ACE16k based stand-alone system for real-time pre-processing tasks

Luis Carranza, Francisco Jiménez-Garrido, Gustavo Liñán-Cembrano, Elisenda Roca, Servando Espejo Meana and Angel Rodríguez-Vázquez

Instituto de Microelectrónica de Sevilla - CNM-CSIC
Edificio CICA-CNM Avda./ Reina Mercedes s/n, 41012 Sevilla, SPAIN
carranza@imse.cnm.es

ABSTRACT

This paper describes the design of a programmable stand-alone system for real time vision pre-processing tasks. The system's architecture has been implemented and tested using an ACE16k chip and a Xilinx xc4028xl FPGA. The ACE16k chip consists basically of an array of 128x128 identical mixed-signal processing units, locally interacting, which operate in accordance with single instruction multiple data (SIMD) computing architectures and has been designed for high speed image pre-processing tasks requiring moderate accuracy levels (7 bits). The input images are acquired using the optical input capabilities of the ACE16k chip, and after being processed according to a programmed algorithm, the images are represented at real time on a TFT screen. The system is designed to store and run different algorithms and to allow changes and improvements. Its main board includes a digital core, implemented on a Xilinx 4028 Series FPGA, which comprises a custom programmable Control Unit, a digital monochrome PAL video generator and an image memory selector. Video SRAM chips are included to store and access images processed by the ACE16k. Two daughter boards hold the program SRAM and a video DAC-mixer card is used to generate composite analog video signal.

Keywords: ACE16k, vision system, stand-alone, vision pre-processing, real-time, full custom control unit.

1. INTRODUCTION

Real time vision pre-processing tasks usually involves a high amount of low-level numerical computations and medium-level tasks which commonly include thousands of decisions that must be made at frame-rate [1] - [2]. To face out these problems largely concerned with number crunching tasks, two digital approaches have been used. On one hand, the serial approach, today leaded by DSPs specifically designed for vision, but traditionally solved using general purpose processors, and on the other hand, the low parallelism approach, tackled using specific designed matrices of tens of processors synthesized either in FPGAs or in ASICs. In both approaches there is a clear physical space between the circuitry devoted to image sensing and the circuitry devoted to digital computing, and the role of the analog processing is restricted to image data acquisition, signal conditioning and data encoding.

ACE chips [3]-[5] represent a mixed signal computation approach to vision pre-processing tasks. These chips comprises an array of elementary analog processors that sense images and processes them according to sequences of instructions that are executed by every elementary processor at the same time. The ACE chip family have been designed to suit vision pre-processing tasks that need high frame rate and low resolution. In particular, the ACE16k chip consists, basically, of an array of 128x128 identical analog sensing and processing units, locally interacting, designed for high speed image pre-processing tasks requiring moderate accuracy levels (7 bits).

VLSI Circuits and Systems II, edited by José Fco. López, Francisco V. Fernández, José María López-Villegas, José M. de la Rosa, Proceedings of SPIE Vol. 5837 (SPIE, Bellingham, WA, 2005) 0277-786X/05/$15 · doi: 10.1117/12.608220

On chip peripheral circuitries provide digital I/O and programming capabilities. The chip has two modes of operation; programming and operating mode. During the operating mode several internal SRAM blocks can be individually accessed and data can be stored in order to prepare the chip for a certain image processing algorithm. On the other hand, in the programming mode, the contents of the memory blocks are selected and transmitted to the cell array in parallel.

The stand-alone system presented in this paper permits to program and execute vision pre-processing algorithms using the ACE16k. The system is designed to store and run different algorithms and to allow changes and improvements. The system's main board includes a digital core, implemented on a Xilinx 4028 Series FPGA, which comprises a custom programmable Control Unit, a digital monochrome PAL video generator and a image memory selector. Two video SRAM are included to store and access images processed by the ACE16k. Two daughter boards holds the program SRAM and a video DAC-mixer card used to generate composite analog video signal.

