Ciencia y Cultura
ISSN: 2077-3323
cultura@ucb.edu.bo
Universidad Católica Boliviana San Pablo
Bolivia

Calderón, Humberto
Next Generation of Smart Machines: a survey of enabling technologies
Ciencia y Cultura, núm. 32, junio, 2014, pp. 89-119
Universidad Católica Boliviana San Pablo
La Paz, Bolivia

Ciencia y Cultura Nº 32 ISSN: 2077-3323 Junio 2014 89-119

Next Generation of Smart Machines: a survey of enabling technologies

La próxima generación de máquinas inteligentes: una revisión de las tecnologías de soporte

Humberto Calderón*

Abstract:

Last century had an accelerated pace in terms of the technological revolution, a different world has been built; one based on electronic devices. Currently, we inhabit a world in which several applications use a computer inside another device, realizing control, communication and multimedia processing tasks; these constitute the largest type of computers used in the market (embedded systems). Since the ENIAC days, computers have become smaller in size, cheaper in price, with lower power consumption and increased performance (more instructions per cycle), therefore, a new era of digital systems is emerging imposed by military, industrial, communications and entertainment needs. Our lives are surrounded by more and more intelligent multimedia environments such as in the workplace, home and on entertainment; handheld devices (cellular phones, PDAs smartphones, tablets, game consoles, etc.) are some examples of embedded systems that we use daily.

Recently, reconfigurable computing technologies are enabling the creation of computational platforms supporting new and smarter human-machine

* Laboratorio de Ingeniería en Computación Universidad Católica Boliviana.
Contacto: hcalderon@ucb.edu.bo Home Page: www.ce.ucb.edu.bo

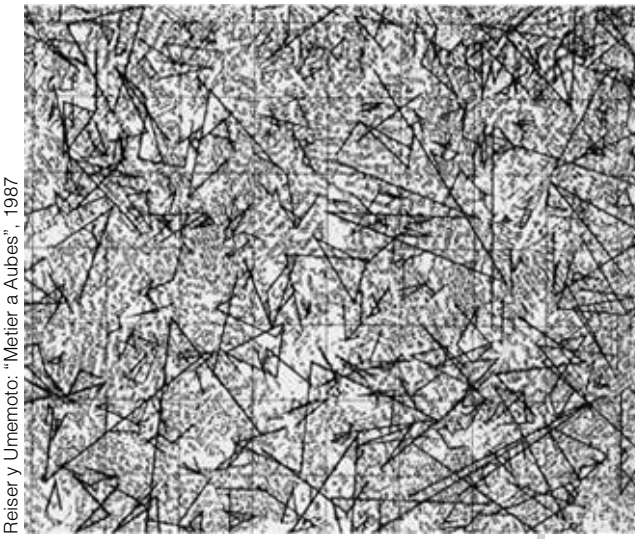
collaborative environments with cheaper non-recurrent engineering costs. Clever machines with highly interactive features go beyond traditional systems; they are rich in new I/O methods (*e.g.* haptic). Those new cooperative environments require intelligent systems in which the system entities possess sensory-motor capabilities, such as visual, auditory and tactile features, as a part of their cognitive capabilities. Given the current technological maturity, innovative and intelligent products that are capable of problem solving, reasoning and learning are becoming possible. This paper reviews technology advancements that could enable the research and development of new intelligent systems in developing countries. The main features of reconfigurable computing as an enabling technology for system development are presented. Furthermore, we introduce the point of view of the Computer Engineering Laboratory (L-IC) at the Bolivian Catholic University and its preliminary architecture for human-machine collaborative environments that will be able to support a wide range of cognitive behaviors within real time constraints.

Keywords: Artificial Intelligence, Soft Computing, Reconfigurable Computing, Case-based Reasoning, Fuzzy Logic, Computer Arithmetic.

Resumen:

El siglo pasado tuvo un acelerado ritmo de revolución tecnológica. Se construyó un mundo diferente, basado en el uso de máquinas con dispositivos electrónicos. En la actualidad, habitamos un mundo en el que varias aplicaciones utilizan un procesador electrónico dentro de otro dispositivo, realizando tareas de control, de comunicación y de procesamiento multimedia. Estos procesadores constituyen el tipo más grande de equipos utilizados hoy en día en el mercado (sistemas incrustados). Desde los días de la ENIAC, las computadoras son de menor tamaño, son más baratas en precio, consumen menor energía y poseen un mayor rendimiento (más instrucciones por ciclo), etc.; por lo tanto, una nueva era de sistemas digitales emerge impuesta por necesidades militares, industriales, comunicacionales y de entretenimiento.

Nuestras vidas están rodeadas de entornos multimedia con mayores capacidades inteligentes en el trabajo, hogar y entretenimiento; los dispositivos móviles (teléfonos celulares, PDAs smartphones, tablets, consolas de juegos, etc.) son algunos ejemplos de sistemas incrustados que usamos diariamente. Recientemente, las tecnologías de computación reconfigurables están permitiendo la creación de plataformas computacionales que apoyan a los nuevos entornos inteligentes hombre-máquina; éstas son desarrolladas con costos de ingeniería no recurrentes más bajos que sus predecesores.



Reiser y Umemoto: "Metier a Aubes", 1987

Las máquinas inteligentes poseen características de alta interactividad y van más allá de los sistemas tradicionales; son ricos en nuevos métodos de I/O (e.g. hápticos). Esos nuevos entornos cooperativos requieren sistemas inteligentes en los que las entidades del sistema poseen capacidades sensoriales-motoras, tales como las características visuales, auditivas y táctiles, como parte de sus capacidades cognitivas. Dada la madurez tecnológica actual, es posible crear productos innovadores e inteligentes que sean capaces de resolver problemas y realizar razonamientos y aprendizajes en tiempo real. Este artículo revisa los avances tecnológicos que podrían permitir la investigación y desarrollo de nuevos sistemas inteligentes en los países en desarrollo. Se presentan las principales características de la computación reconfigurable, mostrándola como una tecnología habilitadora para el desarrollo de sistemas inteligentes. Además, presentamos el punto de vista del Laboratorio de Ingeniería en Computación (L-IC) de la Universidad Católica Boliviana y se presenta una arquitectura preliminar para entornos colaborativos hombre-máquina que será capaz de soportar una amplia gama de comportamientos cognitivos operando con limitaciones de tiempo real.

Palabras clave: Artificial Intelligence, Soft Computing, Reconfigurable Computing, Case-based Reasoning, Fuzzy Logic, Computer Arithmetic.

1. Introduction

The electronic computer era started towards the end of 30s in the past century; it has been almost 85 years since the beginning of John V. Atanasoff's works at Iowa University. The Atanasoff Berry Computer (Kloet & van Tilburg, 2006), led the creation of the ENIAC, that is widely known as the first general purpose electronic computer. Since then, computers have increased their level of parallelism, primarily via the decentralizing of control/execute tasks that were achieved in centralized way in the first generation via the Central Processing Unit (CPU) (Hayes, 1988; Hwang, 1992) The second generation introduced

I/O processors (predecessor of DMA machines), since then, this decentralization tendency has increased in later generations with more customized units that collaborate the CPU in different tasks (Hwang, 1992; Elliott, 2010). The increasing processing needs, principally guided by a market hunger for new multimedia intelligent systems has led the creation of new architectures that are rich in Instruction Set Architectures for highly customized tasks (Lempel, Peleg and Weiser, 1997; Clark, Tang and Mahlke, 2002). Technology advancement to support actual needs includes the widespread use of both time and space parallelism (Kastrup, 2001). Those techniques have increased the number of instructions issued and executed in a machine cycle (performance), all this accompanied by the introduction of new memory hierarchies that try to avoid the memory wall problem (Hennessy and Patterson, 2007). The aforementioned innovations were catalyzed by the advances in technology (processing substrate), principally due to the introduction of the transistor (see Table Nº 1.) and its miniaturizations that perhaps is reaching its natural limit (Chie and Karamcheti, 2013; Palem and Ligamneni, 2012). Currently we can build computers that consume less power per instruction, have smaller sizes and can be built in a 5/285000 of ENIAC size (factor of 1e-5), fact that lead us to use integrated computers everywhere. The following Table presents a summary of the advances in the electronic era since the introduction of the first computers (first generation) to the time in which, the embedded systems are widely used.

