Chapter 6

Design of Digital Advanced Systems Based on Programmable System on Chip

Nordin Aranzabal, Adrián Suárez, José Torres, Raimundo García-Olcina, Julio Martos, Jesús Soret, Abraham Menéndez and Pedro A. Martínez

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/66579

Abstract

This chapter fills up an advanced analysis of the state-of-the-art design in programmable SoC systems, giving a critical overall vision for every designer to implement real time operating systems and concurrent processing. The content of the chapter is divided in the next four main sections.

- First the evolution timeline of FPGA based systems is covered from its beginning until the last AP SoC chips. They are complex devices and it is necessary to have a well-known understanding to utilise them in the more efficient form possible.
- The more important advance digital systems structures and architectures are described. The embedded AP SoCs are analysed and main design methodologies are covered, focusing in hardware and co-design strategies.
- In this section is described the development of a real open source application that covers the fundamental parts in the design of a SoC system, ranging from the hardware development until the software design involving the embedded operating system and the user interface application.
- Finally, the system described in the last section is tested in a real scientific experiment and the results are evaluated.

As conclusions the advantages of SoC systems when running an embedded Linux for interfacing FPGA based designs are highlighted.

Keywords: embedded systems, field programmable gate array (FPGA), system on chip (SoC), codesign hardware/software, embedded Linux, real-time instrument

1. Introduction

FPGAs are devices involved in a continuous evolution in order to offer more features and a better performance. The main characteristics of the FPGAs include the following:
High integration capacity that allows to implement complex digital systems in a single circuit.

- Flexible architecture that is easily adapted to each application.
- Have specific resources for performing arithmetic circuits, by reducing delays and making them more efficient.
- Include specific logical resources to generate internal memory units.
- Reconfigurability, it is possible to change the block function almost in real time.
- Generate adaptive circuits.
- Hardware description programming, ability to run multiple applications in parallel.

FPGAs have stopped being a simple architecture because they now represent powerful integrated systems with a lot of possibilities and different families to choose. In recent years, FPGAs have experimented a great evolution since the first important change that carried out when appeared Virtex II Family in 2001 based on look up tables (LUT) until today's new technologies [1].

It is necessary to know FPGA internal logic architecture in order to make the most of their advantages:

- Performance
- Time to market
- Prize
- Reliability
- Maintenance

Thereby, a review of block diagram and internal functionality will be presented in the next sections, beginning by reviewing the timeline since Virtex II family until new SoCs such as Zynq. **Figure 1** shows the more important Virtex II blocks and components.

One of the most important characteristics of Virtex-II device is that it featured a large number of 18 Kb block RAM memories. The block RAM memory is a true dual-port RAM, offering fast, discrete, and large blocks of memory in the device. The memory was organized in columns, and the total amount of block RAM memory depended on the size of the Virtex-II device. As shown in **Figure 2**, it was also formed by distributed RAM block and high-performance interfaces with external memories such as DDR SDRAM, ZBT SRAM, and QDR SRAM.

http://dx.doi.org/10.5772/66579

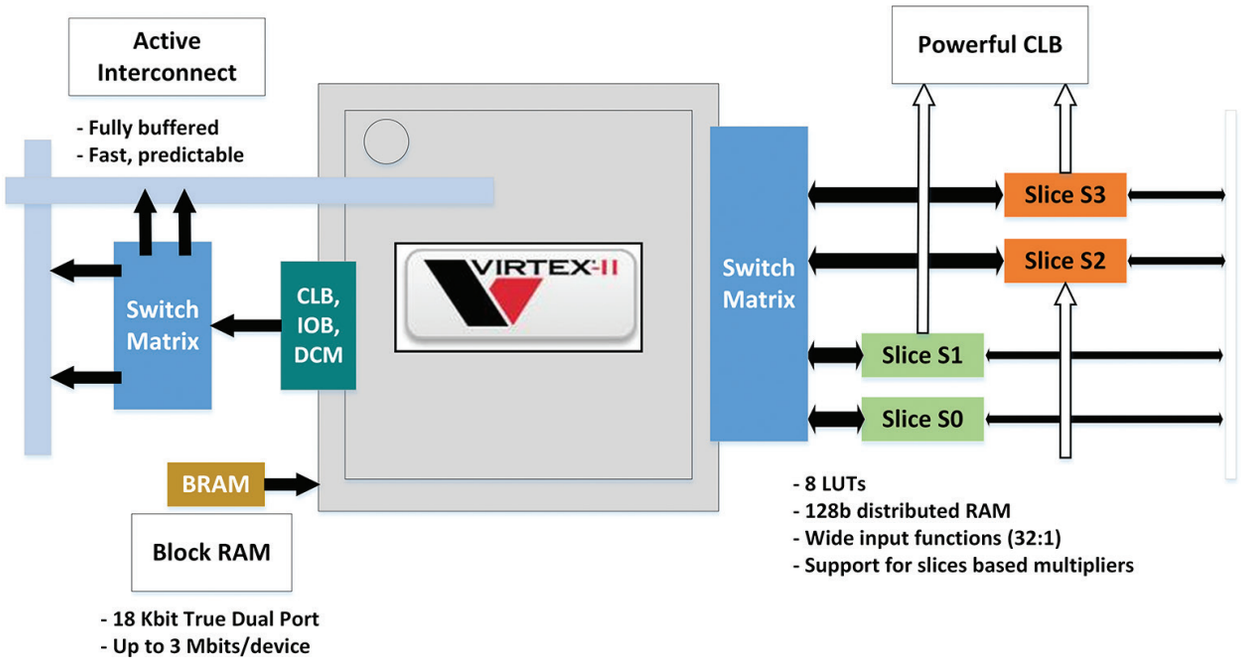

Figure 1. Virtex II internal block diagram.

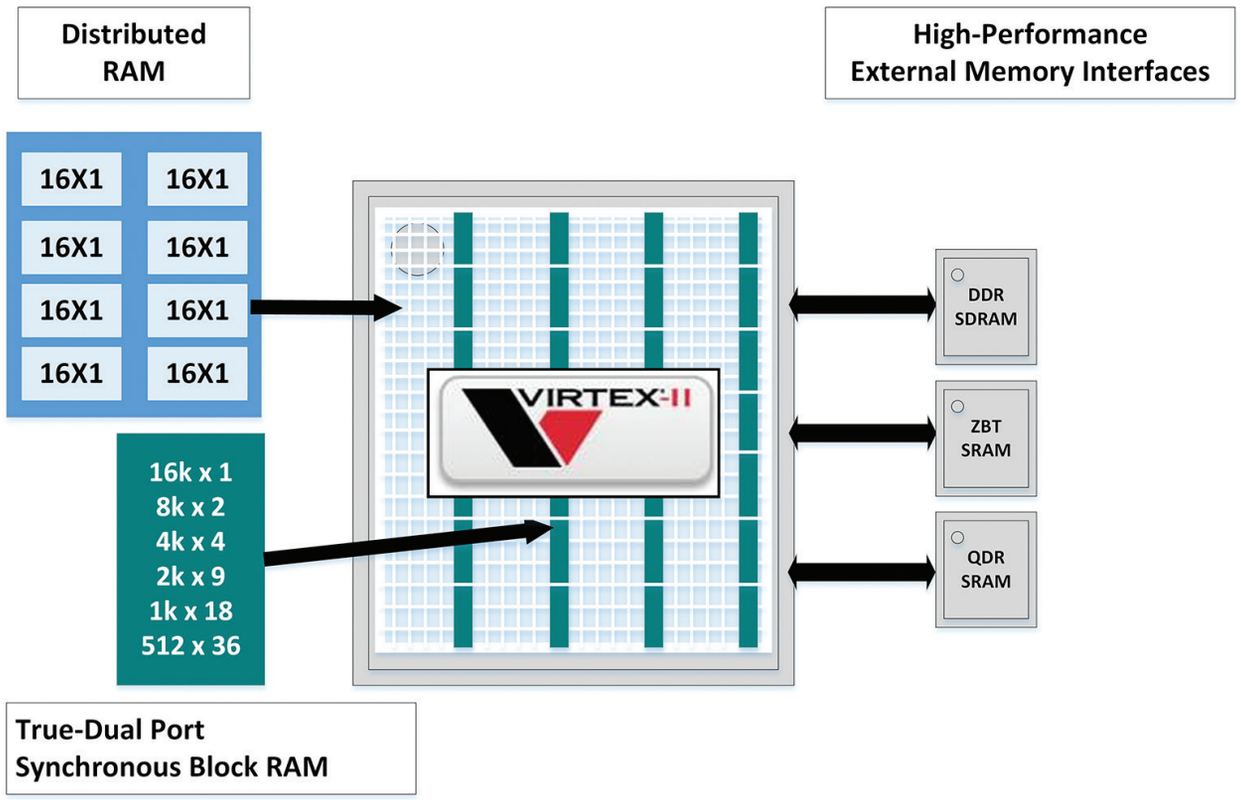

Figure 2. Virtex II memory block distribution and interface.

Another interesting feature included in Virtex II FPGAs was the dedicated 18 × 18 bits hardware multipliers that allow us to implement MAC functions. These modules make possible to carry out two's complement signed operations.

Regarding to Virtual II clock circuits, each FPGA has 16 clock global multiplexors, which manage the clock signal provided from an input port, digital clock manager (DCM), or interconnection local line. The DCMs are provided by external input terminals and allow the FPGA to delay and amplify the clock signal.

Those FPGA models communicate with other systems through input/output blocks (IOB) which are based on two input flip-flops and four output flip-flops, as shown in **Figure 3**. The IOBs include a digital control impedance (DCI) that allows FPGA to set a configurable output impedance in order to adapt the impedance to the PCB track that will be connected. Moreover, it is possible to configure the termination to be compatible with receivers and transmitters in own FPGA, so that, the signal integrity is improved and the reflections are suppressed.

Once described the more important features of the Virtex II, one of the high-end FPGAs that supposed a great evolution in the integrated systems market, next FPGA evolutions will be explained.

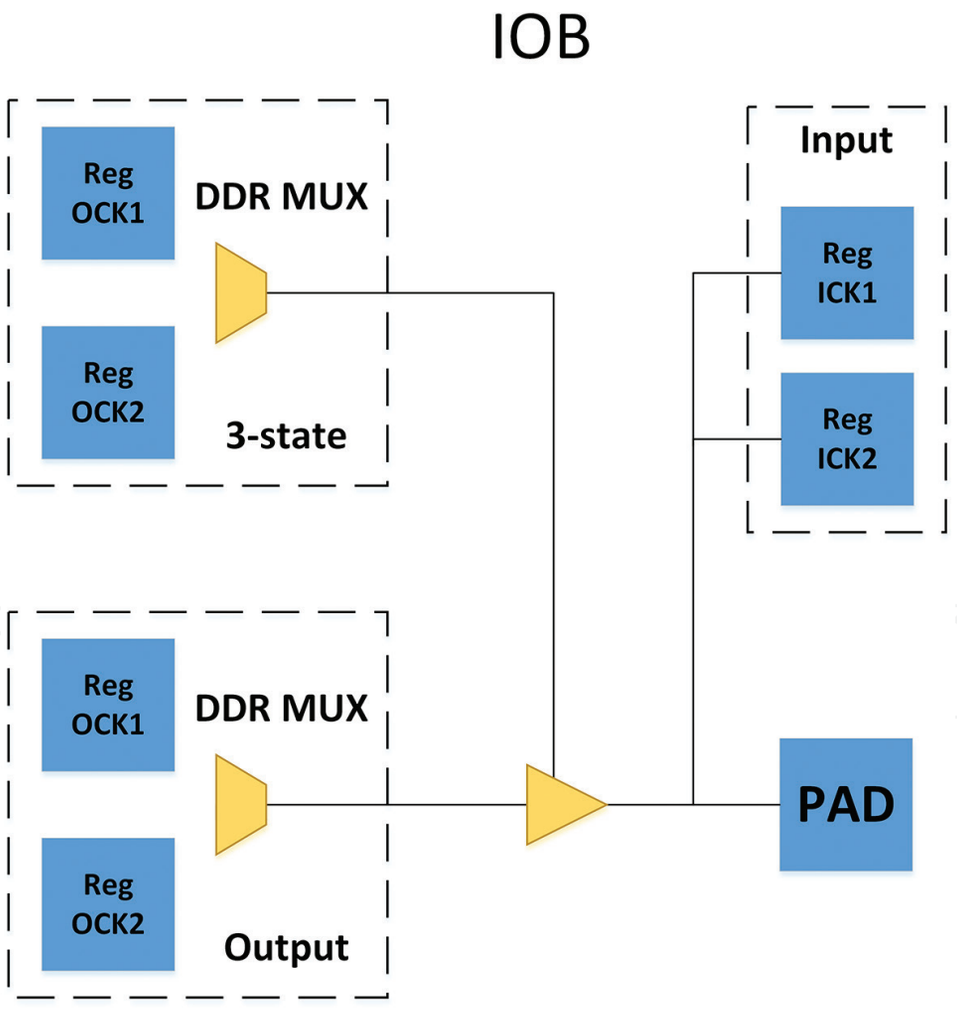

Figure 3. Input/output blocks architecture.

http://dx.doi.org/10.5772/66579

The next model was the Virtex II Pro which added to its predecessor the RocketIO multi-gigabit transceiver (MGT) blocks able to manage very-high data rate (2488–10,312 Gbps) through optical fiber or Gigabit Ethernet. Furthermore, this model could work with selectable 8, 16, and 32 bit buses, included up to 24 transceivers, integrated up to four PowerPC hardware processors that worked at 300 MHz and added more hardware multipliers, I/O terminals, and internal memory. These features are shown in **Figure 4**.

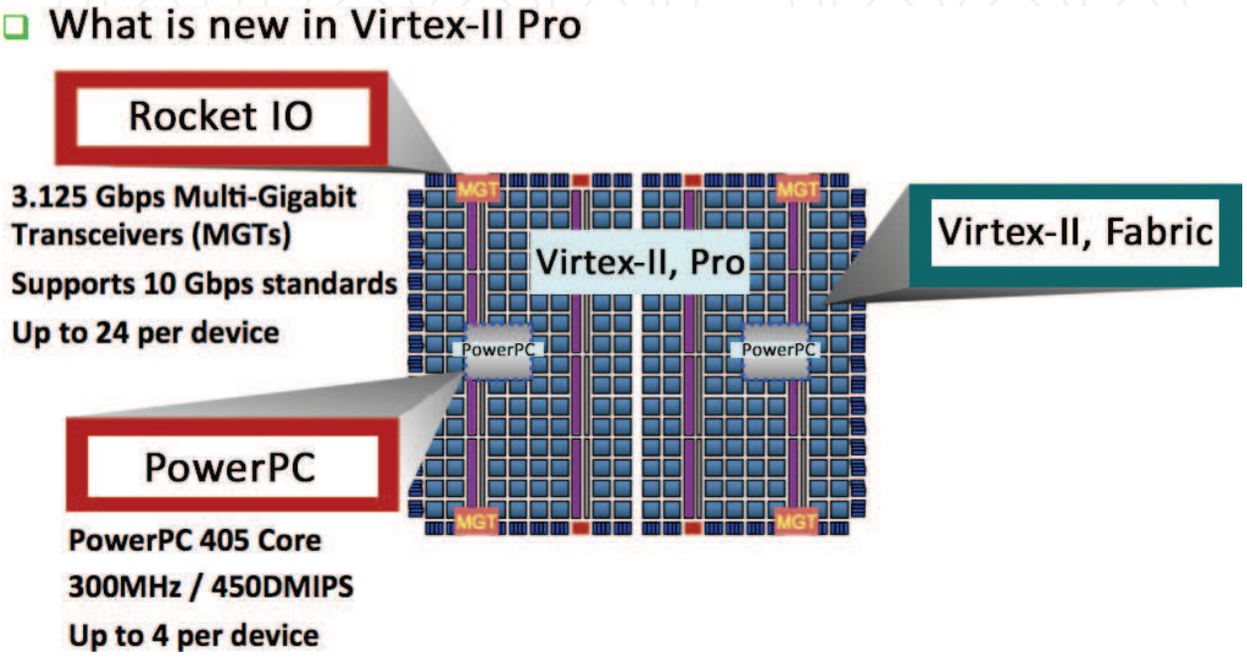

Figure 4. Improvements included in Virtex-II Pro model.