2. SYSTEM DESCRIPTION

The ACE16k Development Board system (ACE16k-DB) is composed of several hardware, and software components. In the hardware side, a main PCB contains basically all the elements of the system, with several smaller PCBs plugged on it. One of these secondary PCBs contains the program memory, which must be previously written using a specific hardware and software and a personal computer. After this memory has been written and connected to the main board, the ACE16k-DB system operates without any external support. In the software side, several elements have been developed to help in the obtention of the program memory contents required to execute specific tasks, and to write and verify the program memory. As will be seen, the hardware system contains a custom Programmable Control Unit, implemented in an FPGA, and therefore specific instruction sets have been defined, and specific assembler and program loader have been written.

Figure 1 shows a diagram of the sub-blocks of the ACE16k-DB system and the associated central software tools.The main components are the ACE16k chip and its associated optical system, the FPGA, the external program memory board, the video SRAM modules, the video signal generator, and the TFT display. Other components include several push buttons, a program indicator seven-segments display and its driver, an EEPROM, a clock generator, and several led indicators.

A four layer mixed-signal motherboard has been designed to hold the whole system. The power, ground planes and power supply's associated circuitry belonging to the ACE16k have been designed to prevent voltage drops and current loops due to the analog nature of the chip's processing core, which is sensitive to voltage and current unbalances. In addition, the main board includes proper footprints and holes to hold a custom designed socket which holds an adjustable optical mount over the chip. This mount permits the inclusion of any commercially available C-mount lens and the establishment of its horizontal and vertical position over the ACE16k.

The roles of the ACE16k chip and the associated optical system are straight forward. The FPGA hosts several subsystems, the most important of them being the programmable Control Unit. From a behavioral perspective, this custom-designed Control Unit is the central element of the system. It performs specific instructions oriented to control the execution of programmable algorithms within the ACE16k-DB system, working on the optically acquired images, and representing the results on the TFT display. The Control Unit instruction sequence is read from the program memory. Other components implemented within the FPGA include a clock divider, which generates clocks of 10, 20, and 40MHz from the main 80MHz clock generator (external to the FPGA), a digital video generator which drives the video mixer (external to the FPGA) with a digital pixel sequence and its corresponding synchronizing signals, a video memory selector and speed-up pipeline registers for fast external data interchange.

The program memory board contains a 256K x 16 bits SRAM, together with a backup battery and the sensing and control circuitry required to switch between external power supply (when present) and the backup batteries. This memory is divided into 8 equal blocks of 32K x 16 bits, each of them corresponding to one program. In the normal use sequence, this memory is first written from a PC, with the help of specific software and a general purpose parallel port, then unplugged from the PC and connected to the main ACE16k-DB board. After power on, the selection of an specific program (among the eight available) is done sequentially using a push-button.

The two identical video SRAMs, of 256K x 16 bits each, are used to store images coming from ACE16k and to drive the digital video generator with images to be represented in the TFT display. The video memory selector, controlled by the control unit, switches the two video memory blocks among these two functions. In this manner, the video storage unit and the digital video generator can operate independently, allowing continuous and simultaneous image storage (of an image coming from ACE16k) and visualization (of a previously stored image).

The video card comprises basically a video frequency range 8bit DAC and a CMOS switch. It is designed to mix the image and synchronizing signals delivered by the digital core's video sub-block and generate analog monochrome PAL composite video. The video board output drives any PAL-video representation system.

The EEPROM is used to configure the FPGA. Alternatively, during a debugging / upgrading phase, it is possible to configure the FPGA from a PC using a JTAG cable. For this purpose bitstream downloading pins has been added onto the main board. These pins become available changing the FPGA configuration scheme with the aid of on board configuration microswitches.The seven-segments program indicator allows the visualization of the selected program number, while several leds, basically for debugging and diagnosis purposes, are used to monitor specific protocol signals coming from ACE16k. Finally, three push buttons are employed to select the program to be executed, to force the reconfiguration of the FPGA from the EEPROM, and to reset the control unit. Figure 2 shows a photograph of the main system board.

