

# DESIGNING THE PROCESSOR INSTRUCTION SET ON A PROGRAMMABLE LOGIC ARRAY

ROTAR DAN<sup>1</sup>, CULEA GEORGE<sup>1</sup>, ANDRIOAIA DRAGOS<sup>1</sup>

<sup>1</sup> Vasile Alecsandri University of Bacău, Calea Mărășești 157, Bacău, 600115, Romania

**Abstract:** The paper presents the authors' original contributions to the synthesis of embedded systems based on programmable logic arrays. An embedded system has one or more central units with a program structure. This allows the optimized design of the instruction set for that central unit.

The paper presents the method used to design central units with a dedicated set of instructions. The advantages and disadvantages of the method are discussed and the design steps are presented. We discuss the shortcomings regarding the portability of the programs and we also show the methods of solving this. In the paper there is also an analysis of the structure used for the dedicated set of instructions.

**Keywords:** programmable logic array, instruction set, software central unit, embedded systems, configurable and adaptable system.

## 1. INTRODUCTION

The central unit instruction set is one of the main components that determines its performance [1] [2]. The central unit instruction set is established at the central unit design and cannot be changed by the user.

The facilities offered by the programmable logic arrays, namely the determination of a flexible physical structure, the possibility of adapting the physical structure to the particular requirements, the possibility of designing the set of instructions for the respective central unit and the flexibility of the offered solutions allow the optimization of the set of instructions and the customization of its usage depending on the application realized.

Thanks to this, it is possible to optimize the flow of the design activities of an embedded system and to increase its performance by creating new features compared to classical methods [3]. Thus, the instruction set of the soft CPU can be adapted to the specificities of a particular application. This can reduce the central unit's programming effort and shorten the execution time of the application by choosing the best set of instructions. Each instruction in this set must strike a balance between the complexity of the instruction (to simplify the application programming effort) and its efficiency (to increase the execution speed of the application).

In order to achieve this goal, one must also establish the instruction set in the design phase of the embedded central unit structure because its physical structure is influenced by this set of instructions.

To determine the set of instructions, an analysis of the application (s) for which the central unit will be used is to be made. This analysis selects in the space of the set of possible instructions, the set of instructions appropriate to that situation. These instructions will be the set of instructions for the central unit that will be used to determine its structure.

## 2. DETERMINING THE SET OF INSTRUCTIONS

The instruction set of a central unit must have several characteristics to determine a functional structure of the central unit. Some of the important features will be discussed below [4] [5].

#### Compatibility with hardware deployment.

The instructions chosen for the set of instructions must allow simple implementation on the chosen hardware structure (programmable logic matrix) with low resource consumption. This helps to achieve the optimal working speed and to save the resources of the physical structure used.

#### The degree of parallelism.

This feature has a double meaning: the feature refers to the ability to perform multiple activities in parallel by the central unit, but the feature also refers to the possibility of using in parallel several physical structures implemented.

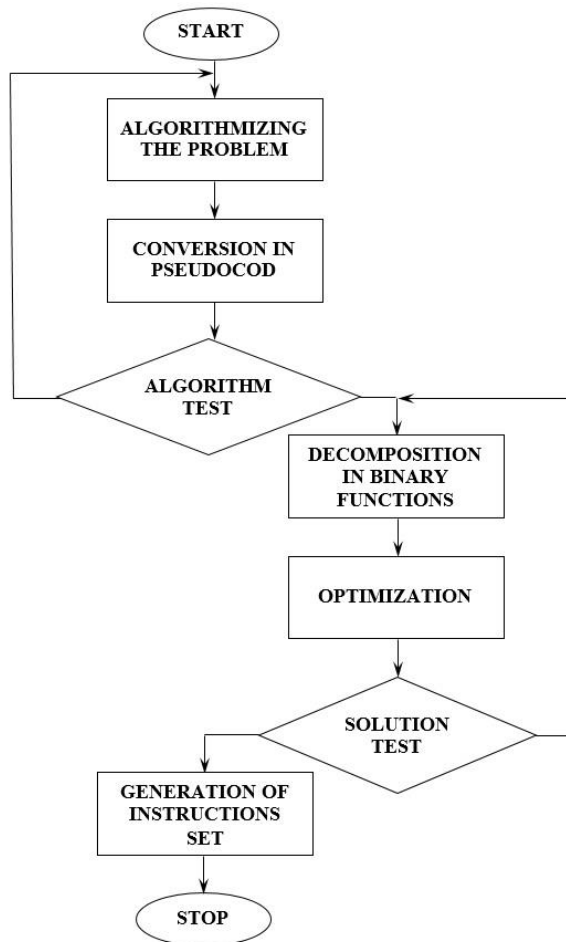


Fig. 1. The method of generating the instruction set.