Table 1. Computer Evolution: Main Features of the first four generations

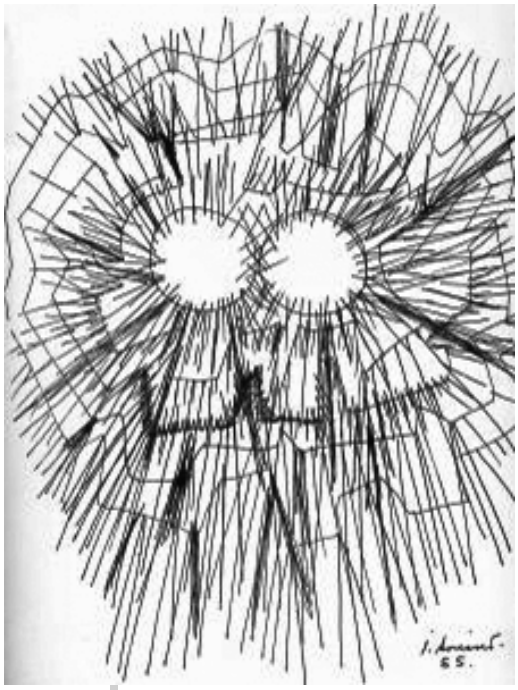
	First Generation (1938-1954)	**Second Generation (1955-1964)**	**Third Generation (1965-1974)**	**Fourth Generation (1975-)**
Control	Centralized y the CPU	Separate input-output processors	Separate input-output processors	Multiple oriented processors and coprocessor CISCs and RISCs
Parallelism	none	Level of processors decentralization of CPU Late basic time and space parallelism	Time and Space parallelism (e.g. Microprogramming, parallel processing, pipelining)	Widespread use of Time and Space parallelism within the die
Arithmetic	Basic, Fixed Point Arithmetic	Fixed and Floating Point arithmetic with logic operations	Multiple Fixed and Floating Point arithmetic units with logic operations	Multiple Fixed and Floating Point arithmetic units with trigonometric support
Base Technology	Vacuum Tubes	Transistor Based	Integrated circuit based	VLSI Microprocessor based

	First Generation (1938-1954)	Second Generation (1955-1964)	Third Generation (1965-1974)	Fourth Generation (1975-)
Memory	Mercury delay line memories and Electrostatic memories Punched cards	Punched cards Ferrite core drums	Semiconductor memories (RAM & ROM) Disk Storage	Floppy disks, massive use of hard disks
Memory management	none	Basic monitors for batch processing	MMU for virtual memories, etc. CACHE	Integrated Caches and virtual memories supports
I/O devices (machine interfaces)	none	I/O Processors	Printers	CRTs, LCDs
Language	Machine Language Assembler	ALGOL, COBOL and FORTRAN Compilers and basic subroutines	FORTRAN-II TO IV, COBOL, PASCAL PL/1, BASIC, ALGOL-68, etc.	C and C++, DBASE, SMALLTALK, etc.
Operating System	none	Batch processing and Multiprogramming Operating system used	Remote processing, Time-sharing, Real-time, Multi-programming Operating System	Mature multitasking Operating Systems
Used for	Scientific computations	Business, industry and commercial organizations	business, industry and commercial organizations	business, industry and commercial organizations and Home

Source: Hayes (1988)

The embedded system era, started with the creation of microcontrollers (Intel, 1978; 1994; Motorola), devices that are small microprocessor with an ISA capable to support bit manipulations and posses memory resource in the same die. Later more complex microcontrollers were introduced (Motorola, 1992) that anticipated the whole functionality of today denominated System on a Chip (SoC). Those systems introduce oriented ISA towards the signal processing market (trying to support the emerging needs of multimedia applications[1]). Nowadays embedded systems are built with improved microcontrollers that are able of supporting media processing (ARM; Diefendorff, Dubey, Hochsprung and Scales, 2000; Eden and Kagan, 1997). They incorporate features that in the past were inside a microprocessor`s domain (e.g. multilevel cache, several cores, etc.), intended to support multimedia processing in a new paradigm that introduces more intelligent features in the machines (*e.g.* smart phones [arm]). The main processing task in multimedia processing actually is vision. The last is true due to the processor`s power consumption to support

1 Multimedia is media content that uses a combination of different content forms. Multimedia includes a combination of text, audio, still images, animation, video and interactivity content forms.



Juan Soriano

pixel processing (stream processing) (ARM; Dobai and Sekanina, 2013), the need of customized hardware, and due to the fact that vision is the most important feature for making machine reasoning. Table Nº 1 presented above, does not mention the fifth generation, and it seems still unclear what the beginning of this generation and the main features that characterize it will be. Probably fifth generation will be the one that supports actively cognition tasks: a generation that is constituted by specialized processors that surpass the Von Newman processing paradigm, processors that can be based on nature and biological observation (*e.g.* Bio-inspired systems) (Mange and Tomassini, 1998; Biologically) . The last becomes a need due to the need of smarter machines and the fact that we cannot sustain Moore's Law (Chie *et al*., 2013 Palem *et al*., 2012), for longer creating smart machines within the Von Newman paradigm. All those facts are guide us to investigate new architectures and micro architectures that support activities that day to day lives require, that support more clever machines, ones that can be able to adapt to our needs in a gentle and cooperative way. Industry and military investments have shown the importance of this topic nowadays (IBM; Painkras, Plana, Garside, Temple, Davidson, Pepper, Clark, Patterson and Furber, 2012).

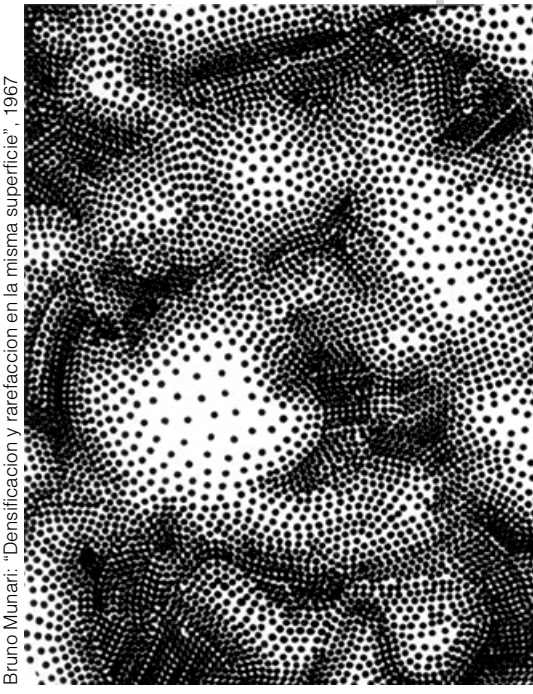
Artificial general intelligence (AGI) research has been delivered since 1952 meeting in Dartmouth College (McCarthy, Minsky, Rochester and Shannon, 1955), and a lot of expertise has been created through the development of symbolic and sub- symbolic AI. Nowadays we want to revisit both paradigms and complement those approaches with highly customized systems, with the aim of creating smart systems to support applications in our everyday life; we are not researching for AGI. Systems that can support processing of cognitive activities should be created, and Soft Computing techniques could allow us to merge symbolic and sub-symbolic AI within customized processing units. In the Computer Engineering Laboratory at the UCB-SP we strongly believe that if we can provide highly customized hardware for Soft Computing, we will be able to reproduce intelligent behaviors within real time constraints.

The L-IC approach

The future includes Human-Machine Collaborative (HMC) environments everywhere, L-IC researches for creating tools (Smart Machines) that can enable those environments using soft computing, including machine-learning (statistical processing) and intelligent signal processing (Hager, Okamura and Taylor, 2006), all these augmented with the revisiting of old paradigms (*e.g.* Case based Reasoning). Indeed some authors have proposed human communication, including both verbal and nonverbal behaviors, which use previously acquired knowledge, as a key aspect when designing a system with human-like cognitive abilities. One such approach was described by Schank (1999):

> For thousands, maybe millions of years, people have been telling stories to each other. They have told stories around the campfire, they have traveled from town to town telling stories to relate news of the day, they have told stories transmitted by electronic means to passive audiences incapable to doing anything but listening (and watching). Whatever that means, and whatever the venue, storytelling seems to play a major role in human interaction. We get reminded of histories and we use them in conversations.

In the light of the Shanck approach, it seems promising to investigate artificial ways of classifying new stories, and then observe the interaction between people and machines using such stories (*i.e.*, cases) for reasoning. Practically, this can be achieved using well known methodologies such as memory based reasoning or reasoning by analogy (Schank, 1999). It is foreseen that these cognitive metodologies could be used to create artificial intelligence to support humans in various activities at home, work, or for simple entertainment.



Bruno Munari: "Densificacion y rarefaccion en la misma superficie", 1967

The last decade has witnessed the rise of several HMC systems in diverse fields ranging from game applications (Benford, Magerkurth and Ljungstrand, 2005) to highly sophisticated military ones (Murray, 2000; Xiao, Lina and Junwei, 2009). These new technological advances enhance the processing capabilities of current compu-

ter systems incorporate existing sensory-motor[2] actions that try to emulate the natural behavior of human beings. Nowadays, last decade's traditional applications are reference points for the research and development (R&D) of new artificial intelligence artifacts covering a broad spectrum of applications, from Intelligent Companions[3] to sophisticated emerging applications such as Personal Orthotics[4]. Cognitive research[5] indicates that multisensory cognitive processes are fundamental for achieving these new R&D goals (Piefer and Bongard, 2006), particularly given the operational needs of Intelligent Entities (IE). These kinds of entities possess the highest degree of interaction with their surrounding environment, an environment that is rich in multiple-media stimuli. Consequently, from the point of view of processing support, due to the large amount of input/output signals that these machines must deal with in the HMC environments (*e.g.* visual, auditive, speech and haptic) (Vernon, Metta and Sandini, 2007), new processing platforms with hardware customization are necessary. For instance, a sensorial processing platform is needed to reduce the high cost of processor and memory use (i.e. multimedia processing). Current processing solutions try to speed up multimedia processing by augmenting the instructions of general purpose processors (GPPs) (Lempel, Peleg and Weiser, 1997; Thakkar and Huff, 1999) only, but these solutions do not yet provide the necessary acceleration (Talla, John and Burger, 2003).

Current research on the development of new processors with the goal of improving human-machine interfaces should consider high-performance and silicon efficient customized units, as well as memory hierarchies when targeting cognitive information processing. New developments in closely related fields, such as video, 3D graphics, pervasive gaming (Benford *et al.*, 2005; Kuzmanov, Gaydadjiev and Vassiliadis, 2003), intelligent control and robotics are facilitating the development of new HMC systems. New markets need to look ongoing R&D activities for machines with increased degrees of intelligence (Vernon *et al.*, 2007; Petriu, Whalen, Abielmona and Stewart, 2004). It is anticipated that a processing platform capable of handling cognitive applications will require the use of multiple-cores (Parkhurst, Darringer and Grundmann, 2006; Zhu, Sreedhar, Hu and Gao, 2007) and a large customized memory

2 Sensory-motor: are cognitive process based on sensory data processing that causes a motor activity.

3 Artificial companion refers to an embodied intelligent device that spends time with you; usually the companion has similar interests to your own and whose company you enjoy.