3. ACE16K OVERVIEW

The ACE16K chip is composed of five main functional parts.

1. The Central Core, consisting of an array of 128 x 128 identical elementary units, plus a ring of surrounding cells - used to establish spatial boundary conditions for analog processing -.
2. The Programming Circuitry, consisting of a set of digital memories where the information about the algorithm to be implemented is stored.
3. The I/O Block, containing analog-to-digital and digital-to-analog converters allowing the chip to be hosted by a completely digital environment.
4. Several Test purpose blocks.
5. An autonomous and synchronous cell addressing scheme.

The functions embedded in the core array include 2D programmable optical sensing, 2D programmable analog processing, 2D programmable digital processing, and 2D memory. All the functional blocks are implemented in a spatially-distributed way; i.e. each "pixel-cell" includes several photoreceptors, an analog processing circuitry - essentially performing local convolutions -, logic processing circuitry, and memory circuitry. The analog convolution parameters, the logic processing circuitry, and the photoreceptors can be programmed in a spatially-invariant form - same parameter values for all pixel-cells. This programmability, combined with the internal pixel-wide storage capability allows the realization of complex image pre-processing algorithms. All internal data transferences are image-wide parallel.

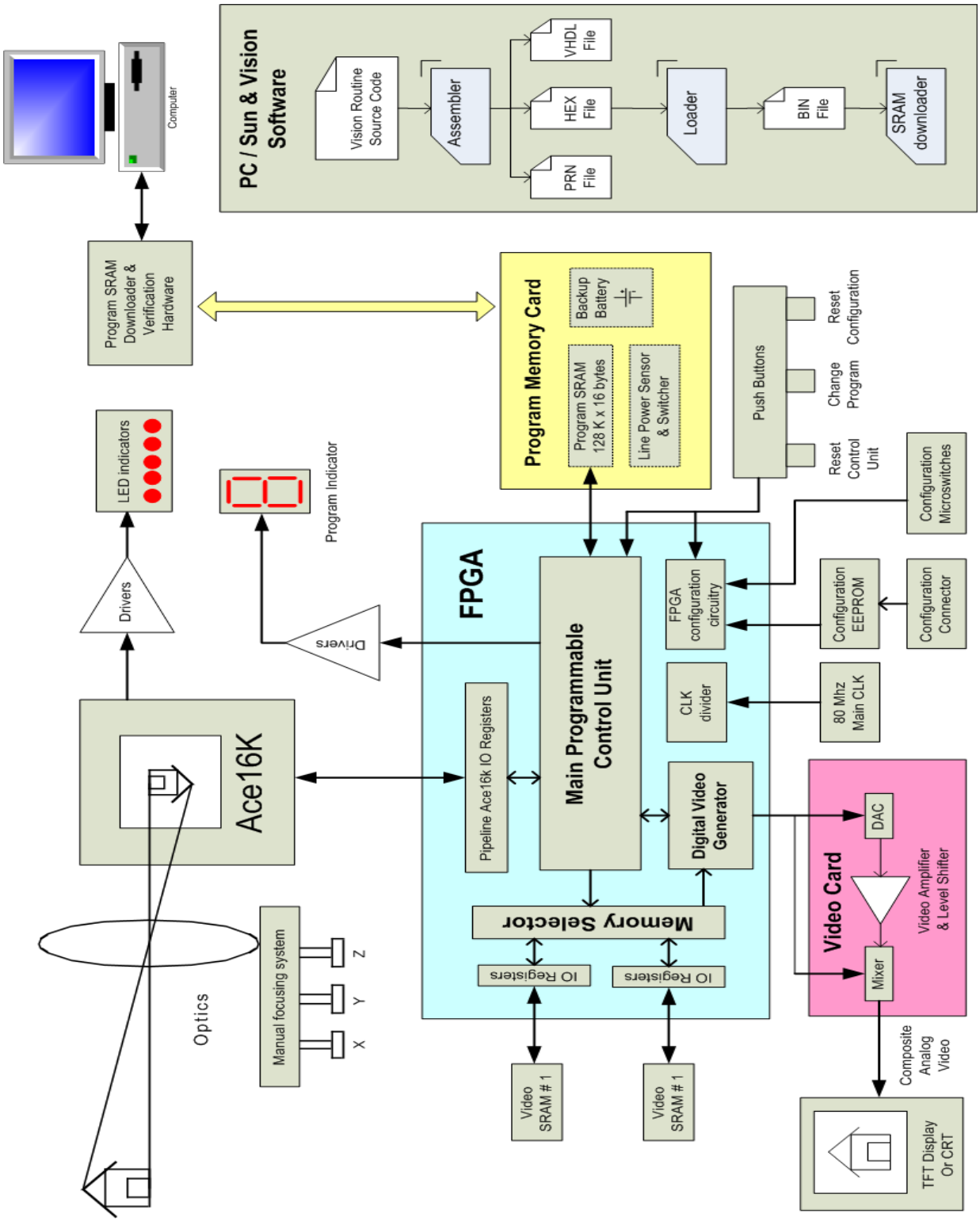

Fig. 1. ACE16k-DB system block diagram

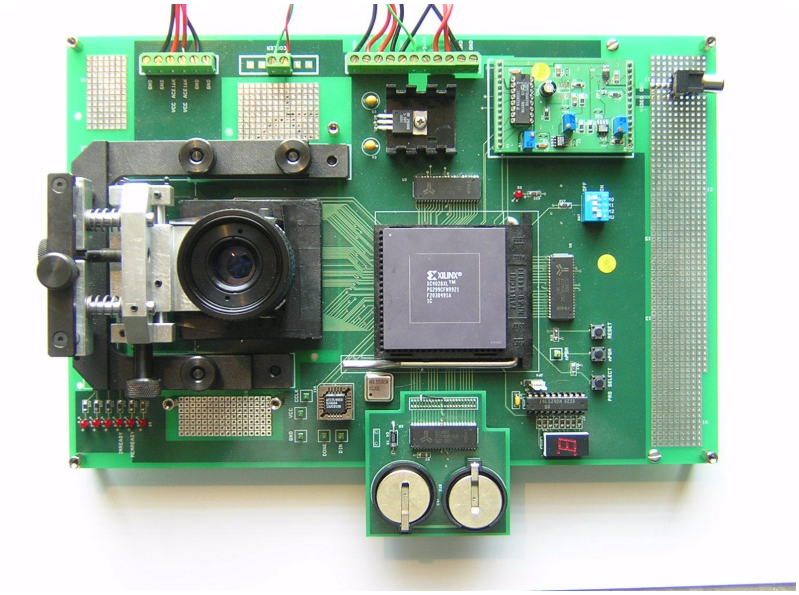

Fig. 2. ACE16k-DB system

In a typical image pre-processing task the ACE16k memory blocks, which comprises 640 words x 16 bits, are first completely filled, storing a set of digital and analog instructions that are suitable for a specific image pre-processing algorithm. The storing order of the instructions is not important; afterwards they are going to be selected (i.e. executed), with the ACE16k chip in operating mode, in an order according to an algorithm. In a conventional image pre-processing task, the chip is programmed once and then it keeps in operating mode while running the visual pre-processing program. The control unit is designed to decode ACE16k execution related instructions faster than any other instruction, in order to increase the visual pre-processing program execution performance.

4. DIGITAL CORE

The systems digital core, implemented using a FPGA, consists basically of a programmable Control Unit, a digital video PAL generator, a memory selector, tri-state bus resources and Input / Output registers allowing fast external access. A clock divider generates signal clocks of 10, 20, and 40Mhz from the main 80Mhz clock generator. Special FPGA hardware resources, mainly low-skew buffers, distributes clock signals and other high fanout control signals throughout the devices adequately.

