EFFICIENT ENERGY CONSUMPTION IN THE DEDICATED EMBEDDED SYSTEMS

ROTAR DAN[1*], CULEA GEORGE[1], ANDRIOAIA DRAGOȘ[1]

[1] *Vasile Alecsandri University of Bacău, Calea Mărăşeşti 156, Bacău, 600115, Romania*

Abstract: Reducing the energy consumption in all fields is currently a continuing concern. Worldwide it has formed a number of entities that seek to encourage the achievement of environmentally friendly electronic products, from several points of view. Given the continued growth of the volume of electronic systems currently used and the planned spectacular expansion in the period ahead, reducing the electricity consumption of the electronic devices and equipment's is a serious concern in the design of electronic systems.
The paper shows how an embedded system used in industrial automation and equipped with a series of energy-saving mechanisms is designed and implemented. For the physical realization of the project a programmable logic array was used because of the flexibility that it offers in the design and implementation of such systems. The mechanisms for energy saving are: power control of the various functional blocks of the embedded system control and the command frequency control of the digital circuits. With an innovative design and a proper control, the embedded functional blocks of the system are placed in the active state or in hibernation, as needed, in order to reduce the electricity consumption. Also, due to the digital character of the component modules it is possible to control the energy consumption by controlling the command frequency of these modules. In this way the modules operate at a variable command frequency, at a high value of the frequency when it increases their request and at a low frequency corresponding to the decrease of the module load.
The regime used in order to improve the energy consumption lowers the amount of energy consumed and decreases the temperature and increases the life of the system components.
Getting results is done on the one hand in the design phase and on the other hand in the operating of the embedded system. In the design phase, several variants of solving a problem are made available and the designer chooses the most convenient one. In the operational phase the power of the modules and their frequency are controlled. The obtained embedded system is designed to a specific application (dedicated system) for which it was designed, thereby obtaining a maximum desired effect.

Keywords: embedded system, reducing electrical consumption, frequency control, programmable logic array, operating mode.

1. INTRODUCTION

Currently there are strong pursuits to achieve sustainable electronic products that have a low energy consumption and to develop appropriate technologies to achieve this goal. Such green electronic products are designed primarily to protect the environment but they also offer certain economic advantages during the use [1].

Apart from this, there are many situations where the electrical power source is limited, for example, in remote and difficult to access area. In such areas it is good that the batteries' only source of power have a longer service

* Corresponding author, e-mail drotar@ub.ro

possible. In some cases, the batteries can be charged from a solar power or from wind power. Even if, in this case, the electricity supply problem is resolved, the need to reduce energy consumption remains, which increases the life of electronic devices [2].

Energy consumption in a digital circuit is represented by the static power dissipation and the dynamic power dissipation (1):

$$P = P_S + P_D \tag{1}$$

The static power dissipation that takes the value given by equation (2):

$$P_S = P_{sc} + P_b + P_l \tag{2}$$

where P_{sc} is the short circuit power, P_b is bias power and P_l is leakage power. Dynamic power dissipation is caused by the switching circuit, and is given by the relation (3):

$$P_D = C_{eff}V^2f \tag{3}$$

where P_D is the power in Watts, C_{eff} is the effective switch capacitance in Farads, V is the supply voltage in Volts, and f is the frequency of operations in Hertz.

For digital systems a series of activities leading to reduced electricity consumption are recommended: clock control, minimizing transitions, asynchronous design, and reversible logic.

This paper presents an original method of designing an embedded system with low energy consumption. Getting a low energy consumption is made in several stages. In the first phase, the design phase, assistance in obtaining the digital low power structure to match user requirements is provided [3]. This stage also determines the hardware and software components necessary for the application.

The software component is optimized depending on the structure of the predefined hardware. The hardware is also optimized and the activities controller is generated in order to keep consumption low. The following describes the experimental research methods addressed in order to achieve an embedded system with low power consumption [4][5].