4 Orthotics is an allied health profession. The field is concerned with the design, development, fitting and manufacturing of orthotics, which are devices that support or correct musculoskeletal deformities and/or abnormalities of the human body.

5 Cognitive science is the interdisciplinary study of how information is represented and transformed in the brain. It includes disciplines like Psychology, Philosophy, Neurosciences Linguistic and Artificial Intelligence.

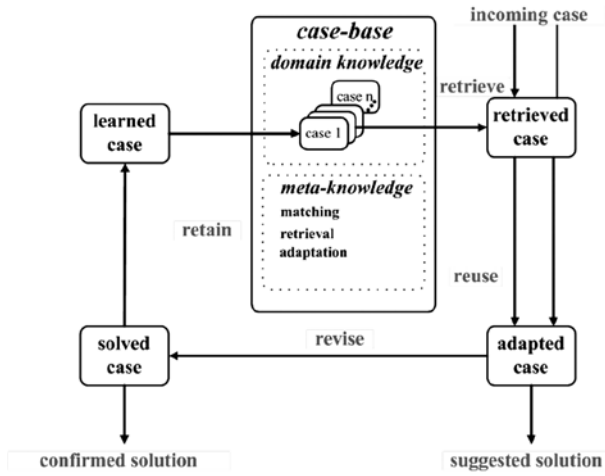
hierarchy. New processing demands will call for processors sets with greater flexibility and adaptability to meet rising multimedia and cognition processing needs.

The rest of this article is presented as follows. In Section 2, the background of the targeted cognitive techniques is reported. Section 3 deals with the main features that involve the cognitive information processing and the advantages of reconfigurable computing offers. Section 4 proposes the cognitive information processing platform. Section 5 concludes the article presenting directions for future work.

2. Background

Artificial general intelligence implements models and efforts proposed by the other members of cognitive sciences with the aim of reproducing intelligent processes in non biological strata. There are several open questions in the AI research community. In order to cope with these questions, researchers are trying to design machines that exhibit reasoning, planning, learning, communicating, environmental perception, as well as motor and object manipulation skills. Artificial general intelligence (or "true AI") has not yet been achieved and remains a long-term goal. Soft computing (SC) is a computer science term close related to AI that offers new insights into well known research dilemmas. SC techniques are often biologically-inspired paradigms, in contrast to traditional techniques based on formal representations of logical systems. Unlike hard computing, soft computing is tolerant to imprecision, uncertainty, partial truth, and approximation. The principal constituents of SC are Fuzzy Logic (FL), Neural Computing (NC) with their Artificial Neural Networks (ANN), Evolutionary Computation (EC), Machine Learning (ML) and Probabilistic Reasoning (PR). Some authors also include belief networks, chaos theory and learning theory. In the light of the work presented here, these techniques can be used to set up a Case-Based Reasoning (CBR) paradigm. Schank (1999) proposed CBR as a method for reasoning about new situations (cases) based on knowledge given from previous experiences (cases),

Figure 1. Case Based Reasoning Cycle



Implementing the CBR paradigm requires a model in which cases are stored in a memory (*e.g.*, "case memory") represented by "domain knowledge" as depicted in Figure Nº 1. CBR processing includes four operation phases which are outlined below:

- Retrieve Phase: When a new "incoming case" is presented to the system, experiences from the memory "domain knowledge" are retrieved based on judged similarities.
- Reuse Phase: The previous case solution is mapped to the new target problem. This may require adapting the existing solution as needed to fit the new situation. Sometimes the new solution is an integration of previous solutions. In other words, a "suggested solution" is used.
- Revise Phase: When solutions are applied, they are tested against the real world, revised as necessary and successful cases become "confirmed solutions".
- Retain Phase: After a solution has been successfully adapted to the target problem, it is stored as a new case in the memory. In this way a case is learned.

The retrieving of a group of "possible match" cases needs to be both effective and time efficient (low latency in case retrieval) whenever a new inquiry is presented to the system.

Traditional processors, like many presented in the literature (*e.g.*, Diefendorff *et al.*, 2000; Carlson, Castelino and Muellera, 1997; Kim, 2008), do not offer

the necessary capability to process multimedia. Moreover, such processors are less appropriated for processing cognitive information. Traditionally the Von Newman machines achieve results by incorporating some extensions to their instruction set architecture (ISAs) dealing with multimedia processing (Eden and Kagan, 1997) and even configuring the width of their data-paths adapting to several needs (*e.g.,* for processing 8 and 16-bit data (Lee and Stoodley, 1998). Nevertheless, the architectural improvements have several drawbacks when these processors are used for multimedia functionalities on the new frameworks (Bormans, Gelissen and Perkis, 2003), especially in scenarios of power aware computing (Benini, Bogliolo, Bogliolo and De Micheli, 2000) and embedded systems[6] like the ones used to set up intelligent systems.

Exploitation of the data parallelism that multimedia applications provide began with the creation of data-path micro-architectures with coarse grained characteristics (Wentzlaff, Griffin, Hoffmann, Bao, Edwards, Ramey, Mattina, Miao, Brown III and Agarwal, 2007). Performance improvements in multimedia processing (Zhu, Hu and Gao, 2007) are based on the space parallelism approach (*e.g.* more additions or multiplications units operating in parallel), as well as with the improvement in clock rate frequencies. Nevertheless, as parallelism, bandwidth (Burger, 1997) and clock rate increase, the inherent *memory wall* problem (Hennessy and Patterson, 2007) becomes more pronounced. Recently, *Many-Core* architectures with tens and hundreds of small cores have been designed for accelerating diverse applications. Those micro-organizations deliver high performance in an affordable power package because they provide quasi linear complexity and power performances (Borkar, 2007; Wang, Zhao, Li and Wang; Sodan, Machina, Deshmeh, Macnaughton and Esbaugh, 2010). Some of these solutions are closely related to the reconfigurable computing research (Wentzlaff *et al.*, 2007). emerging paradigm explained in Section 3.

Cognitive information processing inherently includes vision, audio and haptics processing. This article shows a new model for supporting cognitive information processing. A cognitive processing environment composed of several *customized*, *heterogeneous* and *adaptable* processing cores that overcome the general purpose processing capabilities of the aforementioned processor solutions are envisioned. New hardware solutions must cope with customized architectures for setting up the processing of non connectionist approaches as well as to create networks of small processing elements such as the ones

6 An embedded system is a special-purpose computer system designed to perform dedicated functions. A difference from the general-purpose computer approach, such as a personal computer, an embedded system performs one or a few pre-defined tasks, usually using customized hardware not usually found in a general-purpose computer.

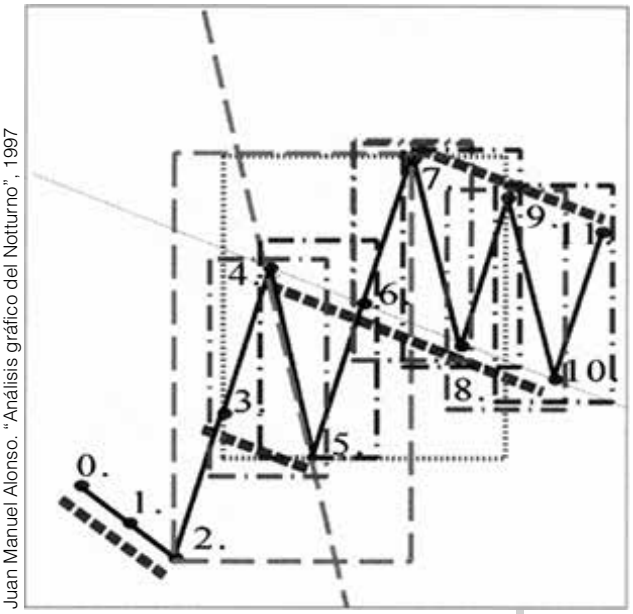
suggested by the connectionism approach (ANN) (Liqing and Bao-Liang, 2010). Furthermore, considering some rule evaluated systems such as FL or even CBR (Pal and Shiu, 2004), which are paradigms that require new technology support, for example fetching several rules or cases concurrently. Therefore new micro-organizations for supporting these cognitive information processing applications must be investigated.

3. Cognitive Information Processing

Cognitive Information processing will require customized hardware solutions not totally based on bio-inspired systems. Traditionally, at the macro system level, hardware acceleration units were based on the concatenation of discrete adders, subtractors, multipliers, dividers and comparators operations (Lee, Murat Fiskiran, Shi and Yang, 2002; Khailany, Dally, Rixner, Kapasi, Mattson, Namkoong, Owens, Towles and Chang, 2001). This concatenation has the scope to set up complex operations, an approach used by several state of the art technology solutions, including reconfigurable related technologies (Miyamori and Olukotun, 1998; Mirsky and DeHon,1996). At the fine grain level, however, you can consider parallelism at the bit level. Important research has sought to diminish the delay and latency in fundamental computer operations such as addition and multiplication. Swartzlander introduces the *Merged Arithmetic (*Swartzlander, 1980) approach which dissolved the boundaries between multiplication and addition during multiply-accumulate operations. These techniques continue to be widely used in Digital Signal Processors (DSP) (Lee, Chung, Yoon and Myung-ok Lee, 2000). Furthermore, *Compound Arithmetic* improves upon merged arithmetic by enabling the operation of additional arithmetic operations (Vassiliadis, Blaner and Eickemeyer, 1994). Compound arithmetic joins a sequence of simple numeric expressions using arithmetic operators like addition, subtraction, multiplication and division. Example of compound arithmetic includes the "*Fast Computation of Compound Expressions in Two's Complement Notation*" (Hakkennes, Vassiliadis and Cotofana, 1997) and "*High-performance 3-1 Interlock Collapsing ALU's"* (Phillips and Vassiliadis, 1994) and "Arithmetic Soft-Core Accelerators" (Calderón, 2007).