Next evolution came with the Virtex 4 and 5 models being that they added dedicated digital signal processors (DSPs) which made possible to carry out more complex computation with a better performance, as shown in **Figure 5**. Moreover, they included other improvements such as:

- Phase-matched clock dividers (PMCD), serial-to-parallel and parallel-to-serial interfaces integrated in the I/O terminals.
- Ethernet media access controller (TEMAC).
- BRAM memories set as FIFOs.
- Regional clock buffers, an advanced clock distribution network.
- Multi-gigabit transceivers up to 11.1 Gbps.
- Voltage and temperature monitoring (Virtex 5).
- Less power consumption.

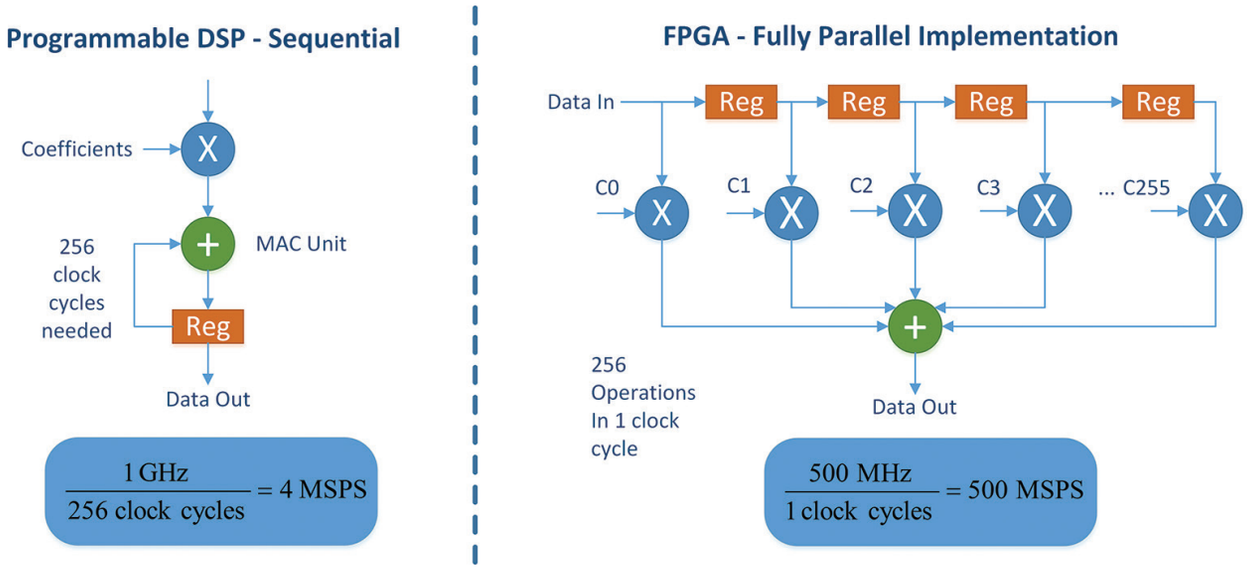

Figure 5. Performance comparison between programmable DSP and new-dedicated DSP.

Nevertheless, the change that stirred up the Xilinx FPGAs took place with the new families focused on low power consumption and high performance. Thereby, it disappears the concept high cost family (Virtex) and low cost family (Spartan), and Xilinx FPGA series are classified as shown in **Figure 6**.

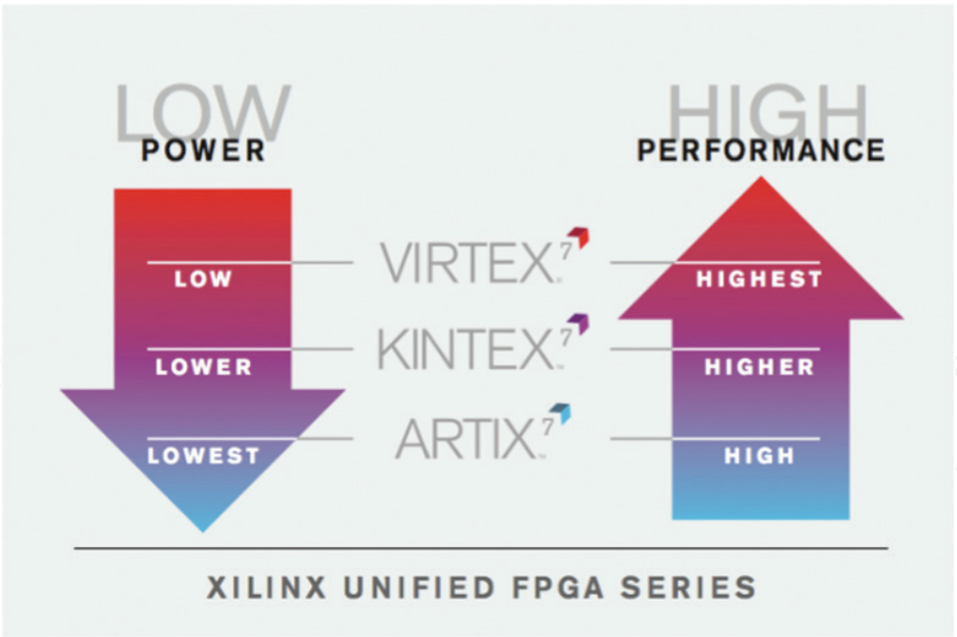

Figure 6. New Xilinx FPGA series range (picture courtesy of Xilinx).

This change was further started immediately after Xilinx promoted to combine a FPGA with external microprocessors by combining various components in a single chip. The new proposed technology was the Zynq-7000 line of 28 nm SoC devices that combine an ARM core with a FPGA [2], as shown in **Figure 7**. This was also joined by a change in the software

http://dx.doi.org/10.5772/66579

tool because new models had to be programmed with Vivado Design Suite instead of the traditional ISE design platform given that it had not been developed to handle the capacity and complexity of designing with a FPGA with a hardware microprocessor core. The new software tool includes high-level synthesis functionality that allows engineers to compile the co-processors from a C-based description.

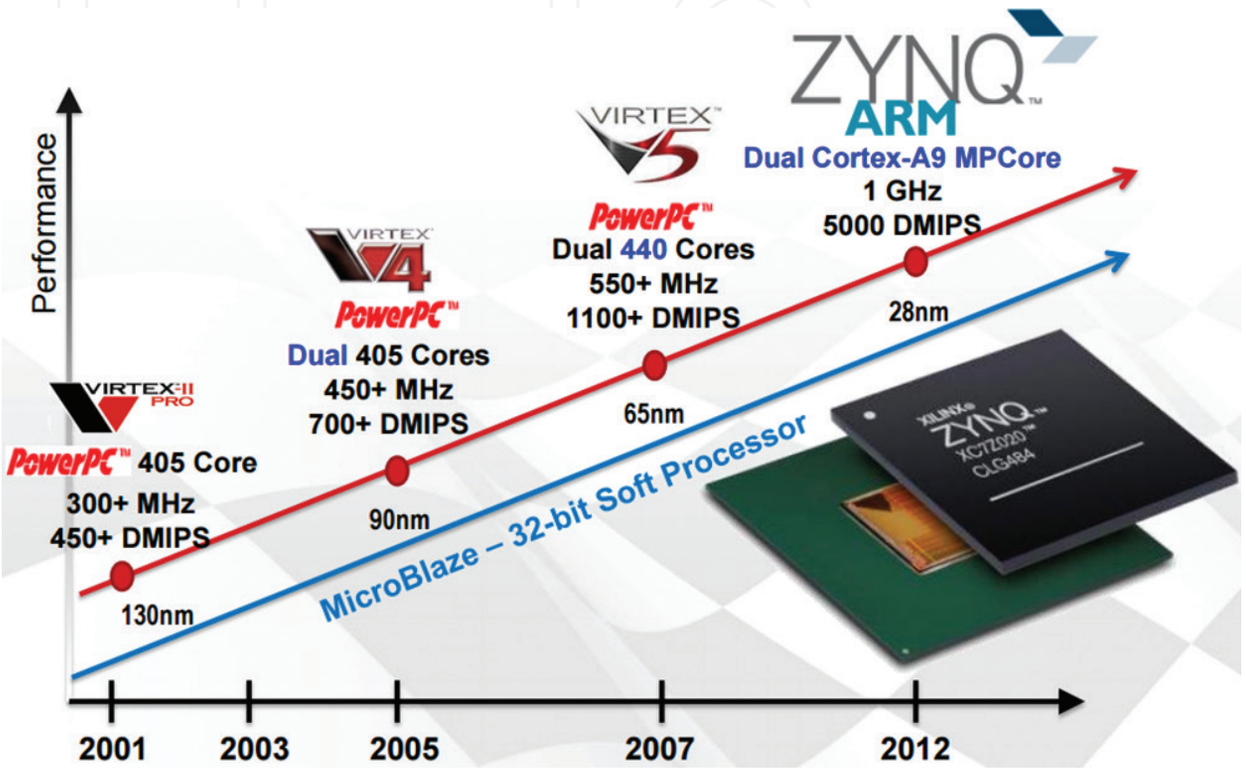

Figure 7. Processors evolution of Xilinx FPGAs (picture courtesy of Xilinx).

The Zynq-7000 family of system-on-chip (SoCs) represents a new concept because it integrates a complete Cortex-A9-processor-based 28 nm system. The Zynq architecture is different from other devices that combine both programmable logic and embedded processors because this kind of systems are focused on the processor instead of FPGA platform, as shown in **Figure 8**. For software developers, Zynq-7000 appears the same as a standard, fully featured ARM processor-based system-on-chip (SOC), booting immediately at power-up and capable of running a variety of operating systems independently of the programmable logic. Next generation of Xilinx SoCs was introduced by the Zynq-7100 model because it integrates digital signal processing (DSP) to meet emerging programmable systems integration requirements.

Therefore, the apparition of a large amount of specific new components, a lot of parameters such as input/output technology, clock signals, DSP blocks, flexibility, scalability, performance, integration, software or hardware processor, consumption or cost have to be taken into account to choose which is the better solution for each case as shown in **Figure 9**.

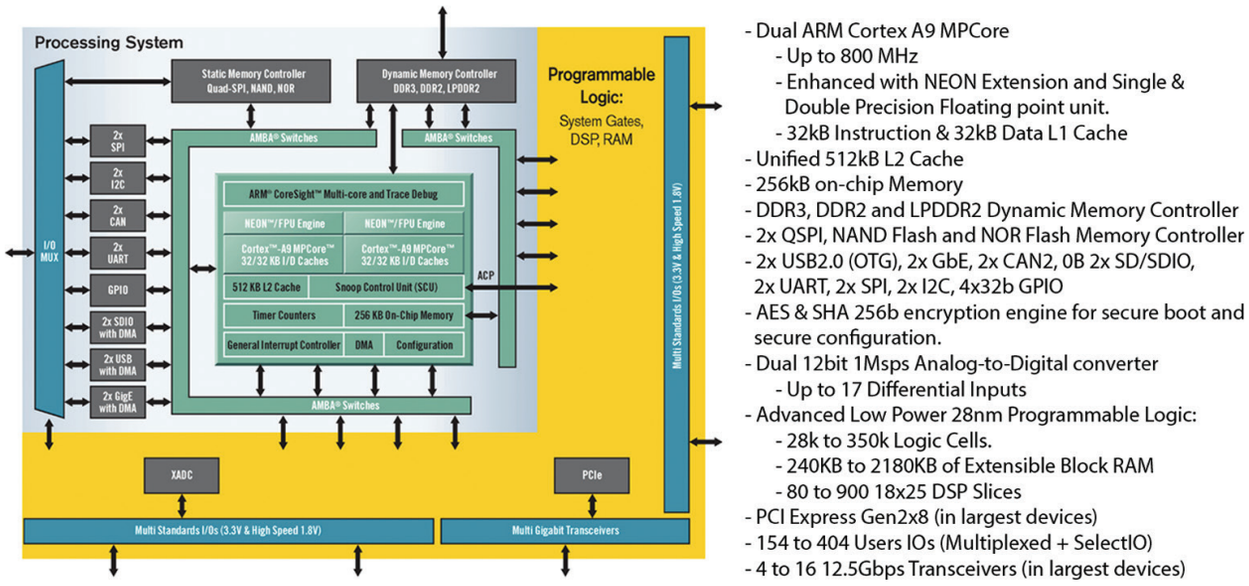

Figure 8. Architecture of SoCs based on programmable logic part (orange) and processing system such as ARM (blue) (picture courtesy of Xilinx).

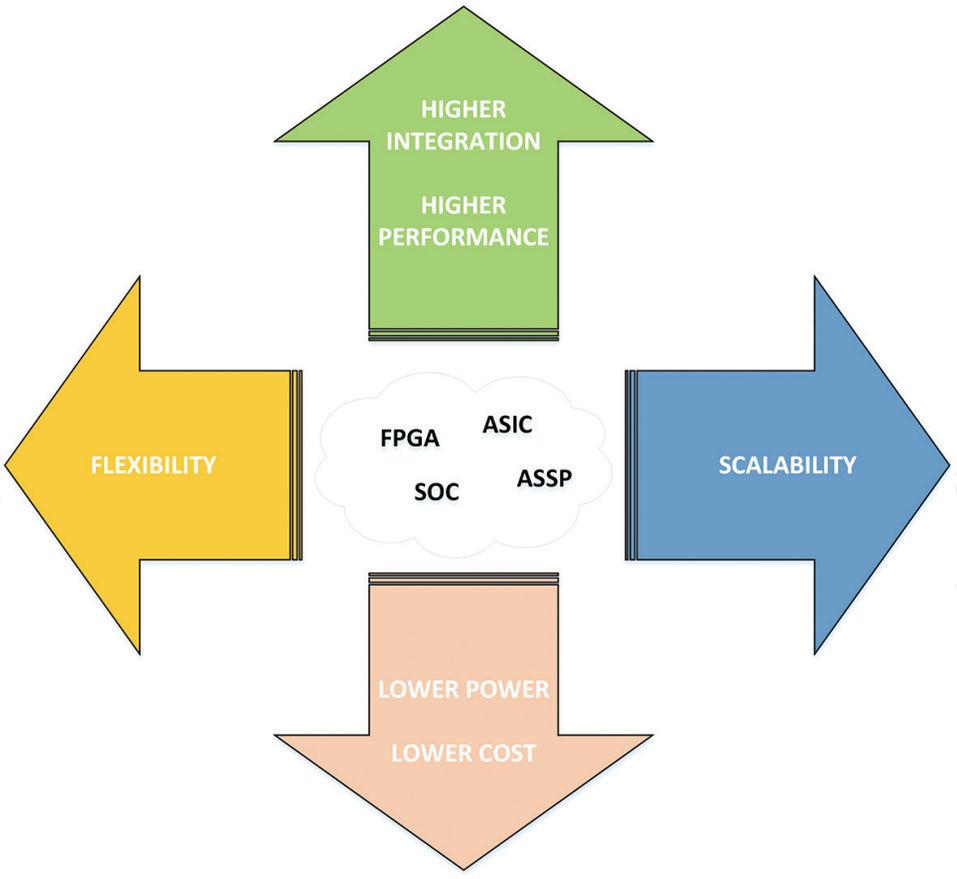

Figure 9. SoCs features.