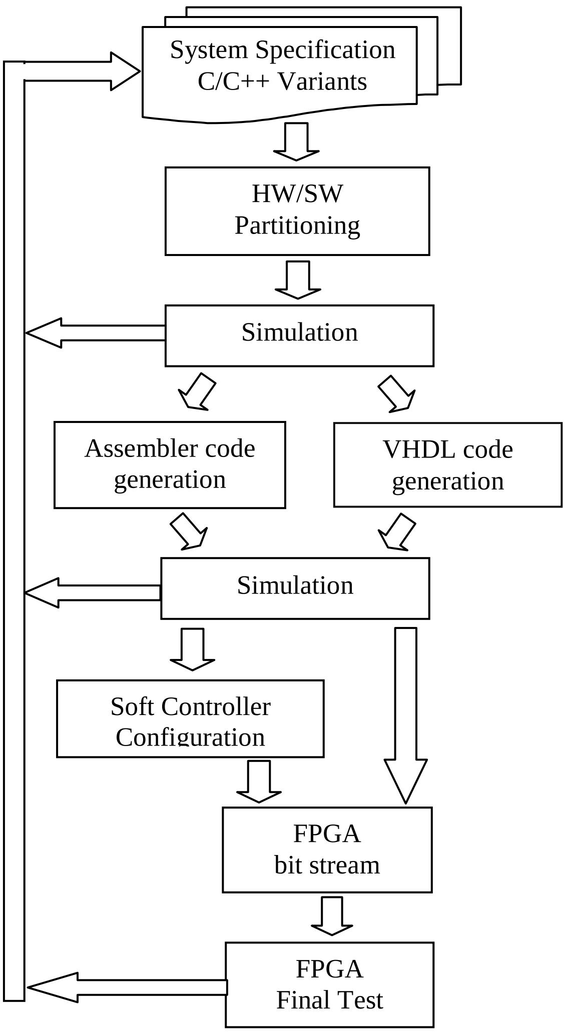

Fig1. The processing stages of the initial system description in a high level language.

2. THE STAGES OF DEVELOPMENT OF THE EMBEDDED SYSTEM

The implementation of an embedded system built using a programmable logic array requires the configuration bitstream file for the matrix. For this purpose, several steps are necessary. These steps are described below. The designer is assisted by a development environment developed for this purpose which helps the energy optimization consumption that increases the system reliability [6].

2.1. Establishing the system components

The embedded system is performed on a programmable logic array because of the advantages offered by these devices to achieve particular embedded systems. Embedded system implementation on programmable logic matrix is based on a predefined set of Intellectual Property (IP) core. The central element is the software processor with a reconfigurable instruction set computer according to the application [7][8].

In the first stage the designer makes a description of what is intended to be achieved using a high level language (SystemC). This description has more automated processing to separate the software from the hardware side. The processing stages of the initial system description in a high level language are shown in Figure 1. The program originally written in SystemC is verified through simulation to determine the correspondence between the description given and what the system wants to do [9].

Once the simulation corresponds to the requirements of the embedded system design, then proceed to determine the hardware and the software components. The determination of these components is based on the processor model used in this synthesis software. The software processor is a 16 or 32-bit processor with the instruction set configurable in the design phase.

The type of processor and the instruction set is determined in the separation phase of the hardware and the software. Follow a new simulation to verify the solution. If satisfactory results are obtained after the verification, then follows the hardware description language (VHDL) generation, with the purpose to synthesize the hardware component in the programmable logic array and the assembly language program generation for the software component.

At this stage, the optimizations performed refer to the ratio between the hardware and the software, to choosing the optimal logical structure, to establish the set of instructions for the software processor and to create the program for the software processor.

2.2. Hardware optimization

The hardware optimization is made in terms of its electricity consumption during operation. Digital circuitry is characterized by a static power consumption and by dynamic power consumption. The static power consumption depends on many parameters and may be less influenced by the designer [12].

The dynamic power consumption depends on the switching frequency and the connected load. In this paper, the energy consumption is optimized by controlling the connected modules supply and by the clock frequency control. The static power consumption is reduced by suspending the operation of the respective module and by entering hibernation during downtime. The dynamic energy consumption is determined by changing the frequency control by a central command, depending on the importance and urgency of the task fulfilled. For example, the communication module is powered at a higher frequency during communications and at a lower frequency between communications.

2.3. Software optimization

The software optimization refers to the application designed for the software controller of the application. As described by the designer, the hardware component is generated and also the corresponding software component of that hardware. The hardware is based on a structure with a particular software processor with the instruction set configurable as needed. The generated program based on the description of the designer is optimized in the sense that it contains as many small instructions and the instructions chosen to determine a reduced activity of the Central Unit.