Compound Arithmetic can therefore help to accelerate the processing of soft computing techniques including machine learning, robot control and mechatronic manipulators. These applications use existing data representation (*e.g.*, 8 bit, 16 bit, single word floating point, etc.) to perform matrix-vector and

matrix-matrix operations (Calderón, 2007). Based on these facts, a group of customized Cognitive Processors (CP) could be created with customized compound and collapsed features (Calderón, 2007). These cognitive information processing devices could speed up with relatively small hardware requirements in comparison to simple space parallelism approaches that processors actually offer.



Juan Manuel Alonso. "Análisis gráfico del Notturno", 1997

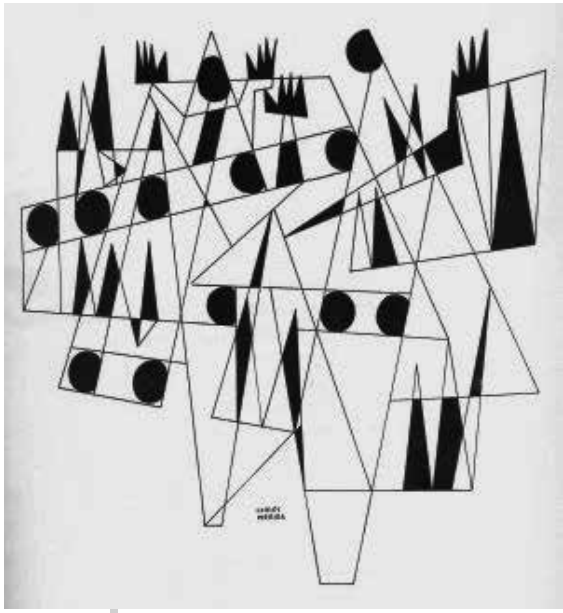
Cognitive Processor technology development requires tools that facilitate the design exploration and platforms that show flexibility and adaptability, features that are intimately linked to Reconfigurable Computing (RC) technologies.

3.1. Reconfigurable Computing (RC): The research tool

Flexibility is the key characteristic for designing novel cognitive information processing devices. Thus, flexible and adaptable hardware systems capable of achieving high degrees of plasticity in both knowledge representation and processing-elements complexity should be investigated. Computational flexibility and adaptability can be achieved with either *variable-structures* or *adaptable-systems* (Hamblen, 2000), paradigms that were introduced by Estrin (1960). Estrin proposed to create a:

> ... fixed plus variable structure computer organization as an approach to removal of structural rigidity from special purpose computation. In essence a variable structure computer system consists of a high-speed general purpose computer (the fixed part, F) operating in conjunction with a second system (the variable part, V) comprised of large of small high-speed digital substructures. The second system can be reorganized into a variety of problem oriented special purpose configurations and this property-use of the same hardware in a variety of computing configurations-makes special purpose computation of a large class of diverse problems on the variable structure computer system feasible.

Since then academic and industrial researchers have been developing reconfigurable computing. Furthermore, during the last two decades we have seen the revisit to this approach (Hartenstein, 2001; Compton and Hauck,



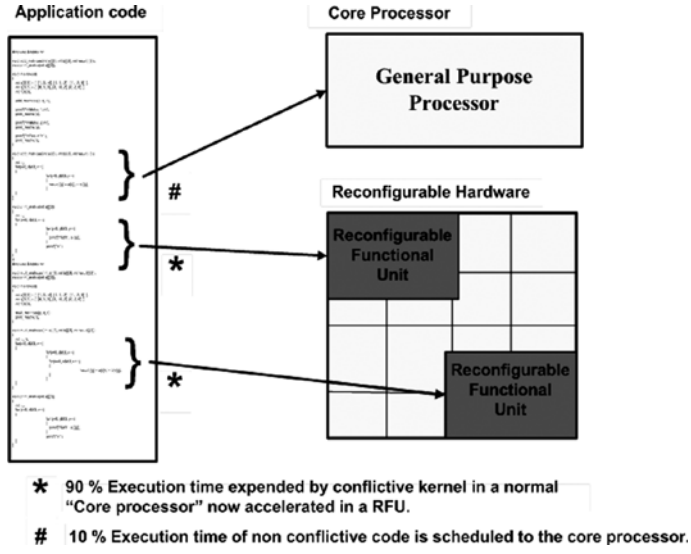
Carlos Mérida

2002; Todman, Constantinides, Wilton, Mencer and Luk, 2005). Current technology is capable of fitting a complete reconfigurable System on Chip (SoC) into a single die (Salminen, Kulmala and Hämäläinen, 2005). Reconfigurable technology uses predominantly Field Programmable Gate Arrays (FPGAs). Additional examples of RC can be found in Mirsky and DeHon (1996); Vassiliadis, Wong, Gaydadjiev, Bertels, Kuzmanov and Moscu Panainte (2004) Kozyrakis, Gebis, Martin, Williams, Mavroidis, Pope, Jones, Patterson and Yelick (2000); Mei, Vernalde, Verkest, De Man and Lauwereins (2003); Kinane, Casey, Muresan and O'Connor (2005); Tumeo, Monchiero, Palermo, Ferrandi and Sciuto. The hardware accelerators are called Custom-Computing-Units (CCUs) or also refereed as Reconfigurable-Functional-Units (RFUs) independently in this work.

Actually, as proposed by Estrin more than five decades ago, a GPP (core processor) augmented with some kind of RFU can be used for optimizing the processing performance (see Figure 2). The rationality behind the Reconfigurable Computing Acceleration Approach uses a RFU as hardware accelerator[7] for speeding up selected kernels while, the non-critical parts, such as code that requires low processor use and small memory bandwidths, are processed by a General Purpose Processor (GPP) as depicted in Figure 2. The GPP is usually responsible for the control of the program's sequencing and issues data or instructions to the specialized RFU.

7 Hardware acceleration is used to speed-up kernels that are executed in software running on the normal general purpose processor. The sequentiality inherent in GPP is accelerated with high concurrent hardware units that process several data usually using the ILP; *e.g.*, the processing of the sum of absolute differences, performs the acceleration in a separate unit from the CPU (functional unit) of several pels in a single instruction.

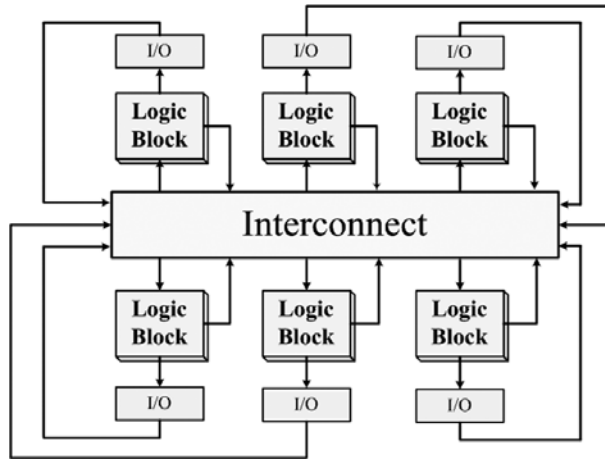
Figure 2. Reconfigurable Computing Acceleration Approach



3.1.1. Reconfigurable Hardware Technologies

Customizable logic devices emerged in the 1970s with the appearance of logic circuitry without fixed structures were called Programmable-Logic-Devices (PLDs). The main idea behind these chips was that logic functions can be realized in sum-of-products form. Therefore, the chips were constructed with an array of *AND* gates used to build the *products* and a set of *OR* gates to produce the *sum* of a combinational logic function. Integrated circuits (IC) created with the aforementioned architecture were called Programmable Array Logic (PAL) (Brown and Vranesic, 2005). From that time the programmable devices have evolved, and the Programmable Logic Arrays (PLA) devices include flip-flops and feedback paths and demonstrate the feasibility for constructing small sequential systems. However, PLAs and PALs were limited by the small sizes of their input-output resources (Dorf,1993). Large combinational and sequential systems are constructed by using Complex Programmable Logic Device (CPLD). CPLDs internally embed a set of Logic Blocks (LB), where each Logic-Block is similar in structure to the PLA or PAL circuits (Wakerly, 2001). Figure 3 depicts schematically a CPLD which is composed of a set of Input/Output (I/O) ports attached to the Logic Block. A centralized interconnect network is used for setting up the CPLD desired functionalities.