2. Programmable SoC structures and architectures: hardware and software codesign methodology

2.1. Structures and architectures

Advance digital systems are those in which the core is a standard computational device or a combination of them. **Figure 10** shows two examples of this kind of devices integrated into an electronic design. Nowadays, the more important computational devices that take part in the front line of the embedded design market are the following:

- Field programmable gate array (FPGA): A digital integrated circuit with programmable logic to be configured by the designer.
- Application-specific integrated circuit (ASIC): A customized embedded circuit designed for a specific application, rather than a general-purpose.
- Microprocessor: An embedded circuit that incorporates a central processing unit (CPU) functions and operates on numbers and symbols represented in the binary numeral system. It is a clock driven, multipurpose, register-based, programmable electronic device that accepts binary data as input, processes it according to instructions stored in its memory, and gives results as output. The logic can be either combinational or sequential.
- Microcontroller: An integrated circuit embedding a processor core, programmable input/output peripherals and memory.
- Digital signal processor (DSP): A microprocessor with an optimized architecture in order to perform the computational operations in digital signal processing.
- Complex programmable logic device (CPLD): An integrated device with programmable logic that derives between programmable array logics (PALs) and FPGAs and the architectural characteristics of both.

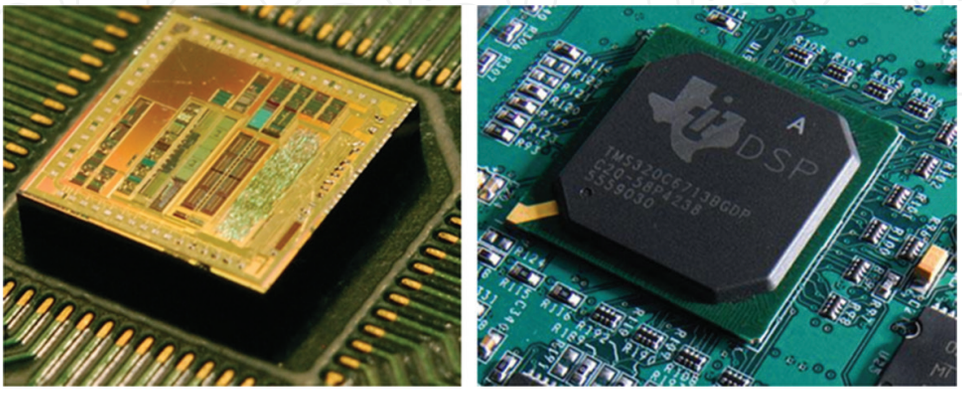

Figure 10. Examples of advance digital systems.

To confront the development of those advance digital systems, two methods can be differentiated. The functionality of the desired application can be reached either by a software or hardware method. However, it is fundamental to have a remarkable understanding of both methodologies in order to obtain the best performance and effectiveness in the designs.

Software method:

- Driven by standard circuit processors (microprocessor, microcontrollers or DSPs).
- Involves programming languages associated with a fixed set of instructions.
- System flow runs sequentially.

Hardware method:

- Based on hardware circuits customizable by the user (CPLDs, FPGAs, and ASICs).
- Involves hardware description languages.
- Implements concurrent processing.

The software method is limited by the sequential flow of the instructions and the particular architecture. While the hardware method is limited by the difficulty that implies the design using the hardware programming languages, the time required for develop specific hardware systems and the high cost of those designs. Currently, in order to overcome the drawbacks of both methods, alternative methods must be determined.

This intermediate option is the most utilized nowadays and is known as integrated system, embedded system, or system on a chip (SoC). A SoC is an integrated circuit that includes a combination of the elements below in one single chip:

- One or several operating units (microprocessors/DSPs/FPAs…).
- Different standard interface circuits (UART, SPI, I2C, Ethernet…).
- One of the several memory units (RAM/FLASH…).
- An analog-to-digital converter (ADC).
- Different specific circuits for the application (audio/graphics/automotive…).

Currently, the options that enable to develop an application based on a programmable system on a chip such as the devices shown in **Figure 11**:

- Those who are already in the market and integrate a microcontroller, a programmable digital and analogic parts and communication ports: FIPSOC (Sidsa), Fusion (Actel), FPSC (Lattice), and PSoC (Cypress).
- Those that through a FPGA of high logic capacity permit to integrate a microcontroller and customizable communication ports: Xilinx and Altera.

The increase in the capacity of integration in the manufacturing of the chips has allowed the development and evolution of the embedded systems. **Figure 12** depicts an example of how the integration capacity has been increased over the time.

http://dx.doi.org/10.5772/66579

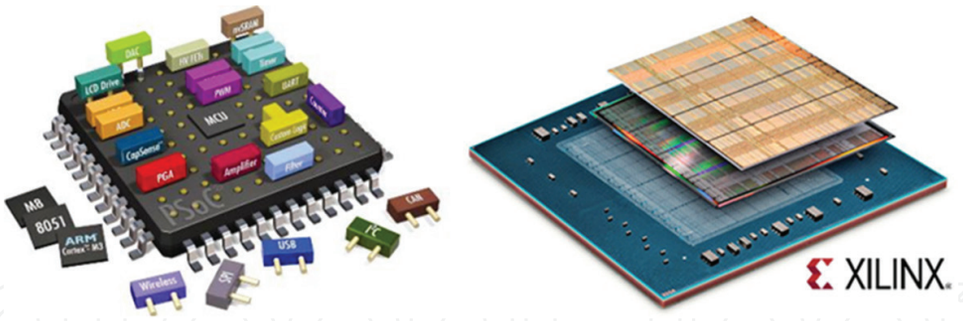

Figure 11. Examples of SoCs (picture courtesy of Cypress and Xilinx).

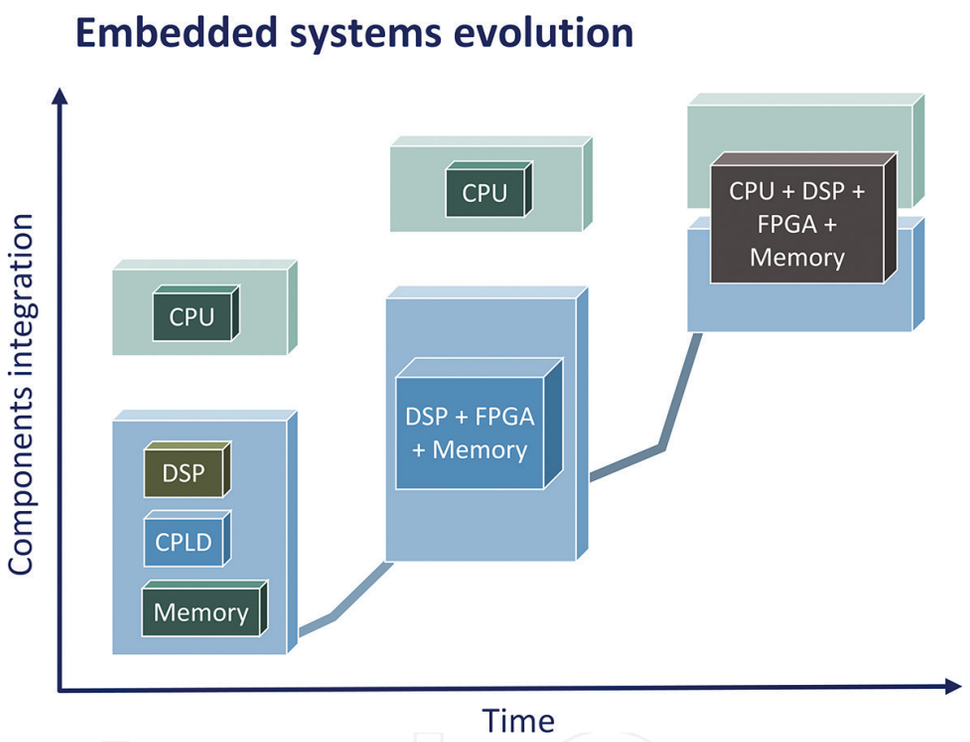

Figure 12. Embedded systems evolution in terms of components integration.

In a design of an application using an embedded system, the following factors need to be taken into consideration:

- The "hardware" is different for each application.
- Some applications may require an operating system (RTOS).
- A compact code implementation is desirable to reduce the size of the program memory.
- A combination between high-level (C) languages and low-level languages (VHDL) is required in order to optimize the processing speed.

In this section, among the presented embedded systems, we will be addressed in more detail the ones based on FPGAs of Xilinx, also known as systems on a programmable chip (SoPCs).

Xilinx, one of the main manufacturers of FPGAs, provides two alternatives to carry out a SoPC-based design:

- Hardware microprocessor block. PowerPC of 32 bits in FPGAs Virtex 2 pro, Virtex 4 FX, and Virtex 5 FXT ARM Dual-Core Cortex-A9 of 32 bits in Zynq.
- Software microprocessor block. Picoblaze of 8 bits in any FPGA. Microblaze of 32 bits in the families Spartan, Virtex, and the new Artix and Kintex.

For a better understanding of the different architectures of the hardware or software microprocessors involved in SoPC designs, various diagrams are presented in **Figures 13–16**. It is always important to take time studying and comprehending the architecture of each microprocessor for being able to choose the most effective solution for the desired application.

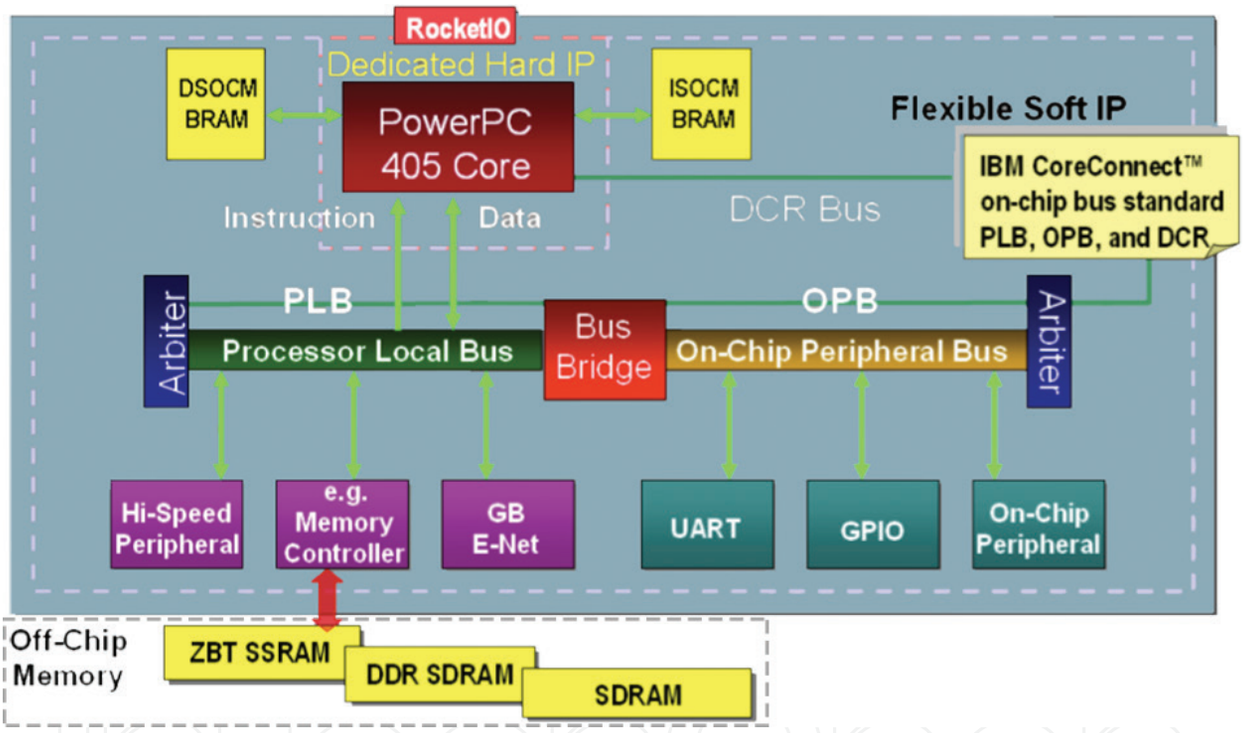

Figure 13. Architecture of an embedded system "hardware" based on PowerPC.

Concretely, from the previous architecture diagrams, it is relevant to recognize the possibilities, advantages, and disadvantages that offer hardware and software microprocessors.

There is only one possible option when choosing a given hardware microprocessor block:

- Commercial microprocessor core designed to be implemented in FPGAs. Advantages: the alternative of having a commercial microprocessor widely used before in the market (8051, PIC, …). Lower development time.