Depending on the physical structure chosen, the program generated for the CPU must manage software components and the work modules. Depending on the activities foreseen in the software component, the system modules are placed in active status when used or in hibernate mode when not in use. For this activity a balance between active and hibernation state must be stricken, taking into account the energy consumed when switching between the two states [13].

4. CONCLUSIONS

Following the experiments, we have found that getting a low power consumption depends on many factors. Firstly, of special importance is the type of application implemented. In this regard, if the application makes a series of continuous activities without the breaks between these activities being large enough, then switching the modules in hibernation doesn't have enough electricity savings.

In this situation the largest contributions are brought in by the strategy of the choice of the instruction set. This reduces the number of circuits for implementing the software controller and the number of activities carried out.

The frequency control of the digital circuits is one of the most effective activities for electricity saving. In this case, it is important that the circuits are able to quickly switch from one frequency to another.

REFERENCES

[1] Yun Wei Li, Ching-Nan Kao, An Accurate Power Control Strategy for Power-Electronics-Interfaced Distributed Generation Units Operating in a Low-Voltage Multibus Microgrid, IEEE TRANSACTIONS ON POWER ELECTRONICS, VOL. 24, NO. 12, DECEMBER 2009, p. 2977-2988.
[2] ***, Minimizing power consumption in FPGA designs, XCELL 19, page 34, 1995.
[3] Vouzis P., Kothare M., Bleris L., Arnold M., A System-on-a-Chip Implementation for Embedded Real-Time Model Predictive Control, IEEE Transactions on Control Systems Technology, vol. 17, no. 5, 2009, p. 1006-1017.
[4] Priyanka Trivedi, Rajan Prasad Tripathi, Design & analysis of 16 bit RISC processor using low power pipelining, Computing, Communication & Automation (ICCCA), 2015 International Conference on, ISBN 978-1-4799-8889-1, 2015, p. 1294 - 1297.
[5] Piguet, C, et al., Low-power embedded microprocessor design, Proceeding Euromicro22, September 1996, pp. 600-605.
[6] Koegst, M, et al., Low power design of FSMs by state assignment and disabling self-loops, Proceedings Euromicro 97, September 1997, p 323-330.
[7] Kailasam, Umadevi, High level VHDL modeling of a low-power ASIC for a tour guide. Graduate Theses and Dissertations, 2004.
[8] Mike Salas, Low Power Design Basics. How to Choose the Optimal Low Power MCU for Your Embedded System, Silicon Laboratories Inc., 2015
[9] Benini L., Bertozzi D., Bruni D., Drago N., SystemC cosimulation and emulation of multiprocessor SoC designs, Computer (Volume:36 , Issue: 4), ISSN 0018-9162, 2003, p. 53 – 59.
[10] Stuart Swan, An Introduction to System Level Modeling in SystemC 2.0, Cadence Design Systems, Inc. May 2001
[11] Ilbeyi Berkin; Batten Christopher, JIT-assisted fast-forward embedding and instrumentation to enable fast, accurate, and agile simulation, Performance Analysis of Systems and Software (ISPASS), 2016 IEEE International Symposium on, p. 284 – 295.
[12] Yankin Tanurhan, Processors and FPGAs Quo Vadis?, Computer (Volume:39 , Issue: 11), ISSN 0018-9162, 2006, p. 108 – 110.
[13] Raabe Andreas, Felke Armin, A SYSTEMC language extension for high-level reconfiguration modelling, Specification, Verification and Design Languages, 2008. FDL 2008. Forum on, p. 55 – 60.
[14] Jagadeesh, George Rosario, Lam, Siew Kei, Srikanthan, Thambipillai, A Short Course on Implementing FPGA Based Digital Systems, Parallel and Distributed Systems, 2008. ICPADS '08. 14th IEEE International Conference on, p. 741 – 744.
[15] Heule Stefan, Schkufza Eric, Sharma Rahul, Aiken Alex, Stratified synthesis: automatically learning the x86-64 instruction set, PLDI '16 Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation, ACM New York, NY, USA 2016, ISBN: 978-1-4503-4261-2, p. 237-250.
[16] Vyas S., Chetan N., Zambreno J., Gill C., Cytron R., Jones P., An FPGA-Based Plant-on-Chip Platform for Cyber-Physical System Analysis, IEEE Embedded Systems Letters, vol. 6, no. , 2014, pp. 4-7.