4.1 PAL Video Generator

This digital core sub-block deals with the generation of the video signal. It is responsible of several tasks that are carried in parallel. The digital video generator circuitry triggers and handles the realization of the following tasks:

1. Video SRAM contents accessing - with the aim of generating the active image scan field signal -.
2. Video SRAM 16 bit output image data to 8 bit DAC input data splitting.
3. DAC input clock signal generation.
4. Memory Selector Arbitrator handling. The Memory Selector Arbitrator (MSA) is responsible of switching the video SRAMs read / write roles. The video generator circuitry signals the MSA of incoming video frames.
5. PAL-video sync signal generation.

4.2 Control Unit

The control unit has been designed with a full custom instruction set oriented basically to control the realization of generic functions within the ACE16k chip. It has been designed at VHDL level, simulated and synthesized. The controller interacts with the ACE16k chip, the program and video SRAMs, the video RAM selector, the digital video generator, the program display and the push buttons.

Instructions and data reside in the same memory, there is no strict separation of instruction and data memory.

The Control Unit uses the *big endian format,* meaning that a word or double word always starts with the high-order byte and ends with the low order byte.

Only one hardware interrupt request is accepted. The Control Unit reacts to an active signal given by the Change Program push button. This leads to a sequence that updates the Program Counter with the next memory block starting address.

All instructions are read in two clock cycles, decoded in one or two cycles, and executed in a variable number of cycles that depend on the specific task performed by the instruction. The instructions are read into a fetch queue, then the attached decoding unit decodes them. Most of the instructions are not executed directly, they are decomposed into a number of elementary steps that, in some cases, will take hundreds of clock cycles. Two execution units allow the realization of parallel processes while writing images in the video SRAM and handling the Memory Selector.

4.3 Control Unit Instruction Set

The Control Unit instruction set fall into three basic categories.

1. ACE16k handling. ACE16k handling instructions are used to program the chip, execute pre-loaded instructions and download pre-processed images. The control unit is designed to decode ACE16k execution related instructions faster than any other instruction, in order to increase the visual pre-processing program execution performance.

2. Control Transfer. Control-transfer instructions include calls, unconditional jumps and up to two nested loops.

3. Video handling / synchronizing. These instructions allow to invert or not the video images and synchronize the video frame output rate with the ACE16k image output rate.

5. SOFTWARE TOOLS

The software elements developed as part of the ACE16k-DB system have been used not only during the development of the algorithms to be run in the system, but also during the development of the entire digital core. The software runs under the Linux operating system, although Windows based versions do not pose significant difficulties. The software elements are basically three: the assembler, the loader, and the memory writer.

5.1 Assembler

The assembler program, written in C language, receives input from a file containing the source code and data to be assembled, written using the mnemonic instructions representations described in the previous section. It's a two-pass assembler. In the first pass collects information into the symbol table and in the second pass uses the symbol table and the rest of the source code to generate three output files coded in hexadecimal, printing format and behavioral VHDL code. The assembler parser searches for source code errors in the second pass, while generating the output files.

The assembler provides the following features:

1. Allows the user to assign names to memory locations.
2. Orders the memory loader to store programs in memory blocks according to the programmer's preferences.
3. Allows the use of comments in the source code.
4. Detects 9 types of common errors that fall into four basic categories: Label related errors, argument-type-range related errors, syntax errors and ACE16k memory-file handling errors.
5. Generates hexadecimal (HEX) and print list (PRN) code.
6. Generates a system behavioural description VHDL code for digital post-synthesis simulations.

5.2 HEX file

The HEX file contains one address hex number and 16 data hex numbers per line, representing the future contents of the program memory. Since there is no strict separation of instruction and data memory, the HEX file becomes hard to use in debugging sessions. This file is used by the memory loader to generate executable binary code suitable for being written in the program memory card.

5.3 PRN file

The assembler generates a print listing file, for debugging purposes, containing the source code written by the user with its hexadecimal machine code representations. Label definitions are included, comments are eliminated.