  Disadvantages: higher cost and a fixed architecture of the microprocessor.

http://dx.doi.org/10.5772/66579

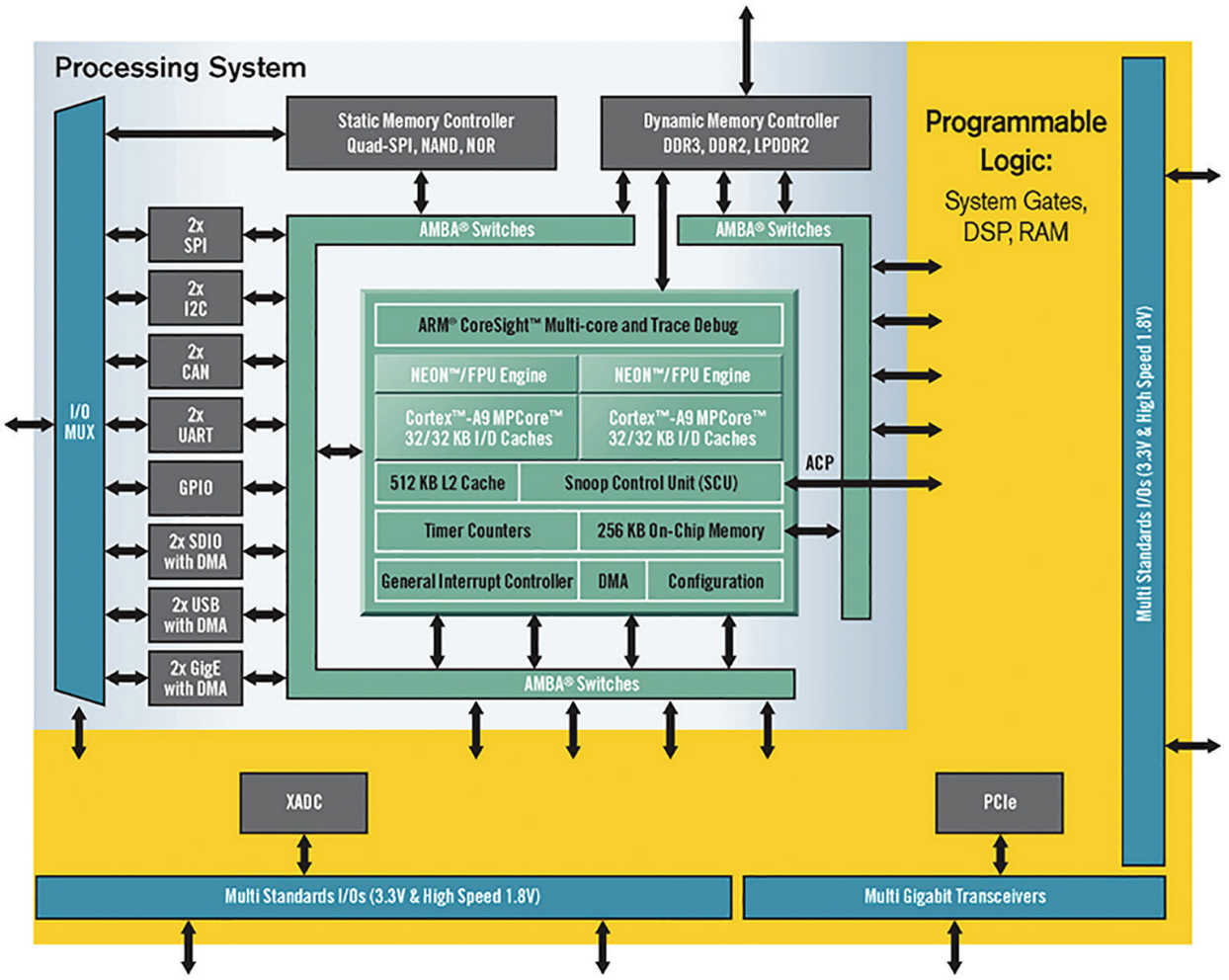

Figure 14. Architecture of an embedded system "hardware" based on ARM (picture courtesy of Xilinx).

The possible options when choosing a software microprocessor block are:

- Microprocessor core designed and customized by the user.

  Advantages: totally customizable architecture depending on the application. The source code of the microprocessor is available to perform modifications. Lower cost and commercialization possibilities.

  Disadvantages: limitations in the architecture.

- Microprocessor core designed by the manufacturer of the FPGAs.

  Advantages: the possibility of having a microprocessor utilized by a high number of users. Exchange of free-distribution programs and peripherals. Lower cost and development time.

  Disadvantages: limitations in the architecture.

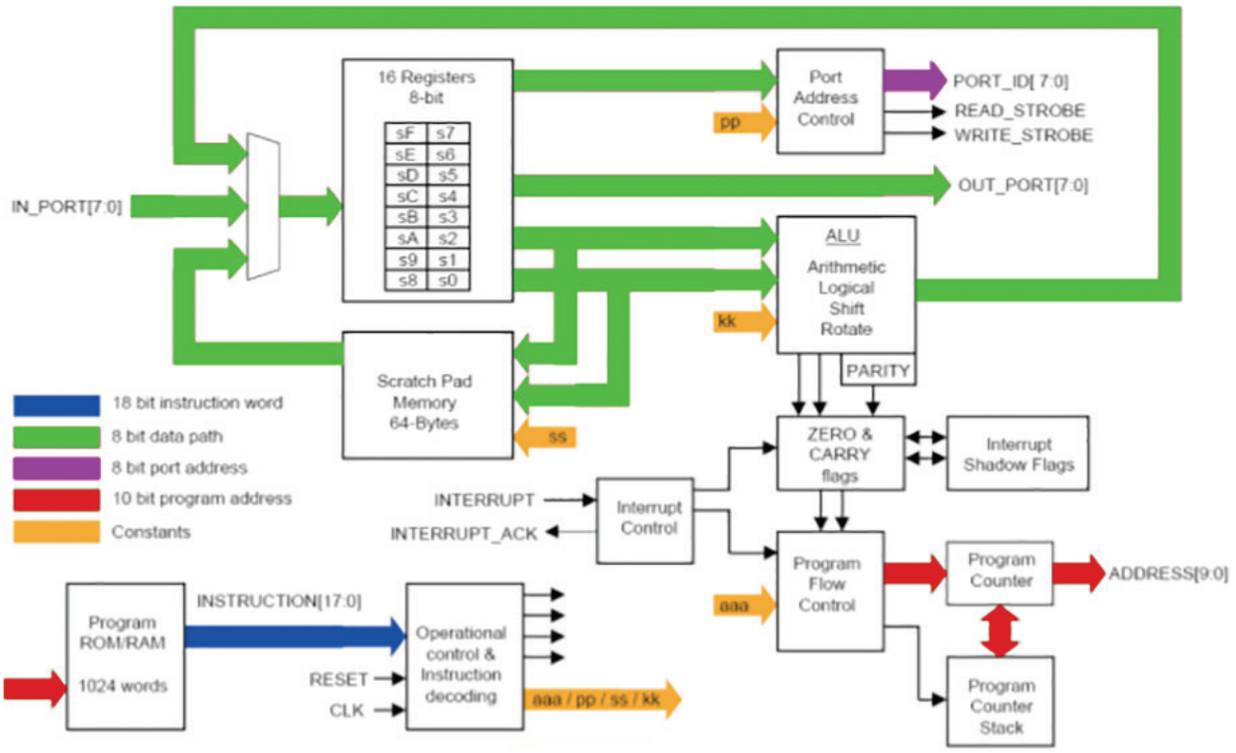

Figure 15. Architecture of an embedded system "software" based on PicoBlaze (picture courtesy of Xilinx).

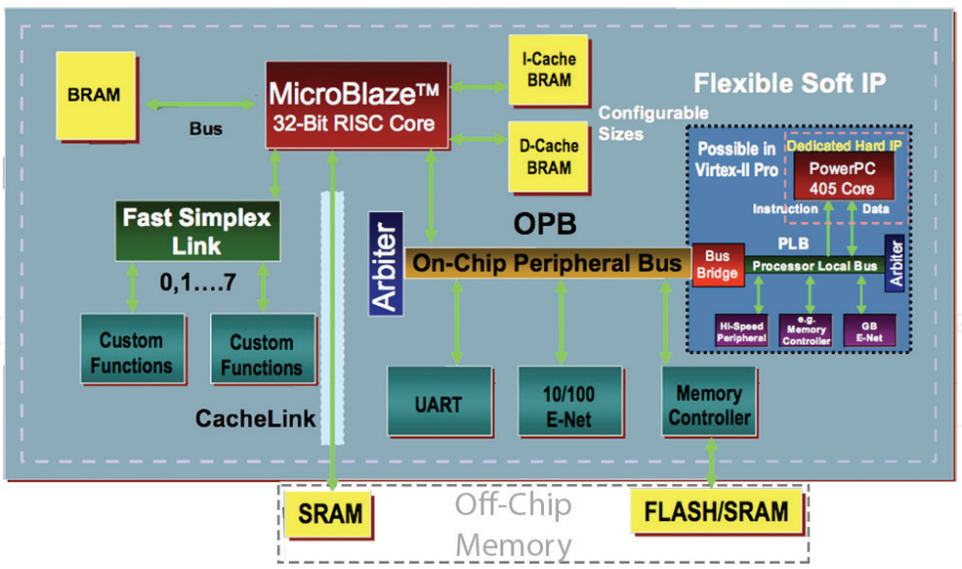

Figure 16. Architecture of an embedded system "software" based on MircoBlaze.

2.2. Codesign hardware/software

After understanding the embedded system, it is important how to stop and comprehend the design. For those designs that involve a high complexity, it is required to work on the

http://dx.doi.org/10.5772/66579

necessary hardware, both for the microprocessor and its peripherals. Accordingly, it is necessary to know what program is going to be executed by the microprocessor and which additional components are involved in the application.

This process is usually known as codesign hardware/software [3] and basically can be defined as: the concurrent design of developers, part of the same team, to complete hardware and software tasks that are involved in an embedded system. Methodologies and tools are utilized which consider hardware/software iteration.

Figure 17 shows a basic codesign flow diagram in which the next stages are involved.

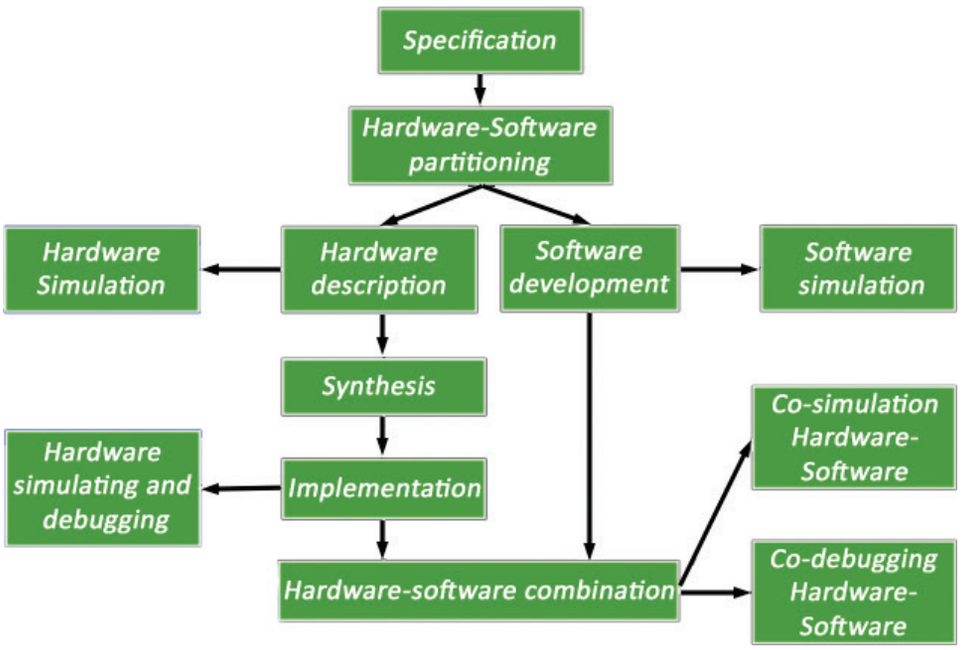

Figure 17. Processors evolution of Xilinx FPGAs (picture courtesy of AVNET).

Specifications:

Define the features of the system that is going to be developed. Hardware/software partitioning determines which functions are implemented through software routines in the embedded microprocessor and which through specific hardware circuits. In the market, certain commercial tools are available for high cost designed specifically to conduct this function.

Hardware description:

- Chose the suitable microprocessor and FPGA for the application.
- Design of the peripherals using HDL languages.

Software description:

- Development of the application in Assembler/C language. Compilation.

Hardware simulation:

- Through simulation programs that normally are part of the selected development tool.

Software simulation:

- Through simulation programs of the implemented programming language.

Co-simulation hardware/software:

- Tools such as EDK, ChipScope, synthesis, and implementation. The specific tool provided by the FPGA manufactured is employed (ISE).

A basic codesign, for example, the control of a text liquid crystal display (LCD), with two rows and 16 characters is presented in **Figure 18** which show the different options that could be implemented.

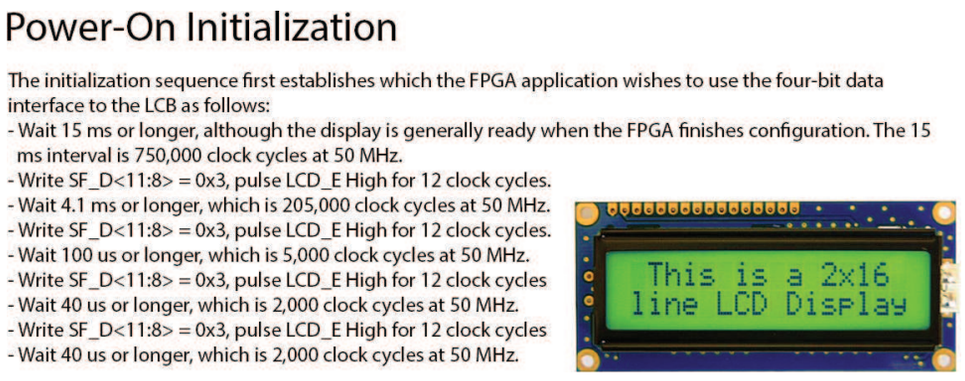

Figure 18. A basic example of codesign.

Hardware/software partitioning:

The operation slowness of the LCD and the complexity of the required control configuration sequences make the application very appropriate to be managed by a microprocessor. Nevertheless, slow operation time forces the microprocessor to attend and control the display for long periods of time, so it would be suitable release the microprocessor from that task.

First option (software):

Implement the control of the display totally by software, adding the minimum hardware. The software of the microcontroller will be in charge of sending and receiving instructions and data to the LCD (low-level routine) and of sending messages (high-level routine).

Second option (hardware):

Implement the control of the display totally using specific hardware, adding the minimum required software for the LCD. In this scenario, the device is totally managed by the hardware. In this option, a FPGA is needed to control the display, which implies higher costs and a great logic area.

Third option (codesign):

The most critical tasks, above all in time, will be implemented in specific hardware while the rest will be programmed through C in order to be executed by the microprocessor. The hard-

ware acts as a microprocessor. In general, working with advance digital systems is the best strategy. An example of codesign is shown in **Figure 19**.

```
void XromLCDPrint2Strings(char *line1, char * line2) {
    int i=0;
        XromLCDSetLine(1);
        for(i=0; i<16; i++){
            if(line1[1])
                XromLCDPrintChar(line1[1]);
            else break;
        }
        XromLCFSetLine(2);
        for(i=0; i<16; i++)[
            if(line2[1])
                XromLCDPrintChar(line2[1]);
            else break;
        }
    return;
}
```

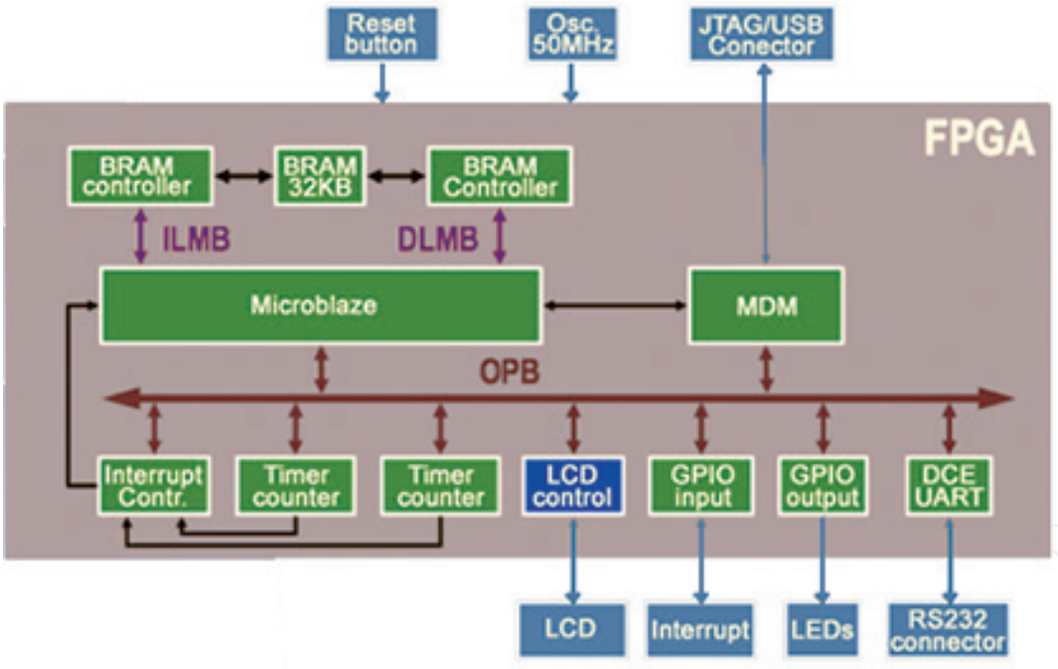

Figure 19. Example of codesign: LCD interface application.