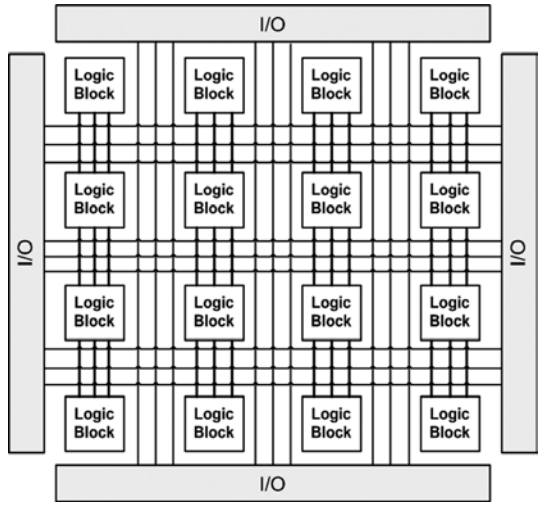
Figure 3. CPLD architecture



Field-Programmable Gate Arrays (FPGAs) (Wakerly, 2001) are an improved technology from the point of view of flexibility and are capable of supporting larger and more complex circuits than CPLD. FPGAs are quite different from CPLD, which do not contain AND neither OR arrays, by being constructed with three components: logic blocks based on memory elements, I/O blocks, and interconnection wires as is presented in Figure 4. The Logic Blocks are built with a small number of inputs and only one output; internally the required functions are implemented with the use of Look-Up Tables (LUT)[8]. FPGAs also contain other memory elements like Flip Flops (FFs), RAMs as well as other fixed circuitry including special arithmetic logic support (Xilinx, 2002). Figure 4 shows that Logic Blocks are embedded in a sea of routing channels in contrast to the simple interconnect network of CPLDs. Those routing channels composed by wires and programmable switches that set up the desired functions. A high level language is used to describe the final functionality of FPGA devices; typically either VHDL (Very-high-speed-integrated-circuit Hardware Description Language) or Verilog HDL.

8 Look-Up Tables (LUT) are memory blocks (*e.g.*, SRAM) used to build logic functions. This efficient way of encoding Boolean logic functions uses this memory arrays that receive 4-6 bits as input and produce usually one or two outputs that are congruent with the logic table.

Figure 4. FPGA Architecture



3.1.2. Reconfigurable Computing terminologies

In the subsection that follows, essential definitions for understanding the architectural characteristics of the reconfigurable computing are presented.

CCU Granularity

This characteristic refers to the degree of adaptability that a reconfigurable system possesses:

- *Fine grained*: refers to systems that are capable to be adapted to new functions at the bit level like in FPGAs. These systems are configured in a bit by bit fashion, modifying small parts of the programmable block, therefore require larger configuration bits compared to other granularity systems.
- *Medium grained:* are connected to systems that work on the nibble size when adapting the functionalities; *e.g.,* small 4-bit ALUs that rearrange together to have different functionalities.
- *Coarse-grained:* bear on systems that are adapted at the word level, *e.g.,* rearrange the use of medium or large ALUs. These systems require less configuration bits compared with the two previous mentioned approaches.

CCU Coupling

This characteristic refers to the way that a CCU or RFU (reconfigurable arrays) are coupled with the core processor:

- *Closely coupled:* in these systems the adaptable functional units are placed in the same data-path of the core processor and therefore are directly controlled by this host processor.
- *Quasi-Tightly coupled:* on such systems the CCU acts like a coprocessor and shares the same die of the core processor. A small overhead in communication is introduced with respect to a closely coupled unit.
- *Loosely coupled:* on these systems the CCU acts like a coprocessor situated on a different die. The communication is achieved through a system bus; and because of that, those systems feature a large latency level when compared with the previous coupled organizations enounced.

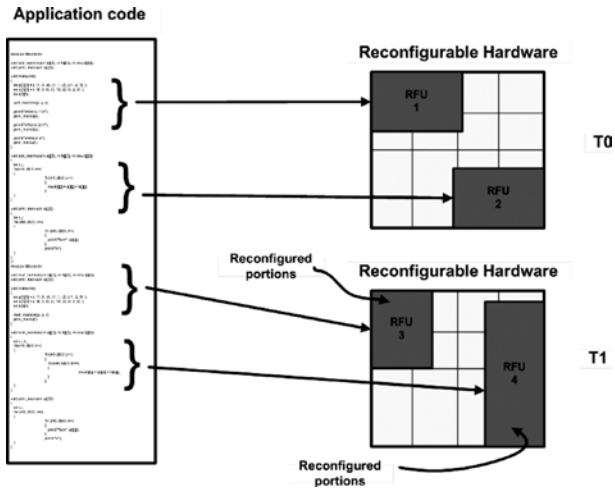
CCU Adaptability

This characteristic refers to the ability that a system has for adapting to new functions statically and dynamically:

- *Static adaptability:* in these systems the operation of the programmable device needs to stop its operations to adapt for new functionalities. After their adaptation (Xilinx, 2004; 2007; Siozios, Koutroumpezis, Tatas, Soudris and Thanailakis, 2005) the systems can start to operate again.

  Generally the CCU is configured once at the set-up time of the systems that it resides.
- *Dynamic adaptability:* in these systems the CCU can partially be adapted to new functions without the interruption of the operations of the entire field programmable logic device. Run time reconfiguration can be done modifying the entire CCU (*e.g.,* swap to/from local memory (Trimberger, Carberry, Johnson and Wong, 1997) the appropriate configuration) and adapting the hardware in a demand basis as is depicted in Figure 5. At time T0 two reconfigurable functional units (RFU 1 and RFU 2) are mapped into the adaptable hardware. Demanded by the application code in time T1, another two RFUs (RFU 3 and RFU 4) are adapted into the reconfigurable resource, occupying the space of the previous ones.

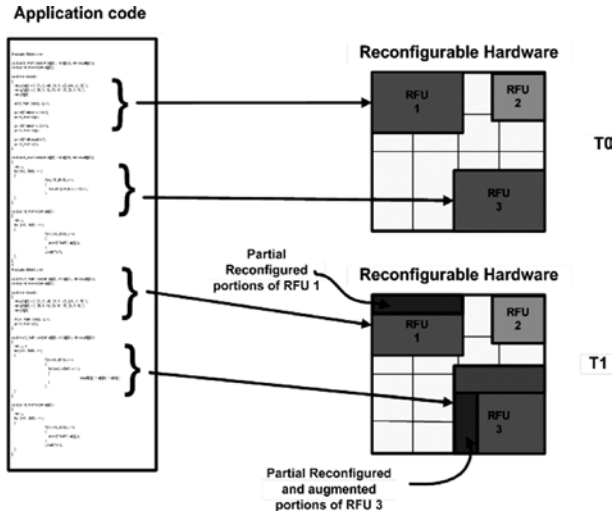
Figure 5. Dynamic reconfiguration



Furthermore, adaptable systems evolve and current FPGAs allow performing partial reconfiguration[9] of an even smaller section of hardware on run time. The potential benefit of dynamic reconfiguration and specifically run-time reconfiguration (RTR) (Siripokarpirom, 2006) is still under investigation.

Figure 6 depicts the partial reconfiguration steps. At time T0 the two RFUs are mapped into the programmable device; at time T1 those RFUs mapped before, are partially modified to achieve new functional characteristics.

Figure 6. Partial dynamic reconfiguration approaches



9 Only parts of the hardware are modified while the other static parts remain on operation.

Processors with adaptable characteristics (Vassiliadis *et al.*, 2004; Kozyrakis, *et al.*, 2000; Mei *et al.*, 2003) have demonstrated great flexibility and a good performance in order to replace the traditional solutions in high demand tasks (Villasenor and Hutchings, 1998), offering spatial and temporal parallelism characteristics (DeHon, 1993) and also the inherent bit level parallelism (Kastrup, 2001). Table 2 summarizes the principal characteristics of adaptable computing compared to traditional solutions.

Table 2. Advantages and drawbacks of Adaptable Processors

	Power	Performance	Flexibility	Time to Market
GPP	Medium	Low	Medium	Medium
ASIC	Low	High	Low	High
ASIP	Medium	Medium	Medium	Medium
RCP	Medium	High	High	Low

RCP: Reconfigurable Computer Processor (e.g., based on IBM)

Adaptable technology (configurable) comes associated with the concept of *Virtual Hardware*, in which any application believes it has an infinitely sized and customizable engine to run on. Therefore, with a system that supports this hardware on demand paradigm, it is possible to overcome the intrinsic hardware obsolescence of other technologies due to their low flexibility (see Table 2). Nevertheless, the main drawback of such adaptable systems is that current technology requires significant time for reconfiguration[10]. Several studies show the importance of Run Time Reconfiguration (RTR) (Cadenas, Megson and Plaks, 2000; Heron, Woods, Sezer and Turner, 2001; Wirthlin and Hutchings, 1998), and some researchers look for hiding the configuration latency time. Another approach proposes working with different contexts[11] (Scalera and Vázquez, 1998); the dynamic of this solution is based on switching the context on demand. A similar approach (Trimberger *et al.*, 1997) stores different configurations in the internal memory of the FPGAs, and has the capability to change context in a single cycle. The previously presented solutions constitute the first approaches in order to hide the configuration time and fulfill the goal of having a fully customizable platform that works on demand basis.

10 FPGA technology is evolving and the configuration time of the devices is diminishing gradually. For example, the Virtex II PRO XC2VP50 device, working at 50MHz requires the program 2,628 frames, 47.55 ms for storing 19,005,696 bits (Kuzmanov *et. al*, 2003). The reconfiguration time is reduced in one order of magnitude as can be seen in (Noguera, 2007), nevertheless, it remains huge for some applications and reconfiguration latency is still a major shortcoming of the current FPGAs.