5.4 VHDL file

The VHDL source code generation facility of the assembler is the key feature for fast digital development. The assembler generates VHDL code representing the behaviour of the circuit modules surrounding the FPGA, customized for the specific functions requested to be performed by the source code. These VHDL models reflect the behaviour of the ACE16k chip (only at interface level), of the program memory (containing the programmed binary code), the twin video memories, the clock generator and the push buttons.

The capability of generating this modules was added to help during the design and post-synthesis simulation of the control unit, digital video generator, memory selector and clock divider at VHDL level, allowing the simulation of the synthesized digital core with the behaviour and delays expected from its surrounding circuitry, and saving a huge number of hours of stimuli writing and hardware corrections. To develop and test the digital core, around 100.000 VHDL code lines have been automatically generated by the assembler and simulated by the post-synthesis tools.

5.5 Loader and Program memory writer

The loader program receives HEX files as input and translates the hexadecimal code into a binary form directly writable in the program memory. The binary file generated by the loader is an exact copy of the future contents of the program memory. In order to avoid system hangs the loader fills with zeroes the unused parts of the program memory.

The program memory writer is used to control the hardware that writes the program RAM. It writes the contents of the binary file, generated by the loader, to the program SRAM through a general purpose parallel port (PPI 8255), which in turn is connected to an ISA slot of the PC. After writing the memory, its content is read and verified. The program memory writer allows the user to check the backup batteries. Afterwards, the program RAM card can be unplugged from the parallel port and plugged into the ACE16k-DB system main board.

6. CONCLUSIONS

A programmable stand-alone system, the ACE16k-DB, for real time vision pre-processing tasks at frame rate, based on the ACE16k chip has been presented. The system is based on specific purpose hardware and software elements: a massive parallel mixed-signal processing chip, a digital core synthesized on reconfigurable logic and a software toolbox set developed for this system. The flexible nature of the ACE16k-DB design leaves it open to architectural and functional

improvements such as the inclusion of cache memory and search-decode-execute instruction pipeline stages, as well as an extension of the Control Unit instruction set, in order to make it suitable for image post-processing, decision-making and actuation tasks.

7. ACKNOWLEDGMENTS

The authors deeply appreciate the support from Carlos Jesús Jiménez Fernández and José Miguel Mora Gutiérrez in the use and configuration of the synthesis and simulation tools and the exploitment of the Xilinx FPGAs resources.

This work has been partially financed by the projects IST2001 – 38097 (LOCUST), TIC2003 – 09817- C02 – 01 (VISTA), and ONR-NICOP N000140210884.

References

1. T. Roska and A. Rodríguez-Vázquez (Editors): Towards the Visual Microprocessor, John Wiley & Sons Ltd. (2000)
2. Sonka, Hlavac, Boyle: Image Processing, Analysis and Machine Vision, International Thomson Publishing Inc. (1998)
3. R. Domínguez-Castro, S. Espejo, A. Rodríguez-Vázquez, R. Carmona, P. Foldesy, A. Zarándy, P. Szolgay, T. Sziranyi, and T. Roska: A 0.8μm CMOS programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage, IEEE J. Solid-State Circuits, vol. 32, (1997) 1013–1026
4. G. Liñán, S. Espejo, R. Domínguez-Castro, and A. Rodríguez-Vázquez: ACE4k: An analog I/O 64x64 visual microprocessor chip with 7-bit analog accuracy, Int. J. Circuit Theory Applicat., vol. 30, no. 2/3 (2002) 89–116
5. Angel Rodríguez-Vázquez, Gustavo Liñán-Cembrano, L. Carranza, Elisenda Roca-Moreno, Ricardo Carmona-Galán, Francisco Jiménez-Garrido, Rafael Domínguez-Castro, and Servando Espejo Meana: ACE16k: The Third Generation of Mixed-Signal SIMD-CNN ACE Chips Toward VSoCs, IEEE Transactions on Circuits and Systems—I: Regular Papers, vol. 51, no. 5 (2004) 851-863