3. Implementation of a real SoC application: description of a programmable SoC real application based on Linux with web server, database, and concurrent processing

This section is presented a real application based on a Zynq®-7000 All Programmable SoC (AP SoC) which has dual-core central processing units (CPUs) and a field programmable gate array (FPGA) in one chip, a versatile architecture that enables to lower costs and enhances the efficiency in regard to another system of analogue characteristics. The whole

implemented software and hardware development that is described in this section covers from the user interface application to the free distribution operating system based on an embedded Linux [4] [5], which avoids closed source software and license costs. Additionally, extra open software is compiled to run in the specific system in order to produce a more efficient and reliable application. A web server based on Apache 2 provides the remote control and monitoring functionality, and the data storage and management is performed by a database application based on MySQL engine. Moreover, for a dynamic iteration between the web user interface and the database, a PHP server scripting language is compiled to run on the operating system.

The embedded system is designed to manage the Geowire, a novel electromechanical device that measures the temperature inside the pipes of vertical borehole heat exchangers (BHEs) throughout the thermal response test (TRT). The TRT is the standard method to quantify the thermal characteristics of borehole surrounding subsoil by measuring the temperature evolution at inlet and outlet. However, it assumes that the homogeneous isotropic subsoil calculates an average conductivity value for the overall geological domain. The Geowire provides additional information during the TRT by recording a series of depth-dependent temperature profiles during the TRT, which allow the system to identify the heterogeneity of ground stratigraphy. Thus, the minimum depth of the drilling for the maximum heat transfer can be calculated to save installation costs and build more optimized BHE.

3.1. Description of the device involved in the application

The Geowire measures the temperature inside the geothermal pipes by controlling the vertical displacement of a wired digital sensor. The sensor is connected to a cable furling inside a watertight case while a slip ring allows the transmission of the signal from the rotating structure to the Zynq board. The temperature probe goes out from the fluid output pipe connection and a servomotor rotates the furling to release or collect cable while a weight maintains it tightly by the effect of gravity. Before the cable goes outside of the case, it is guided between a roll with a magnetic encoder that transmits the signal outside the case to measure the displacement of the wire. Then, the software application is designed to control the servomotor and calculate the exact position of the sensor. In this way, a user interface makes possible to define customizable acquisition sequences by indicating the depth-dependent points of interest, the sampling time and the number of temperature samples per each point. Once the data for the acquisition process is defined, the device will begin to automatically sample and storage the data while the progress can be followed remotely in real time through the user interface.

The device was developed in such a way that it can be easily incorporated in geothermal pipes utilized during the TRT with or without water flow. The temperature inside the pipes can be measured with a maximum spatial resolution of 1 cm, a maximum temperature resolution of 0.0625°C, and an acquisition time smaller than 1 second. An electromechanical limit switch is employed to determine the starting point of the measurement path. This is activated when the weight used to sink the sensor pushes it. The provided signal from this switch is used to calibrate the measurement point every time when the sensor goes down and up. Also, it is

http://dx.doi.org/10.5772/66579

connected with the driver of the servomotor as an additional security measure to stop the motor and avoid the weight and the probe to roll inside the device enclosure. **Figure 20** shows a representation of Geowire enclosure parts.

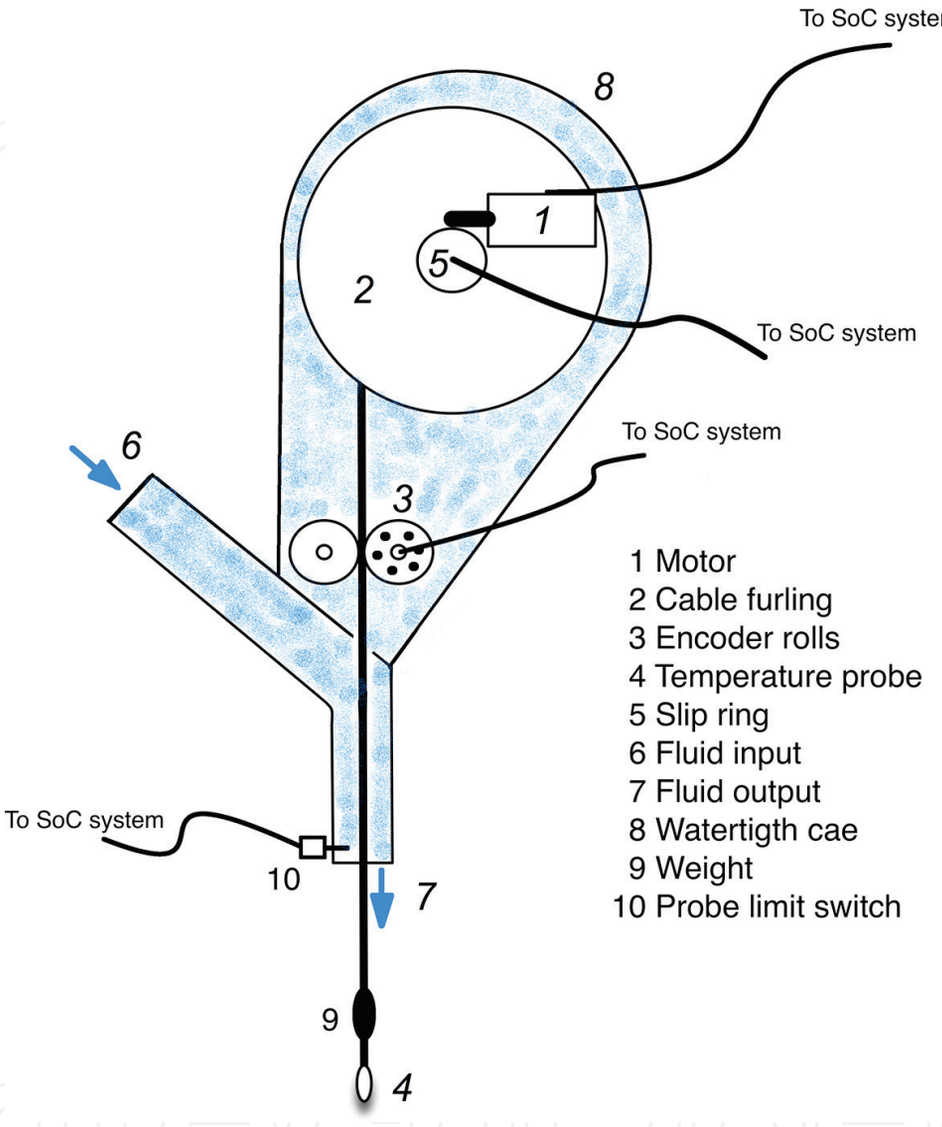

Figure 20. A representation of the different parts that comprehend the Geowire enclosure.

3.2. Application system architecture

The temperature measurement instrument described in the previous section is controlled by a μClinux OS that has been embedded in the dual-core ARM processor of the development board Digilent Zybo Zynq-7000. The board has been implemented as the CPU of the system that manages the performance of the secondary elements that compose the device through a user interface application. One of the most interesting characteristics of this kind of platforms is the possibility of using PMODs (peripheral module) through connectors with some fixed pins from the ARM and ADCs and reconfigurable pins from the programmable logic. So, eases the redesign tasks where any upgrade or modification is required. In

this application, these connectors are utilized to communicate with the motor driver, read an encoder, manage a real-time clock (RTC), and control a driver to read a digital temperature sensor. Concretely, these peripherals are connected with the SoC in order to carry out the next tasks:

a. Motor driver: It is based on a one-quadrant digital servo amplifier driver and its main task is to provide protection against possible current overloads, power surge or brown-out, polarity inversions and shortcuts in the motor. This driver integrates a microcontroller which follows the parameters sent by Zybo board processors. Thereby, by using an RS232, communication protocol is possible to manage the motor control settings. Thus, the CPU of the system regulates the speed either in clock wise (CW) or counter clock wise (CCW) directions and reads configuration, information or alarm registers from the driver.

b. Encoder: A magnetic encoder measures the angular movement of a roller that is rotated when the wire of the temperature sensor comes out from the Geowire to be inserted in the geothermal pipes. The wire is guided through two rollers inside the enclosure of the device to ensure the rotation by friction. When the roller rotates, the encoder generates several digital pulses that are sent to a core implemented in VHDL that measures the distance traveled by the sensor. An important advantage of this encoder is the suppression of mechanical contacts that may originate tear and wear problems. Furthermore, this technology holds the encoder electric connections shielded against water or humidity so that the water can flow inside the device enclosure without damaging the electric connections.

c. RTC: The PMOD connected to the port of the Zybo board is based on a real-time clock which contains a battery to maintain the operating system time upgraded when the system wakes up after a shut down. In this way, the RTC provides the time and date to the systems, every time it sends a request via an I^2C (inter-integrated circuit) communication protocol.

d. Temperature sensor: This application is very important to know the exact temperature inside geothermal pipes and does not lose information during the long communications between the sensor and the CPU. Thereby, a digital temperature sensor with a resolution of 0.0625°C and an acquisition time smaller than 1 second was used. The sensor implements a 1-wire communication protocol with a parasite power that derives from data line, hence only two wires are need to operate. A typical and complicated problem that may appear in this kind of communications where the cable length is quite large is related to electromagnetic interference (EMI). For that, the design of a PMOD module board to host an I^2C to 1-wire bridge device was carried on, a chip that comes with a built-in electronic configuration in order to reduce noise and perform more reliable communications.

e. Temperature probe limit switch: An electromechanical limit switch inside the pipe of the Geowire is activated when the weight used to descend the probe pushes it. Concretely, the produced signal from the limit switch is read through one of the GPIO ports of the ARM

http://dx.doi.org/10.5772/66579

in the Zybo in order to calibrate the initial position of every measuring down-up path of the sensor. The motor driver also detects this signal as a security measure that does not allow the motor to continue rolling up, avoiding the weight and the sensor to get inside the device enclosure.

Figure 21 shows the architecture application diagram and **Figure 22** shows the SoC device with peripherals connected.

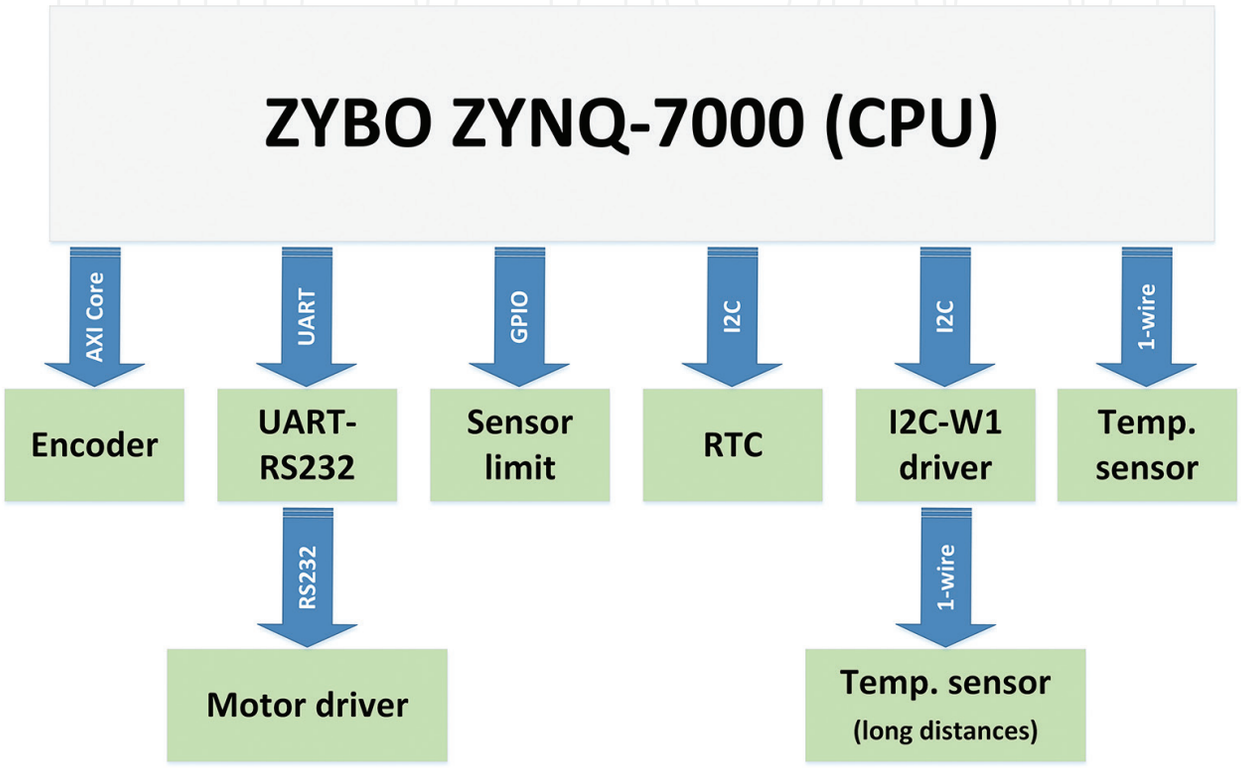

Figure 21. Implemented system architecture.

3.3. Hardware implementation

In order to implement the described application, the Zybo development board based on the Xilinx Zynq-7000 AP SoC architecture was employed. It integrates both a dual-core ARM Cortex A9 processor and Xilinx 7 series FPGA. SoCs provide a very interesting solution due to the combination of both FPGA reprogram ability and flexibility, and powerful ARM processors. Thus, the processor can run operating systems and manage parallel FPGA hardware processing.