11 In this work context refers to the minimal set of bit stream needed to establish the whole set up of any particular digital system block into an FPGA

4. A cognitive information processing platform

The basic ideas underlying Artificial Intelligence (AI) were proposed by John McCarthy (Andresen, 2002) and later formalized by himself (Mc Carthy, 1990) and colleagues of Dartmouth college meeting including Marvin Minsky (Minsky, 1985 and 2006). Both are also noted for developing this branch of Cognitive Sciences. Several projects have resulted from the AI research community that attempted to create intelligent mechanisms capable of generalization, planning, reasoning by example, designing, etc. Those projects have been successful in narrow knowledge domains, but usually require huge amounts of computational processing resources and were not designed to work in real time scenarios.

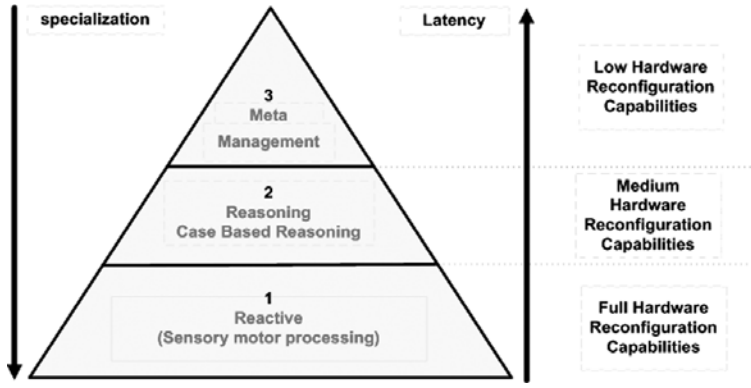
The EPIC architecture for modeling human information-processing and performance presented in Meyer and Kieras (1997a and 1997b) was one of the pioneering attempts to create a customized platform for cognitive processing and used a rule based processing system. Another pioneering and interesting work in intelligent entities was done in the MIT's Media Lab. From this laboratory two main projects emerged: Cognition (COG) and Kismet (Brooks, *et al.*, 1998; Aryananda, 2001; Breazeal, 1999; Breazeal and Scassellati, 1999). Also the fully articulated humanoid i-cub was recently created giving another platform to investigate AI (Vernon *et al.*, 2007). These projects mark some important landmarks in the development of intelligent robots capable of learning, as well as a higher degree of social interaction compared with previous solutions. The functionality of MIT systems was implemented with a set of computers and microcontrollers in the COG's case (Brooks, ***et al.***, 1998) and four Digital Signal Processor (DSP), and two micro-controllers (Breazeal, 1999; Breazeal and Scassellati, 1999) in Kismet's case. These GPPs have low performance when real time is a requirement (see Breazeal comments in Breazeal, 1999; Breazeal and Scassellati, 1999).

4.1. Outlining the Steps for the Development of Smart Machines

The design cost of complex and customized chips is decreasing as new design tools and Hardware Description Languages like VHDL and Reconfigurable Technologies are developed. High performance electronics are improving more rapidly than Moore's Law with the arrival of reconfigurable technologies, especially with the new developments on FPGA devices (Xilinx).

Reasoning with a set of sensor data considers the sensor fusion reasoning approach (transversal reasoning). Case reasoning is based on transversal reasoning across the planar fields of auditory, visual and haptic experience. Each way of reasoning is suited for specific domain problems, (*e.g.,* vision, audio, haptic, etc.) within our proposed multiple layered architecture. Figure 8 depicts the main three layers of the proposed approach.

Figure 7. A proposed Cognitive Hardware Architecture: from point of view of processing



The following subsection presents a brief description of the three main components depicted in Figure 8. It is foreseen that a rather simple structure could enable a Cognitive Hardware Architecture that supports adequately Cognitive Information Processing.

4.1.1. Reactive Layer

Animals respond to stimuli in their environment based on the assumption that these responses are the basic intrinsic mental activity. A reactive layer is created to replicate this behavior within the proposed system. To achieve this, it is foreseen that a group of customized processors will be used to process vision, audio and haptic sensory signals. The representation and organization of the domain knowledge for setting up an "action-reaction" mechanism that is fundamental for the aim of replicating intelligence. The investigation of different knowledge representations based on the soft computing techniques could enable a more complete solution for replicating cognitive information processing. In this proposal, reactions are categorized in two ways: reactions to external stimulus (e.g., sensorial, machine states) and reactions to internal needs (e.g., adaptations). The reactive layer is then subdivided in two sub layers. Each dedicated processor of the first sub-layer is intended to operate with particular

transducer inputs, resulting in fast reactive feedback. The second sub-layer carrying out reasoning with all transducer inputs achieving case based reasoning in a transversal way. This second sub-layer has a larger and more intrinsic processing time than the first sub-layer. The soft computing processors, therefore, must deal with different latencies, the first sub-layer requires being fast with real hard-deadlines, while the second sub-layer has softer deadlines, but still has real time response features.

4.1.2. Reasoning Layer

The achievement of more complex goals is carried out in a case based reasoning layer. This layer monitors and predicts behaviors. The reasoning layer rules, controls (modifies) and carries out a parallel reasoning that is based on the reactive layer's outcomes and converts some of these outcomes into motor actions via the specialized processor of the Reactive Layer as necessary. This layer establishes a "Quasi-Consciousness" approach by remembering a previous similar situation to solve the given processing requirements. The different solution options are evaluated in accordance of the Meta-Management Layer. Tasks such as categorization, evaluation, selection and elimination are the fundamental processing functions of the reasoning layer.

4.1.3. Meta-Management Layer

The Meta-Management Layer is in charge of assigning the high level models and modifying the machine beliefs that influence actions, memory-structure, learning and planning. By monitoring the activities of the two preceding layers, the Meta-management Layer is responsible for the knowledge of the domain's knowledge created from the previous reasoning and reactive layers. For example, this layer is in charge of regulating the different problem solving paradigms (soft-computing processors), the different knowledge representational schemes and various indexing techniques used throughout the architecture. The layer also governs the effective use of multiple reasoning mechanisms for building the Transversal Case Based Reasoning paradigm.

5. Conclusions and future work

We propose a new hardware architecture design that may meet the cognitive information processing needs found in intelligent robotics and cognitive sciences fields in the best way. Our proposal focuses on using transversal case-based reasoning on a group of dedicated processors and hardware accelerators.

The proposed three layer architecture defines an innovative cognitive architecture that operates in HMC environments. This approach represents a shift from traditional processing paradigms into what we call cognitive information processing hardware. Such a platform must provide new mechanisms for concurrent retrieval, indexing and learning. We suggest further integrating the advances in soft computing and case based reasoning to meet more effectively the cognitive information processing needs. By using the main features of reconfigurable hardware design, our proposal allows for a long term research direction and capitalizes the following advantages over traditional approaches.

1. Life Cycle: The life cycle is extended because obsolescence of the solution project system is delayed throughout use of reconfigurable technologies.
2. *Adaptability*: New features, needs and emerging constraints are more easily addressed by adapting previously designed units, and reconfiguring them on demand for new needs, even at actual run time. The set of cognitive-processors can be easily adapted due to their "soft-cores" intellectual property (IPs) features. For example, extending the ISAs and creating new customized arithmetic units for new needs.
3. *IP reuse*: Reconfigurable units are in essence IP units; thus they are easily adaptable to new circumstances and projects, which establishes a new technology niche.
4. *Innovation*: Reconfigurable subunits of the Cognitive information processing platform (i.e., neuro-fuzzy controllers, fuzzy logic machines, **etc**.) can become new cores for the development of new micro-organizations for intelligent entities, e.g., new intelligent unmanned autonomous systems.

In summary, the research, design and development of the proposed set of processors able to cope with the needs of cognitive throughout transversal case based reasoning requires a multidisciplinary team. The resulting product would benefit researchers in the fields of computer science, computer engineering, mechatronics, cognitive sciences and even psychology. We see it as an exciting opportunity to set up networks of research in this emerging field.

Recibido: 1 de mayo de 2014
Aceptado: 28 de mayo de 2014

References

1. Andresen, Scott L. *John McCarthy: father of AI.* Intelligent Systems, 2002.
2. Aryananda, Lijin. *Online and Unsupervised Face Recognition for Humanoid Robot: Toward Relationship with People*. International Conference on Humanoid Robots, 2001.
3. ARM, *Cortex™-A9 NEON™ Media Processing Engine Revision: r3p0 Technical Reference Manual.* <http://infocenter.arm.com/help/topic/com.arm.doc.ddi0409g/DDI0409G_cortex_a9_neon_mpe_r3p0_trm.pdf>
4. Benford, Steve, Carsten Magerkurth and Peter Ljungstrand. *Bridging the Physical and Digital in Pervasive Gaming.* Communications of the ACM, 2005.
5. Benini, Luca, Alessandro Bogliolo, Ro Bogliolo and Giovanni de Micheli. *A Survey of design techniques for system-level dynamic power management.* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2000.
6. Biologically. *Inspired Computing Approaches To Cognitive Systems: a partial tour of the literature*. Digital Media Systems Laboratory. <http://www.hpl.hp.com/techreports/2003/HPL-2003-11.pdf>
7. Borkar, Shekhar. *Thousand Core Chips. A Technology Perspective.* 44th Design Automation Conference DAC'07, 2007.
8. Bormans, Jan, Jean Gelissen and Andrew Perkis. *MPEG-21: The 21st Century Multimedia Framework*. IEEE Signal Processing Magazine, 2003.
9. Breazeal, Cynthia. *Early Experiments using Motivations to Regulate Human-Robot Interaction*. Proceedings of the 1999 Symposium on Imitation in Animals and Artifacts (AISB99), 1999.
10. Breazeal, Cynthia and Brian Scassellati. *A Context-Dependent Attention System for a Social Robot.* Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI99), 1999.
11. Brooks, Rodney. A., Cynthia Breazeal, Matthew Marjanovic, Brian Scassellati and Mathew M. Williamson. *The Cog Project: Building a Humanoid Robot.* Computation for Metaphors, Analogy and Agents, Springer Lecture Notes in Artificial Intelligence,Springer-Verlag, 1998.
12. Brown, Stephen and Zvonko Vranesic. *Fundamental of Digital Logic with VHDL Design.* McGraw-Hill. ISBN 0-07-246085-7, 2005.
13. Burger, Doug. *Limited bandwidth to affect processor design*. IEEE MICRO,1997.
14. Cadenas, J. 0., G. M. Megson and T. P. Plaks. *Quantitative evaluation of three reconfiguration strategies on FPGAs: a case study*. The Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region, 2000.
15. Calderón, Daniel R. Humberto. *Arithmetic Soft-Core Accelerators.* ISBN 978-90-807957-7-8, 2007.
16. Carlson, David A., Ruben W. Castelino and Robert O. Muellera. *Multimedia Extensions for a 550-MHz RISC Microprocessor*. IEEE *Journal of Solid-State Circuits*, 32 (11), 1618-1624, 1997.

