzza, M., Perri, S., and Cornosello, P., "MORA: A new coarse-grain reconfigurable array for high throughput multimedia processing," Ed. Springer, book "Embedded computer systems: architectures, modelling, and simulation," 159-168 (2007).
- [36] Brunelli, C., Garzia, F., and Nurmi, J., "A coarse-grain reconfigurable architecture for multimedia applications featuring subword computation capabilities," *Special Issue, Real-time image Proceedings*, 21-32 (2008).
- [37] Smit, G., Kokkeler, A., Wolkotte, P., and Hölzenspies, P., "The Chameleo Architecture for Streaming DSP Applications," *EURASIP Journal on Embedded Systems*, (2007).
- [38] Lanuzza, M., Perri, S., Corsonello, P., and Margala, M., "A new reconfigurable coarse-grain architecture for multimedia applications", *Adaptive Hardware and Systems (AHS'07)*, 119-126 (2007)P. Garcia, K. Compton, M.

- Schulte, E. Blem, W. Fu, "An Overview of Reconfigurable Hardware in Embedded Systems", EURASIP Journal on Embedded Systems, vol 2006, pp. 1-19.
- [39] Kelem, S., Box, B., Wasson, S., Plunkett, R., Hassoun, J., and Philips, C., "An elemental computing architecture for SD radio," Software Defined Radio Forum Technical Conference, (2007).
- [40] Koohi, A., Bagherzadeh, N., and Pan, C., "A fast parallel reed-solomon decoder on a reconfigurable architecture," First IEEE/ACM/IFIP International conference on Hardware/Software codesign and system synthesis, 59-64 (2003).
- [41] Parizi, H., Niktash, A., Kamalizad, A., and Bagherzadeh, N., "A reconfigurable architecture for wireless communication systems," Proceedings of the Third International conference on Information Technology: New Generations (ITNG'06), 250-255 (2006).
- [42] Pan, C., Bagherzadeh, N., Hosein, A., and Koohi, A., "Design and analysis of a programmable single-chip architecture for DVB-T base-band receiver," Proceedings of Design Automation and Test in Europe conference, 468-473 (2003).
- [43] Sato, T., Watanabe, H., and Shiba, K., "Implementation of dynamically reconfigurable processor," International symposium on VLSI Design Automation and Test in Europe, 323-324 (2005).
- [44] Bocchi, M., De Bartolomeis, C., Mucci, C., Campi, F., Lodi, A., Toma, M., Canegallo, R., and Guerrieri, R., "A XiRisc-based SoC for embedded DSP Applications," IEEE Custom, integrated circuits conference.
- [45] Lodi, A., Bocchi, M., Mucci, C., Deledda, A., Campi, F., Toma, M., Dominicis, M., and Guerrieri, R., "Design and implementation of a reconfigurable heterogeneous multiprocessor SoC," IEEE Custom Integrated Circuits Conference, 93-96 (2006).
- [46] Mei, B., De Sutter, B., Vander AA, T., Wouters, M., DuPont, S., and Kanstein, A., "Implementation of a coarse-grained reconfigurable media processor for AVC decoder," Journal of signal processing systems, 51, 225-243 (2008).
- [47] Haynes, S., Stone, J., Cheung, P., and Luk, W., "Video Image Processing with the Sonic Architecture," Configurable computing, 50-57 (2000).
- [48] Pact Corporation, <http://www.pactxpp.com> (2009).
- [49] Pillement, S., Sentieys, O., and David, R., "DART: A functional-level reconfigurable architecture for a High energy efficiency," EURASIP Journal on Embedded Systems, (2008)