Specifically, the processor runs with a clock of 650 MHz and the FPGA is connected to a clock core of 100 MHz. Regarding to communication between the processors, the FPGA cores, and the memories, the AXI-4 interface (Advanced eXtensible Interface) was selected. This interface belongs to the fourth generation of the ARM advanced microcontroller bus architecture (AMBA) interface specification [6]. Furthermore, AXI-4 makes easier the integration of the FPGA IPs and reduces the design effort due to it is optimized in order to achieve more

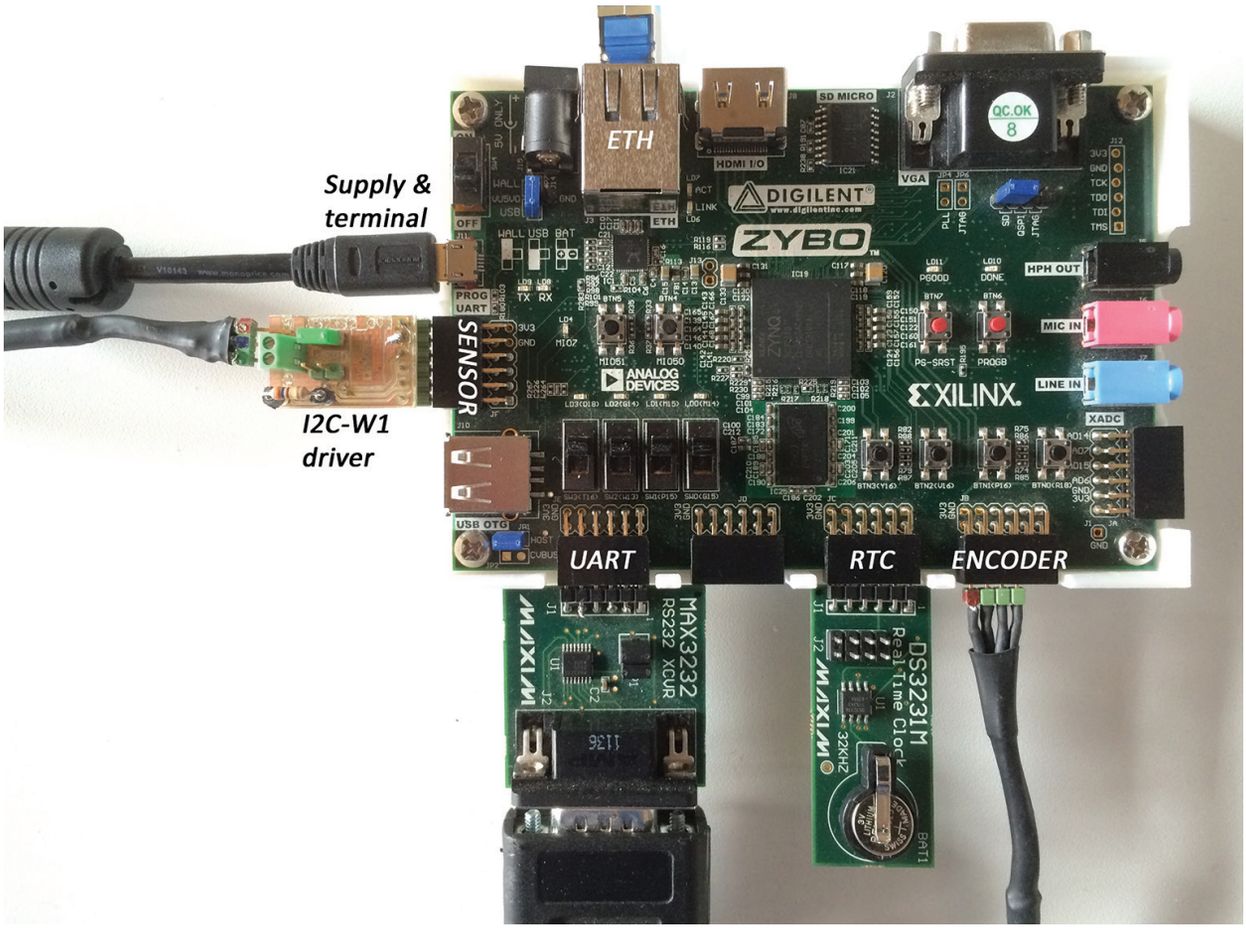

Figure 22. ZYBO board with all the peripherals connected.

flexibility and performance. The processor communicates with the processing system (PS) and programmable logic (PL) peripherals through AXI-4 bus protocol that also enables the data exchange between the external modules across the ports of the Zynq architecture. The other peripherals that compose the hardware are listed as follows:

- DDR3 volatile memory: it loads the operating system and processes.
- SPI-FLASH nonvolatile memory: it is employed to storage the FPGA hardware configuration.
- Ethernet port: it carries out TCP/IP network communications in order to access the services through the implemented web server.
- MicroSD card: this slot stores the embedded Linux kernel and its filesystem together with a copy of the hardware configuration and the bootloader.
- UART-USB port: this peripheral makes possible to communicate with the system console and perform maintenance tasks.
- UART port: this serial communication sends the commands to the servomotor in order to control it.

http://dx.doi.org/10.5772/66579

- I2C port: with this peripheral is managed both temperature sensor driver and the RTC device.
- GPIO port: it detects the sensor limit and communicates with the temperature sensor.
- A custom core implemented in VHDL: reads the signals from the encoder, identifies the direction of rotation, and stores the information in a BRAM register. What is more, the core communicates with the CPU of the system and sends a signal when the distance established in the main program is reached. The core runs in parallel with the ARM processors in order to guarantee a continuous detection of the pulses and avoid loss of information in faster acquisition processes

The software that was employed to carry out the hardware development is the Vivado Design Suite 4.4 of Xilinx. It is an interesting software suite that enables fast and efficient programming of the SoCs and also can automatically generate configurable cores and several interface circuits by using the block design tool.

The previous hardware and interconnections description can be checked in the block scheme which is shown in **Figure 23**.

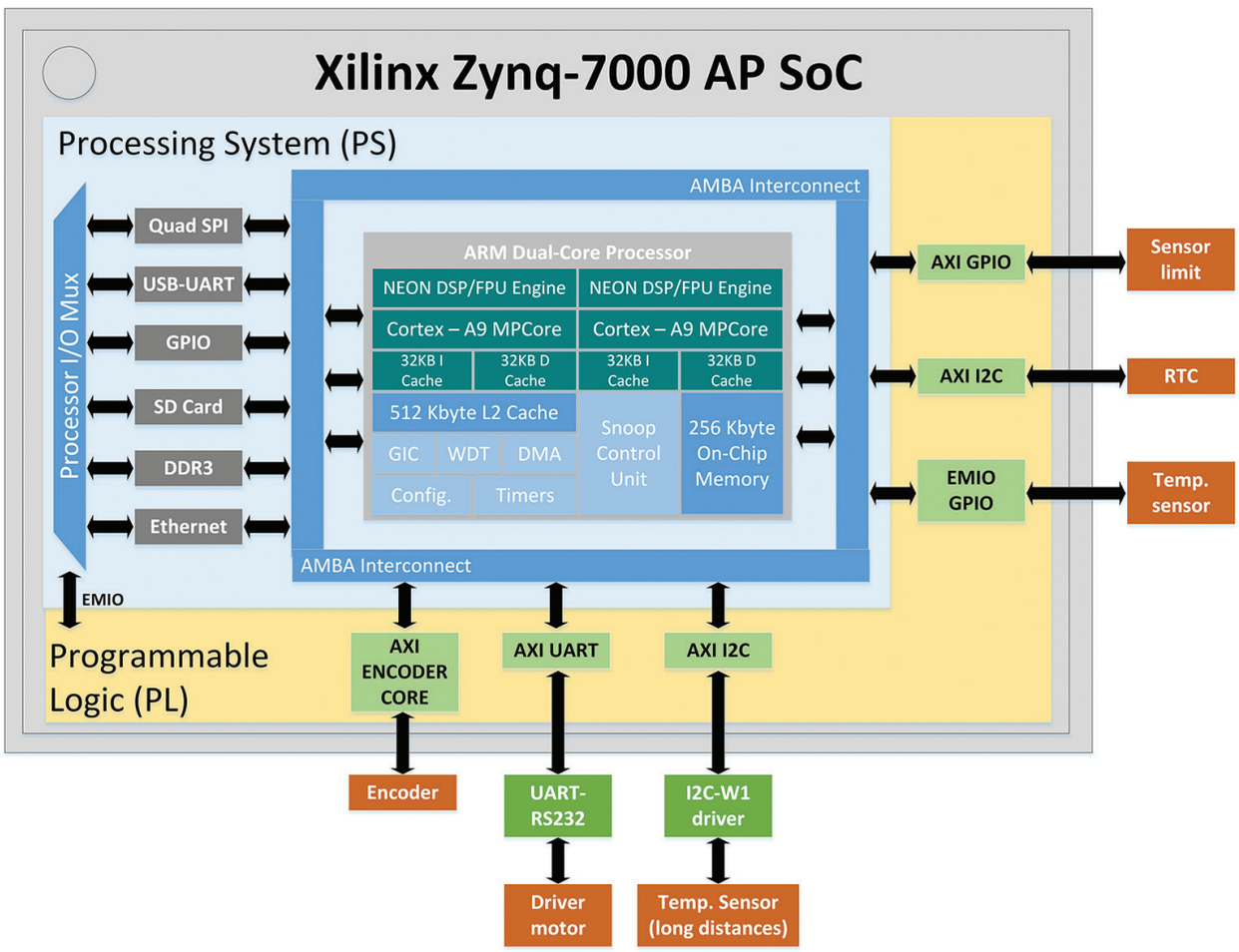

Figure 23. Implemented architecture in Zynq-7000 AP SoC chip.

3.4. Embedded operating system

The source code of an embedded Linux operating system was configured and compiled to run over the previously described hardware. This operating system is based on a modified version of the standard Linux kernel without a memory management unit (MMU), which is optimized for running in embedded systems, but keeping Linux robustness. The main advantage of this system is its well-known free open-software characteristic. However, it has another important advantage, such as all the available resources can be managed by the operating system, while the programmer is abstracted from that layer of hardware. Furthermore, there are a lot of libraries and utilities that can be integrated in Linux make easier the upgrade or development of new applications.

The operating system was implemented using the Linux version provided by the tools of uClinux distribution, which is supported by the ARM processor architecture integrated in the selected SoCs. Specifically, a stable version, in this case the version 3.14, was configured and compiled for the hardware platform in order to create the kernel image (uImage). By using Vivado Suite tool, a first stage bootloader (FSBL) for Zynq was built to set the FPGA with the previously defined hardware, load the operating system image in the DDR3 memory, and begin executing it (BOOT.bin). As the root file system, a prebuilt ramdisk image provided by Xilinx was mounted in a microSD card and wrapped in one part of the FSBL header to boot with it. Moreover, a device tree blob (DTR) project application was established by using the announced tools in order to describe the hardware in a data structure file that the kernel can understand. Thereby, the details do not need to be hard coded in the operating system, making it portable. Once the bootloader, the kernel image and the device tree were configured and generated, they were copied into the microSD card root partition, so the operating system can boot and access the mounted file system.

At last, the database, webserver, and main application processes are executed over the operating system. Looking for a powerful-reliable webserver and database, as well to improve the interaction between them, Apache 2 (webserver), MySQL (database), and PHP (server-side scripting language) were cross compiled for the ARM processor with the toolchain arm-xilinx-linux-gnueabi provided by Vivado. These processes are an open source but are not available in this kernel distribution, so a hard work was carried out to compile and run properly. Nevertheless, the benefits of incorporating them compensate with the later cost-effective development, reliability, features, and performance. In a similar manner, in order to manage the compiled and installed processes, the software of Vivado Suite was employed to program and cross compiled the executable applications. The relation between the most important parts and files is shown in **Figure 24**.

The executable applications are based on a state machine that manages the system flow that interacts with the peripherals and the database.

3.5. Webserver and database application

The main application is known as LAMP (Linux-Apache-MySQL-PHP), a combination of Linux as the operating system, Apache as the webserver, MySQL as the relational database management

http://dx.doi.org/10.5772/66579

system, and PHP as the object-oriented scripting language. The advantages of the LAMP stack technology compared with other systems over closed system platforms are as follows:

- Open sourced.
- Highly secure.
- Highly flexibility.
- Scalability.
- Interoperability.

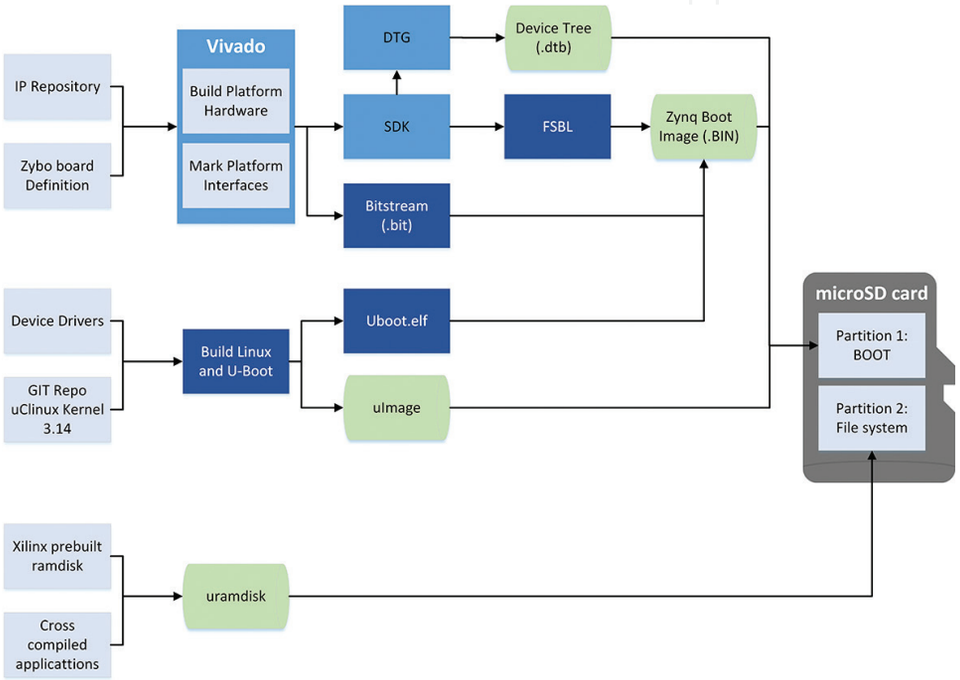

Figure 24. Diagram of files to configure and boot the system.

These characteristics help the designer to develop new applications because it is relatively easy and there is plenty of documentation available. Therefore, the LAMP stack combination makes possible to create truly database-driven and dynamic website that is easy to update and provides a lot of resources to support users. The embedded operating system layer architecture used in this application is shown in **Figure 25**.

Likewise, Apache is the dominant and most important webserver in the word due to its reliability and performance, so that combining it with a powerful database as MySQL and the possibility of using server-side scripting offer very cost-effective and versatile applications.

Commonly, SQlite [7] is implemented as database management in embedded systems because its resource optimization. Nevertheless, in this study due to the powerful processor of Zynq chip, a MySQL database was cross compiled to run in this specific architecture. MySQL offers the following advantages regarding to SQlite:

- Permits multiple queries and modifications at the same time.
- More data types.
- Better compatibility.
- Each table is in a different file (as views and objects).
- Reduces the latency in the queries.
- It is possible manage users with different levels of access.