17. Chien, Andrew A. and Vijay Karamcheti. *Moore's Law: The First Ending and A New Beginning*. IEEE Computer Magazzine, 2013.
18. Clark, Nathan, Wilkin Tang and Scott Mahlke. *Automatically Generating Custom Instruction Set Extensions*. Proceedings of the Workshop on Application-Specific Processors (WASP), 2002.
19. Compton, Katherine and Scott Hauck. *Reconfigurable Computing: a Survey of Systems and Software*. ACM Computing Surveys (CSUR), 2002.
20. DeHon, Andre. *Reconfigurable Architectures for General Purpose Computing.* PhD. Thesis, Massachusetts Institute of Technology, 1993.
21. Diefendorff, Keith, Pradeep K. Dubey, Ron Hochsprung and Hunter Scales. *AltiVec Extension to PowerPC Accelerates Media Processing.* IEEE MICRO, 2000.
22. Dobai, R. and L. Sekanina. *Towards evolvable systems based on the Xilinx Zynq platform*. Proceedings of the IEEE International Conference on Evolvable Systems (ICES), 2013.
23. Dorf, Richard C.. *Electrical Engineering Handbook.* CRC Press - ISBN 0-8493-0185-8, 1993.
24. Eden M. and M. Kagan. *The Pentium(R) processor with MMX TM technology*. Proceedings of the IEEE Compcon, 1997.
25. Elliott, Jim. *IBM Mainframes. 45 Years of Evolution. IBM Corporation*, 2010. <http://www.vm.ibm.com/devpages/jelliott/pdfs/zhistory.pdf>
26. Estrin, Gerald. *Organizations of computer systems - the fixed plus variable structure computer*. Proceedings of the Western Joint Computer Conference, 1960.
27. Hager, Gregory D., Allison M. Okamura and Russell H. Taylor. *Human-Machine Collaborative Systems: Intelligent Virtual Fixtures and Space Applications*. AAAI Symposium, 2006.
28. Hakkennes, Edwin, Stamatis Vassiliadis and Sorin Cotofana. *Fast Computation of Compound Expressions in Two's Complement Notation*. Proceedings of the 15th IMACS World Congress on Scientific Computation, Modeling and Applied Mathematics, 1997.
29. Hamblen, James O. *Rapid prototyping using field-programmable logic devices.* IEEE MICRO, 2000.
30. Hartenstein, R.. *A Decade of Reconfigurable Computing.* Proceedings of the Design Automation and Test in Europe (DATE'01), 2001.
31. Hayes, John P. *Computer Architecture and Organizations*. McGraw Hill. ISBN 0-07-027366-9, 1988.
32. Hennessy, John L. and David A. Patterson. *Computer Architecture: A Quantitative Approach.* Morgan Kaufmann Publisher. ISBN: 1-55860-596-7, 2007.
33. Heron, J. P., R. Woods, S. Sezer and R. H. Turner. "Development of a Run-Time Reconfiguration System with Low Reconfiguration Overhead". *Journal of VLSI Signal Processing*, 28(1-2), 97-113, 2001.
34. Hwang, Kai. *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. ISBN-10: 0070316228, 1992.
35. IBM. *IBM creates new foundation to program SyNAPSE chips inspired by human brain.* <phys.org/pdf295153713.pdf>

36. Intel. *MCS-48 Family of Single Chip Microcomputers: Users Manual*, 1978. <http://ftp.df.lth.se/pub/bitsavers.org/pdf/intel/8048/9800270D_MCS-48_Family_Users_Manual_Jul78.pdf>

37. ---------- *Mcs@51 Microcontroller Family User's Manual*, Order Nº: 272383-002, 1994. <http://www.ise.pw.edu.pl/impuls/emisy/8051_um2.pdf>

38. Kastrup, Bernardo. "Automatic Synthesis of Reconfigurable Instruction Set Accelerators". PhD. Thesis, TU/e, ISBN 90-74445-50-0, 2001.

39. Khailany, Brucek, William J. Dally, Scott Rixner, Ujval J. Kapasi, Peter Mattson, Jinyung Namkoong, John D. Owens, Brian Towles, and Andrew Chang. *Imagine:Media Processing with Streams.* IEEE MICRO, 2001.

40. Kim, Jong-Myon. *The Impact of Multimedia Extensions for Multimedia Applications on Mobile Computing Systems*. Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 2008.

41. Kinane, Andrew, Alan Casey, Valentin Muresan and Noel O'Connor. "FPGA-based conformance testing and system prototyping of an MPEG-4 SA-DCT hardware accelerator". Proceedings of the 2005 IEEE International Conference on Field-Programmable Technology, 2005.

42. Kloet, Bas & Paual van Tilburg, *John Vincent Atanasoff: Inventor of the Digital Computer.* Technische Universiteit Eindhoven, 2006 <http://paul.luon.net/essays/HoC-Atanasoff.pdf>

43. Kozyrakis, C., J. Gebis, D. Martin, S. Williams, I. Mavroidis, S. Pope, D. Jones, D. Patterson and K. Yelick. "Vector IRAM: A Media-oriented Vector Processor with Embedded DRAM", Proccedings of 12th Hot Chips Conference, 2000.

44. Kuzmanov, Georgi, Georgi Gaydadjiev, and Stamatis. Vassiliadis. *"Loading rm-code: Design Considerations"*. Proceedings of the Third International Workshop on Computer Systems: Architectures, Modeling, and Simulation. SAMOS03, 2003.

45. Lee, Corinna Grace and Mark Graham Stoodley. "Simple Vector Microprocessors for Multimedia Applications". Proceedings of the 31st Annual ACM/IEEE International Symposium on Micro architecture, 1998.

46. Lee, Ruby B., A. Murat Fiskiran, Zhijie Shi and Xiao Yang. "Refining Instruction Set Architecture for High-Performance Multimedia Processing in Constrained Environments". Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP02). 2002.

47. Lee, Seung-min, Jin -hong Chung, Hyung-seok Yoon and Mike Myung-ok Lee. "High Speed and Ultra-Low Power 16X16 MAC Design using TG techniques for Web-based Multimedia System". Proceedings of the IEEE International Conference on Asia and South Pacific Design Automation(ASP-DAC'00), 2000.

48. Lempel, O., A. Peleg and U. Weiser. "Intel's MMX Technology-a new Instruction Set Extension". Proceedings of the IEEE Compcon '97, 1997.

49. Liqing, Zhang and Lu Bao-Liang. *Advances in Neural Networks*. 7th International Symposium on Neural Networks, 2010.

50. Mange, Daniel and Marco Tomassini. *Bio-InspiredComputing Systems Toward Novel Computational Architectures*. Evolution and learning in autonomous robotic agents, Lausanne, 1998.

51. McCarthy, John, Marvin L. Minsky, Nathaniel Rochester and Claude E. Shannon. *A Proposal for the Dartmouth Summer Research project on artificial Intelligence*, 1955. <http://www-formal.stanford.edu/jmc/history/dartmouth.pdf>

52. McCarthy, John. Formalizing *Common Sense: Papers by John McCarthy*. Ablex. ISBN-13: 978-0893915353, 1990.

53. Mei, Bingfeng , Serge Vernalde, Diederik Verkest, Hugo de Man and Rudy Lauwereins. "ADRES: An Architecture with Tightly Coupled VLIW Processor and Coarse-Grained Reconfigurable Matrix". Proccedings of the FPL, 2003.

54. Meyer, D. E., and D. E. Kieras. *"A Computational Theory of Executive Cognitive Processes and Multiple-Task Performance: Part 1. Basic Mechanisms". Psychological* Review, 104(1):3-65, 1997a.

55. ---------- *"A Computational Theory of Executive Cognitive Processes and Multiple-Task Performance: Part 2. Accounts of Psychological Refractory-Period Phenomena". Psychological Review*, 104(4):749-79, 1997b.

56. Minsky, Marvin. *Society of Mind*. Simon & Schuster, 1985.

57. ---------- *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon & Schuster, 2006.

58. Mirsky, Ethan and André DeHon. *MATRIX: A Reconfigurable Computing Architecture with Configurable InstructionDistribution and Deployable Resources*. IEEE International Symposium on FPGAs for Custom Computing Machines, 1996.