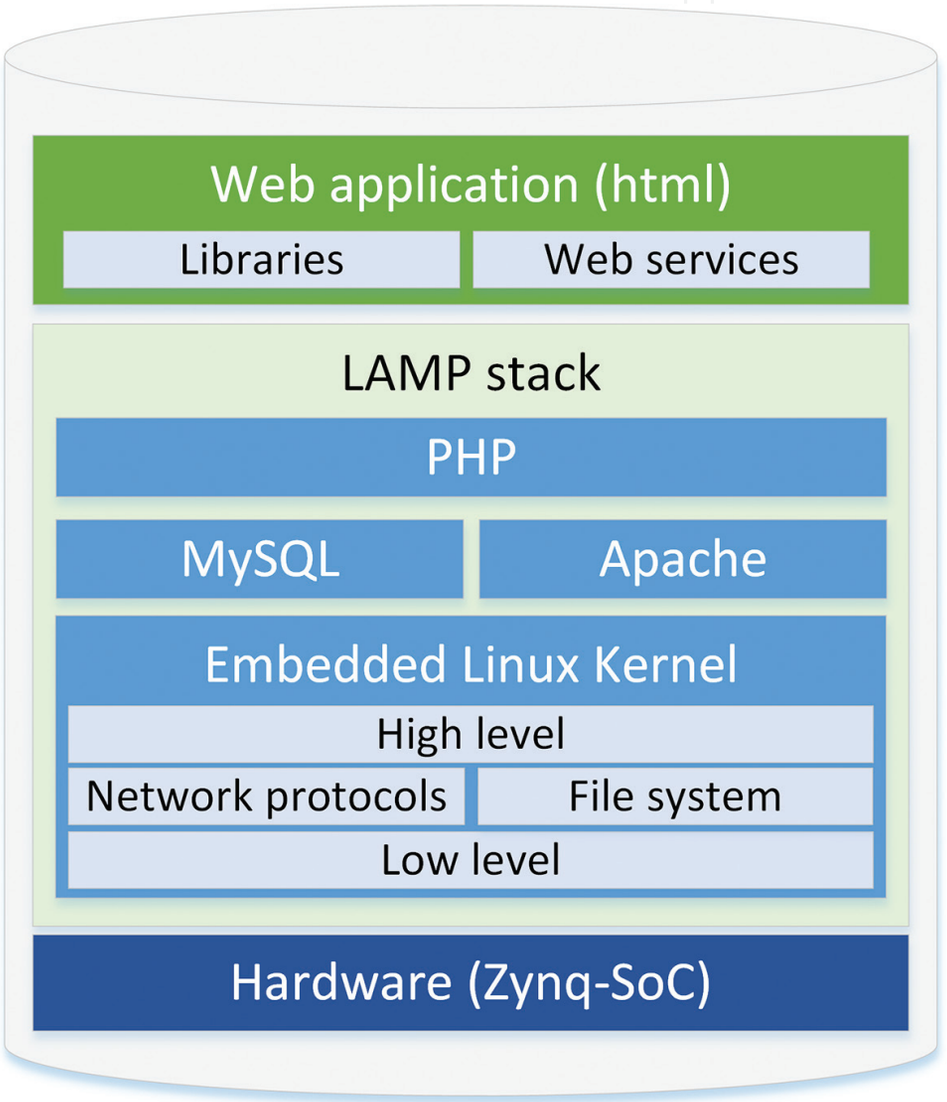

Figure 25. Embedded operating system layer architecture.

The higher software layers of the web application are divided into the following categories: HTML (HyperText Markup Language), JS (JavaScript), CSS (Cascading Style Sheets), and PHP files. This allows the user to control, configure, and monitor the performance of the

http://dx.doi.org/10.5772/66579

Geowire device through surfing the pages of the programmed web interface. Hence, the web interface is structured in three sections:

- Settings: It allows the user to set up motor driver parameters and visualize motor performance indicators and alerts.
- Test: It is possible configure database tables in order to begin a temperature acquisition process at preestablished sets of depth or load previously implemented acquisitions profiles. These parameters are storage in the database together with a timestamp during the acquisition process.
- Graph and charts: In this section, the user can visualize the representation of previously measured depth-dependent temperature profiles, visualize real-time acquisition processes, or download a Microsoft Excel spreadsheet with the recorded temperature profiles.

The design of the web application interface is shown in **Figure 26**.

Basically, when the client requests the application files to the web server in order to render the web interface, the PHP files are first processed by the server. Then, the output of those PHP scripts is transmitted to the client so dynamic content can be added. In this manner, when the client makes a PHP request that script is processed in the server which can manage the database or the system applications.

Thereby, it is possible to execute the server side applications that define the system operation flow from the client interface and manage the device peripherals. The implemented LAMP stack client-server communication diagram is shown in **Figure 27**.

Moreover, initially a validation is required in order to protect the web content from no authorized accesses. In this manner, even if the system is connected to Internet, the data content will not be accessible to the public if they are not accredited.

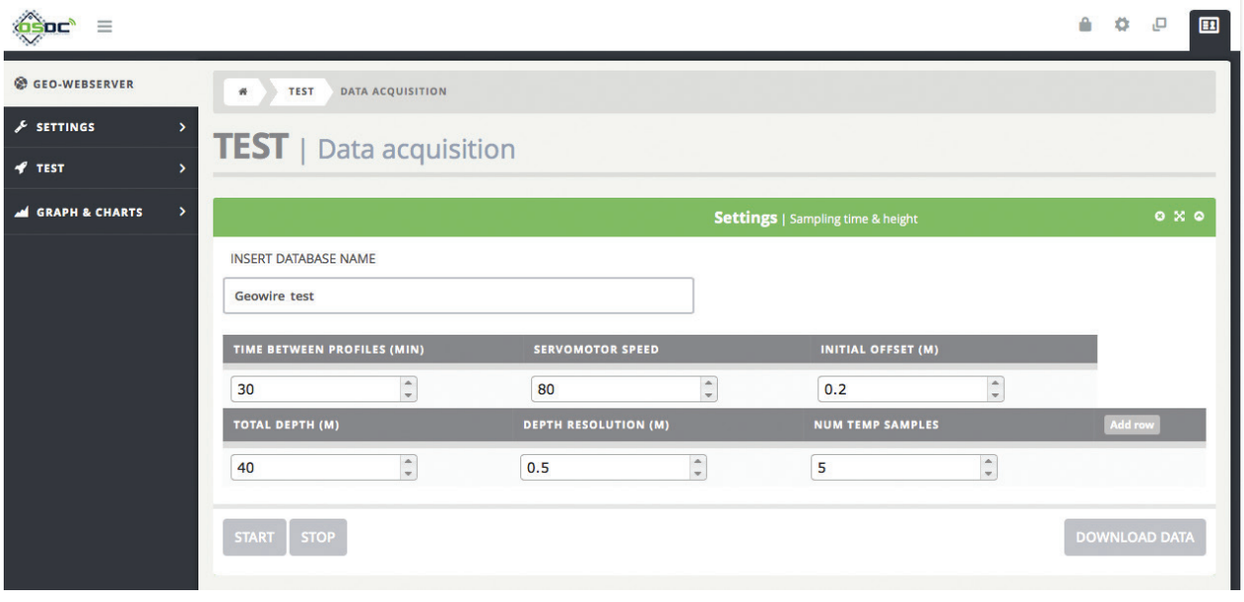

Figure 26. Web application interface for the acquisition process.

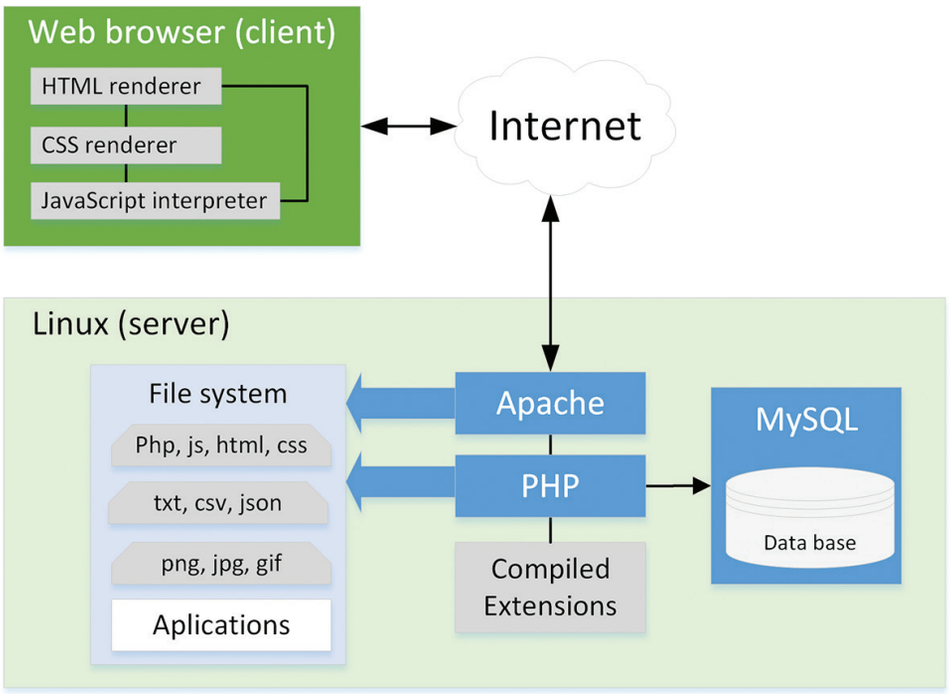

Figure 27. Implemented LAMP stack client-server communication diagram.

4. Results

4.1. System implementation and laboratory results

Figure 28 presents the obtained results after the hardware postimplementation using the Vivado tool. There are no more resources available for the mixed-mode clock manager (MMCM) and the resources for the input/output are 78%, but still there are plenty of resources for use more memory and look up table (LUT). One of the advantages of a system based on the AP SoC architecture is that it permits to implement the operating system in the ARM processor, instead of using the resources of the FPGA to build a software microcontroller. Thus, in this design, the free resources available in the programmable logic of Zynq chip would make possible incorporate more hardware specialized functions or configure peripherals with different functionalities.

In comparison with other embedded systems based on ASICs, the AP SoCs give the same function by porting the uClinux system except the performance of the processors. A design using an ASIC could incorporate faster clock rates and more powerful core processors that consume less power than the implemented system. However, the programmable logic in the AP SoC offers more flexibility. The reconfigurable IP cores make easier to reduce the design cycle which can be a key factor for the consumer electronic industry. The peripheral can be easily added or modify to the hardware with less effort according to the necessities of the customers. Additionally, the designer can implement multiprocessor by using the dual core ARM and the logic in the FPGA, which will improve the performance of the embedded system.

http://dx.doi.org/10.5772/66579

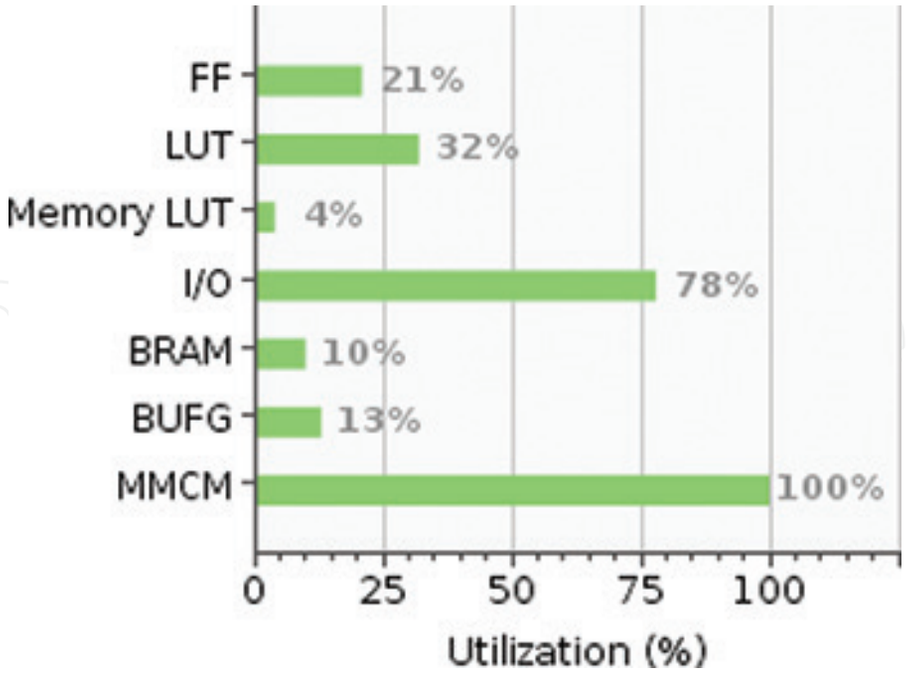

Figure 28. Resources utilization after the hardware postimplementation.

The webserver robustness was test by a concurrent access of different users to the web application. A number of 20 users from different devices were able to access the different sections of the interface without affecting the system performance. Neither the speed nor the efficiency was affected in comparison with the access of a single user.

The digital temperature sensor was calibrated using a thermal bath and accurate thermometer. The calibration was implemented using 5 points, from 0°C and by increasing 5 until 25°C, normally the temperatures during a TRT are comprehend inside that range. Then, a linear trend line was calculated to fit the data from the digital sensor with the recorded temperatures through the accurate sensor. The calibration equation is applied to the obtained data from the sensor before saving the values in the database.

The wire connected to the sensor passes through three rollers that rotate when the wire is released to lower the sensor inside the geothermal pipes. In one of these rollers, six magnets were attached along the diameter and separated the same distance between them. On the external part of the device enclosing a hall effect sensor that detects the polarity changes of the magnets. This mechanism is the encoder that measures the angular movement of the roller. Because the roller diameter, the separation between the magnets and the wire diameter is known, the system program is able to calculate the distance traveled by the sensor. The maximum spatial resolution of 1 cm was achieved after testing and calibrating the instrument by *in situ* measurements. In order to ensure the repeatability of the device for longer distance increments more trials were conducted. For a distance increment of 0.5 m, a deviation of ±1 cm was observed after 20 trials. Besides, a maximum deviation of ±5 cm was observed between distance intervals of 50 m after 20 trials (**Table 1**).

Distance interval	Deviation
0.5 m	±1 cm
50 m	±5 cm

Table 1. Geowire distance interval measurement deviation.

4.2. Experimental results throughout a distributed thermal response test in a borehole heat exchanger

In this section, an experimental test of the implemented system in one of the four BHE installed on the campus of the University of Liege (Liege, Belgium) is presented, with the aim of testing its performance and analyzing the obtained results.

The installation is equipped with double-U geothermal pipes of 100 m long, over a surface area of 32 m^2. Deposits of sand and gravel characterize the site geology until a depth of approximately 8 m. Then, the bedrock follows until the end of the borehole, which consists mainly of siltstone and shale inter-bedded with sandstone, while fractured zones are detected in the rock mass mostly until a depth of 35 m. In this installation, thermal behavior of fractured bedrock stratigraphy was being investigated throughout TRTs and distribution temperature sensing (DTS) technique. Hence, during the insertion of the geothermal pipes, fiber optics thermometers were tapped every 50 cm in direct contact with the outside part of the pipe wall. Given the relatively small borehole diameter (136 mm), spacers were not used during the installation and the distance between the U-legs is in the order of 3 cm. Among the four available boreholes, the test was conducted in B2, which was backfilled with a bentonite-based commercial material (Füllbinder, $\lambda = 0$ .95 W/mK).