59. Misra, Janardan and Indranil Saha. *Artificial neural networks in hardware: A survey of two decades of progress.* Neurocomputing, 74, 239–255, 2010.

60. Miyamori, Takashi and Kunle Olukotun. "REMARC: Reconfigurable Multimedia Array Coprocessor". Proceedings of ACM/SIGDA International Symposium on Field Programmable Gate Arrays, 1998.

61. Motorola. Single–Chip Microcontrollers (CSIC), Motorola Master Selection Guide. <http://noel.feld.cvut.cz/hw/motorola/books/sg73/pdf/2_5csic_msg.pdf>

62. ---------- MC68HC16Y1 16-Bit Modular Microcontroller, Freescale Semiconductor, Inc., 1992. <http://cache.freescale.com/files/microcontrollers/doc/prod_brief/MC68HC16Y1TS.pdf?fr=gdc>

63. Murray, John. "Wearable Computers in Battle: Recent Advances in the Land Warrior System". Proceedings of the 4th IEEE International Symposium on Wearable Computers, 2000.

64. Noguera, Juanjo. *Application driven Research in Partial Reconfiguration.* TU Delft – CE Colloquium. <http://ce.et.tudelft.nl/cecoll/slides/07/0524noguera.pdf, 2007>

65. Painkras, Eustace, Luis A. Plana, Jim D. Garside, Steve Temple, Simon Davidson, Jeffrey Pepper, David M. Clark, Cameron Patterson and Steve Furber. "SpiNNaker: A Multi-Core System-on-Chip for Massively-Parallel Neural Net Simulation". Proceedings of the IEEE Custom Integrated Circuits Conference (CICC), 2012.

66. Pal, Sankar K., and Simon C. K. Shiu. *Foundations of Soft Case-Based Reasoning*. ISBN 0-471-08635-5, 2004.

67. Palem, Krishna and Avinash Ligamneni. "What to Do About the End of Moore's Law, Probably!". Proceedings of the 49th Annual Design Automation Conference , 2012cxcFd.

68. Parkhurst, Jeff, John Darringer and Bill Grundmann. "From Single Core to Multi-Core: Preparing for a new exponential". Proceedings of the 2006 IEEE/ACM international Conference on Computer-aided design ICCAD '06, 2006.

69. Petriu, E. M., T. E. Whalen, R. Abielmona and A. Stewart. *Robotic Sensor Agents: a New Generation of Intelligent Agents for Complex Environment Monitoring*. IEEE Instrumentation and Measurement Magazine, 2004.

70. Phillips, James and Stamatis Vassiliadis. "High-Performance 3-1 Interlock Collapsing ALU's". Proceedings of the 15th IMACS World Congress on Scientific Computation, Modeling and Applied Mathematics, 1994.

71. Piefer, Rolf and Josh C. Bongard. *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press. ISBN:0-262-16239-3, 2006.

72. Russell, M. and S. Fischaber. "OpenCV based road sign recognition on Zynq". Proceedings of the IEEE International Conference on Industrial Informatics (INDIN), 2013.

73. Salminen, Erno, Ari Kulmala and Timo D. Hämäläinen. *HIBI-based multiprocessor SoC on FPGA*. IEEE International Symposium on Circuits and Systems (ISCAS05), 2005.

74. Scalera, Stephen M. and José R. Vázquez. "The Design and Implementation of a Context Switching FPGA". Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, 1998.

75. Schank, Roger C. *Dynamic Memory Revisited*. Cambridge University Press. ISBN:0-521-63302-8, 1999.

76. Shreekant (Ticky) Thakkar and Thomas Huff. *Internet Streaming SIMD Extensions*. IEEE Computer, 1999.

77. Siozios, K., G. Koutroumpezis, K. Tatas, D. Soudris and A. Thanailakis. "DAGGER: A Novel Generic Methodology for FPGA Bitstream Generation and its Software Tool Implementation". Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS05), 2005.

78. Siripokarpirom, Rawat. "Platform Development for Run-Time Reconfigurable Co-Emulation". Proceedings of The Seventeenth IEEE International Workshop on Rapid System Prototyping, 2006.

79. Sodan, Angela C., Jacob Machina, Arash Deshmeh, Kevin *Macnaughton and Bryan Esbaugh. Parallelism via Multithreaded and Multicore CPUs*. IEEE Computer Society Press, 2010.

80. Swartzlander, Earl Eugene. *Merged Arithmetic*. IEEE Transactions on Computers, 1980.

81. Talla, Deepu, Lizy Kurian John and Doug Burger. *Bottlenecks in Multimedia Processing with SIMD style Extensions and Architectural Enhancements*. IEEE Transactions on Computer, 2003.

82. Todman T. J., G. A. Constantinides , S. J. E. Wilton , O. Mencer and W. Luk. "Reconfigurable Computing: Architectures and Design Methods". IEE Proceedings on Computers and Digital Techniques, 2005.

83. Trimberger, S., D. Carberry, A. Johnson and J. Wong. "A time-multiplexed fpga". Proceedings of The 5th Annual IEEE Symposium on FPGAs for Custom Computing Machines, 1997.

84. Tumeo, Antonino, Matteo Monchiero, Gianluca Palermo, Fabrizio Ferrandi and Donatella Sciuto. "A Pipelined Fast 2D-DCT Accelerator for FPGA-based SoCs". Proceedings of the IEEE Computer Society Annual Symposium on VLSI,ISVLSI'07, 2007.

85. Vassiliadis, S., B. Blaner and R. J. Eickemeyer. "SCISM: A Scalable Compound Instruction Set Machine". *IBM Journal of Research and Development*. 1994.

86. Vassiliadis, Stamatis, Stephan Wong, Georgi Gaydadjiev, Koen Bertels, Georgi Kuzmanov and Elena Moscu Panainte. *The MOLEN Polymorphic Processor.* IEEE Transactions on Computers, 2004.

87. Vernon, David, Giorgio Metta and Giulio Sandini. *The iCub Cognitive Architecture: Interactive Development in a Humanoid Robot*. IEEE International Conference on Development and Learning, 2007.

88. Villasenor, J. and B. Hutchings. *The Flexibility of Configurable Computing*. IEEE Signal Processing Magazine, 1998.

89. Wakerly, John F. *Digital Design: Principles and Practices*. Prentice Hall. ISBN 0-13-089896, 2001.

90. Wang, Honggang, Jide Zhao, Hongguang Li and Jianguo Wang. *Parallel Clustering Algorithms for Image Processing on Multi-core CPUs*. International Conference on Computer Science and Software Engineering, 2008.

91. Wentzlaff, David, Patrick Griffin, Henry Hoffmann, Liewei Bao, Bruce Edwards, Carl Ramey, Matthew Mattina, Chyi-Chang Miao, John F. Brown III and Anant Agarwal. *On-Chip Interconnection Architecture of the Tile Processor*. IEEE MICRO, 2007.

92. Wirthlin, Michael J. and Brad L. Hutchings. *Improving Functional Density using Run-Time Circuit Reconfiguration [FPGAs]*. Transactions on Very Large Scale Integration (VLSI) Systems, 1998.

93. Xiao Xue; Yao Lina and Luo Junwei. "The Reuse Policy in Developing Multi-agent System". *International Journal of Distributed Sensor Networks*. 2009.

94. Xilinx. *UltraFast Design Methodology Guide for the Vivado Design Suite*, UG949 (v2014.1) April 2, 2014. <http://www.xilinx.com/support/documentation/sw_manuals/ug949-vivado-design-methodology.pdf>

95. ---------- *9 Reasons why the Vivado Design Suite Accelerates Design Productiviy.* <http://www.xilinx.com/publications/prod_mktg/vivado/Vivado_9_Reasons_Backgrounder.pdf>

96. ---------- *Zynq-7000 All Programmable SoC Overview*. DS190 (v1.6) December 2, 2013. <http://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf>

97. ---------- *PlanAhead User Guide.* UG632 (v14.3) October 16, 2012. <http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/PlanAhead_UserGuide.pdf>

98. ---------- *LogiCORE IP AXI Interconnect (v1.06.a)*. DS768 December 18, 2012. <http://www.xilinx.com/support/documentation/ip_documentation/axi_interconnect/v1_06_a/ds768_axi_interconnect.pdf>

99. ---------- *Xilinx Redefines Power, Performance, and Design Productivity with Three New 28 nm FPGA Families: Virtex-7, Kintex-7, and Artix-7 Devices.*

<http://www.xilinx.com/support/documentation/white_papers/wp373_V7_K7_A7_Devices.pdf>

100. ---------- *Virtex II PRO Platform FPGA Handbook UG012,* Xilinx, 2002.

101. ---------- *Xilinx In-System Programming Using an Embedded Microcontroller*, 2004. <http://direct.xilinx.com/bvdocs/ appnotes/xapp058.pdf>

102. ---------- *Configuration and Readback of Virtex FPGAs Using JTAG Boundary-Scan*, 2007. <http://www.xilinx.com/ bvdocs/appnotes/xapp139.pdf>

103. Zhu, Weirong, Vugranam C. Sreedhar, Ziang Hu and Guang R. Gao. *Synchronization State Buffer: Supporting Efficient Fine-Grain Synchronization on Many-Core Architectures.* The 34th International Symposium on Computer Architecture, 2007.

104. Zhu, Weirong, Ziang Hu and Guang R. Gao. "On the Role of Deterministic Fine-Grain Data Synchronization for Scientific Applications:A Revisit in the Emerging Many-Core Era". Proceedings of the Workshop on Multi-Threaded Architectures and Applications - MTAAP, 2007.