Fiber optics makes possible to obtain continuous, high-resolution temperature profiles along the pipes length by applying the DTS technique [8]. The temperature resolution of the fiber optic measurements presented in this study (standard deviation) was in the order of 0.05°C. Temperature was recorded every 20 cm (sampling interval) with a spatial resolution of 2 m.

On 15 December 2015, an enhanced TRT of a heat injection of 2 kW and a duration of approximately 7 days was conducted in one of the single U-pipes of B2 which are disposed in a parallel configuration. The fiber optic cables were installed along the single U-pipe in which the heat was injected while the Geowire was inserted in the nonheated single U-pipe (observer pipe), which was filled with water. **Figure 29** presents a cross section of the borehole with the U-pipes, the location of the temperature measurement instruments where the heat was injected.

From the user interface, the Geowire temperature acquisition sequence was settled to lower the sensor until a depth of 40 m. Then, the sensor was established to stop every 0.5 m for a period of 5 s in order to achieve a thermal stabilization and avoid the possible convective

http://dx.doi.org/10.5772/66579

effects, produced by the moving water when lowering the sensor. After 5 s, the device was programmed to record three samples of temperature, one every second and storage the average value in the specified database.

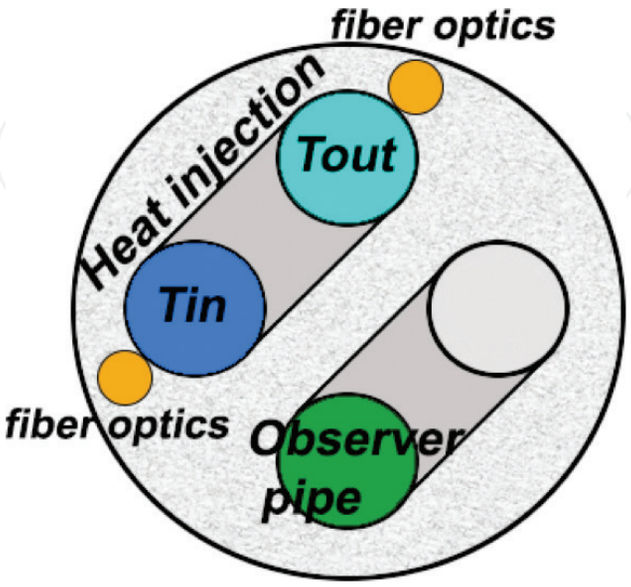

Figure 29. Cross-section of the BHE utilized in the experiment.

Figure 30 shows the Geowire adapted in a structure and the different components of the implemented system before the beginning of the test in the field.

The section of Graph & Charts of the web application allows the users to visualize the recorded data remotely during a real test. Once the device is running it is possible to observe real-time charts or load previously stored temperature profiles from the database. **Figure 31** shows the user interface for a temperature profile obtained during the announced test.

Figure 32 presents the recorded temperature profile with the proposed system inside the observer pipe, and the registered fiber optic temperature measurements (along pipe inlet and pipe outlet outer surfaces) during the heating phase of the TRT. The oscillations observed in the recorded datasets along the observed pipe may be attributed to the varying distance through depth between the observer pipe and the U-heated pipe, as well as to the ground heterogeneity. Moreover, in the first ~18 m, temperature is also affected by the air temperature as previously studied by Radioti et al. [9].

The main advantage of the recorded temperature profiles is that, by applying the proposed procedure of Aranzabal et al. [10], it can contribute to calculate a detailed depth-dependent thermal conductivity profile of the BHE subsoil surrounding layers. Basically, it consists in an iterative simulation process of a numerical model in order to fit simulation results with experimental data.

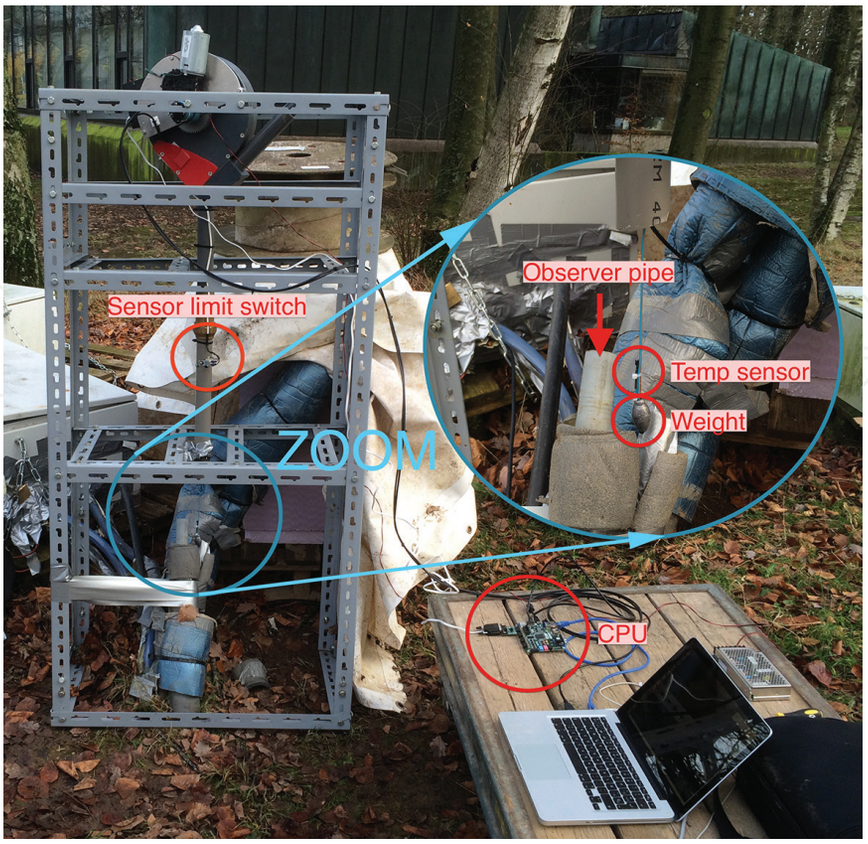

Figure 30. The implemented system before the beginning of the acquisition process in a BHE.

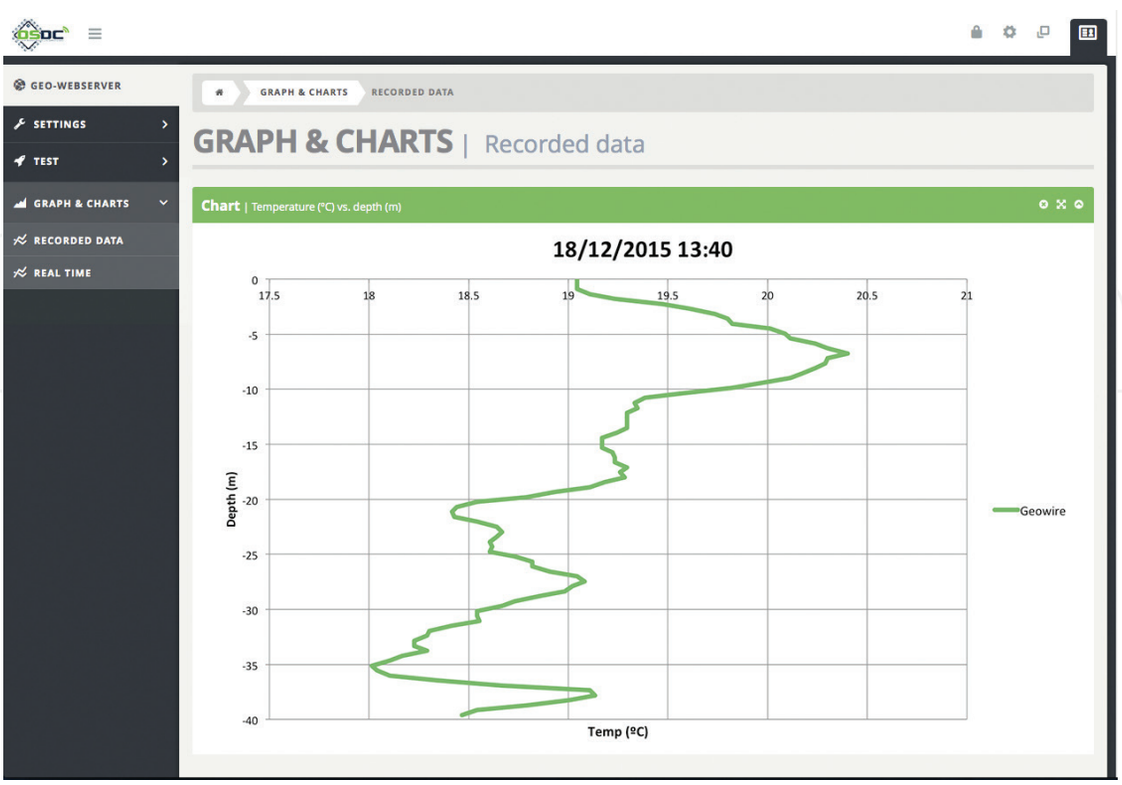

Figure 31. Recorded temperature profile along the depth through the user interface.

http://dx.doi.org/10.5772/66579

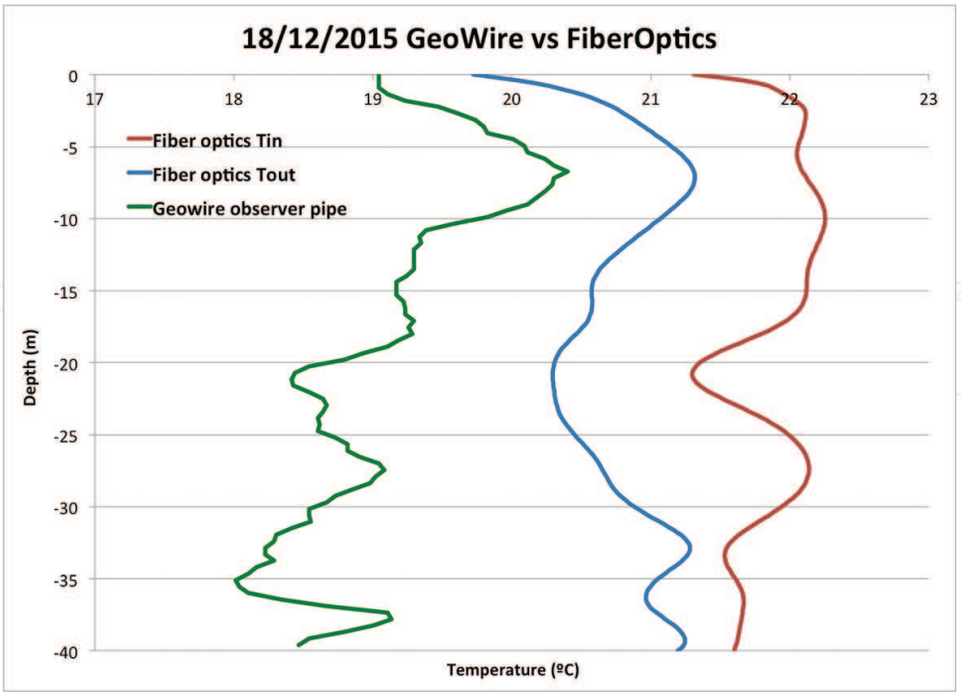

Figure 32. Recorded vertical temperature profile by the Geowire and the fiber optics for 18 of December.

5. Conclusions

The embedded systems have increase its popularity given the society evolution towards a more technology-based and automated necessities. There is no doubt about how fast this market is progressing, one reason is because more resources are being invested and second due to the constant increase in the integration capacity of the components inside the chips. For that reason, many designers need to update continuously their know-how about the methods to design SoC-based applications, which combine processing systems and programmable logic, in order to be competitive.

This chapter has been covered the evolution timeline of FPGA-based systems from its beginning until the final AP SoC chips. They are complex devices and it is necessary to have a deep understanding to utilize them in more efficient way. In this manner from all the advance digital systems, the SoCs are analyzed in more detail because they provide a solution that combines the different digital and analogic elements embedded only in one chip. That permits to develop more versatile applications but it is necessary a well-known comprehension of the analysis and design methodologies.

The introduced codesign hardware/software methodologies are very useful technique to develop time-effective applications. A good coordination between the developers of the same team is required, being the tools that can help to simulate software and hardware designs fundamental to achieve competitive results.

This study we presented the evident improvements in the SoC systems, which can run an embedded Linux for interfacing FPGA-based designs. Thus, the development of

a real application has been described and the advantages over other system have been highlighted.

Author details

Nordin Aranzabal*, Adrián Suárez, José Torres, Raimundo García-Olcina, Julio Martos, Jesús Soret, Abraham Menéndez and Pedro A. Martínez

*Address all correspondence to: nordin.aranzabal@uv.es

Department of Electronic Engineering, University of Valencia, Burjassot, Valencia, Spain

References

[1] Stephen M. Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology. Proceedings of the IEEE Vol. 103, No. 3; 2015. DOI: 10.1109/JPROC.2015.2392104

[2] Sang Don K. and Seung Eun L. An ARM Cortex-M0 Based FPGA Platform in Teaching Computer Architecture. International Journal of Computer and Information Technology Vol. 04 – Issue 06; 2015

[3] Schaumont P. A. Practical Introduction to Hardware/Software Codesign. In: Springer; 2013; ISBN: 978-1-4614-3737-6

[4] Tyson S. H, James O. H. Using an FPGA Processor Core and Embedded Linux for Senior Design Projects. IEEE International Conference on Microelectronic Systems Education; 2007

[5] Szabolcs H and Sándor-Tihamér B. Implementation of Embedded Linux Systems on FPGA Based Circuits for Real Time Process Control. International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics; 2015

[6] Young-Taek K, Taehun K, Youngduk K, Chulho S, Eui-Young C, Kyu-Myung C, Jeong-Taek K and Soo-Kwan E. Fast and Accurate Transaction Level Modeling of an Extended AMBA2.0 Bus Architecture. Proceedings of the conference on Design, Automation and Test in Europe Vol. 3; 2005

[7] Kun Y, Linying J, Liu Y and Heming P. Research of Embedded Database SQLite. Application in Intelligent Remote. International Forum on Information Technology and Applications; 2010

[8] Hermans T, Nguyen F, Robert T. and Revil A. Geophysical methods for monitoring temperature changes in shallow low enthalpy geothermal systems. Energies; 2014. DOI: 7 (8): 5083–5118

[9] Radioti G, Delvoie S, Radu J. P, Nguyen F, and Charlier R. Fractured bedrock investigation by using high-resolution borehole images and the Distributed Temperature Sensing technique. In: Canada. ISRM Congress 2015 Proceedings - Int'l Symposium on Rock Mechanics; 2015

[10] Aranzabal N, Martos J, Montero A, Monreal L, Soret J, Torres J, García-Olcina R.. Extraction of thermal characteristics of surrounding geological layers of a geothermal heat exchanger by 3D numerical simulations. In: Applied Thermal Engineering; 2016. DOI: 99: 92–102

IntechOpen