#### Turing Complete.

The set of instructions must allow for a program to solve any problem. In particular, solving a certain class of problems needs to be done in the shortest possible time.

#### Operators and their addressing mode.

The effectiveness of the instruction set is essentially the number and type of operands they use, and their addressing modes. In this implementation, for simplicity, fixed-length operands are used with addressing modes: immediate, indirect by register, paging and indexing.

In order to determine the appropriate set of instructions, the set to be used for the implementation of the central unit, several steps can be performed that can be retaken if necessary (Figure 1).

In the first stage, the problem to be solved is described in pseudocode. This identifies the number and type of input variables and targets to be achieved. For this situation, the set of functions necessary to solve the proposed problem is generated.

The function system thus obtained is processed in the sense of its decomposition into elementary functions characteristic of the structure (programmable logic matrix) chosen and its minimization by detecting common parts and the elements that can be eliminated. These elementary functions are the basis for generating the set of optimized instructions for the chosen issue.

Once the set of instructions is set, move to the central unit's synthesis. For synthesis, a finite data flow machine is used. At this stage, additional physical optimization can be applied based on the set of instructions used.

As has been shown in this phase of the project, I use fixed-dimensional instructions with a linear coding method. This simplification is applied at this stage for testing the chosen solution and the possibility of checking the compliance.

### 3. THE MEMORY SYSTEM

The memory system is based on ROM and RAM that can be synthesized on a programmable logic array. ROM is intended for firmware and has no role in the synthesis of the instruction set [6]. The structures needed to run the programs are built into RAM.

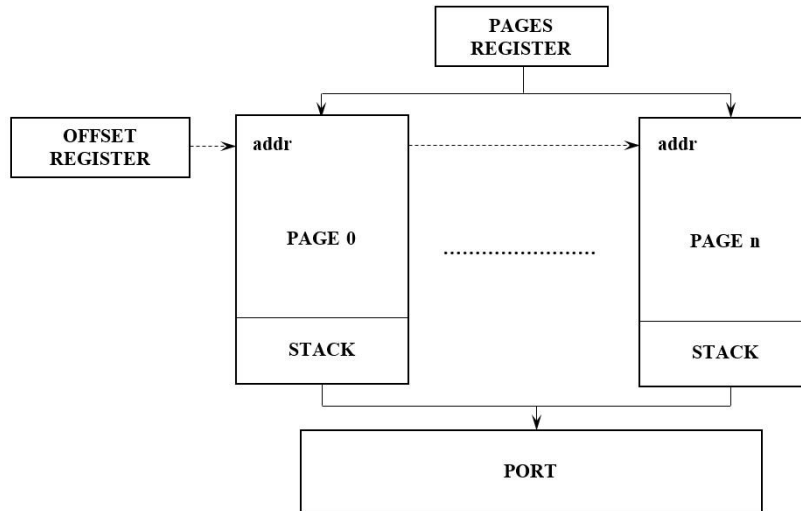


Fig. 2. The structure of working memory

The instruction handlers consist of numeric values, input / output ports, memory locations and / or registers. Memory locations along with the registers are made with RAM which gives you great flexibility in using the central unit instructions.

At this stage of the project, the size of the memory location used is fixed at 16 bits, which requires paging of the memory in pages of 64 kB. For this reason, the addresses used in the execution of the programs have two components: a page

address and an instruction address on that page. This mechanism has the advantage of modularization of programs in modules with a maximum size of 64ko, modules that can be relocated in any area of RAM work memory.

Each module contains its own stack used to execute the program sequence in that module.

Input / output ports are also regarded as memory locations, which simplifies how programs are made.

To increase the amount of software applications that can be included in the system, there is a mechanism for paging memory in 64 kb segments. Each segment is independent and can contain a complete application (the segment has its own stack) the addressing is being made relatively at the beginning of the segment.

### 4. THE INPUT / OUTPUT SYSTEM

The input / output system is a special memory area, the input / output ports being viewed as shared memory areas of any page in the system.

System ports can be used as memory locations that have reserved addresses, since these locations have a special behavior. First, these addresses have the same use on each system memory page.

On the other hand, in order for the system to adapt to slower peripheral devices, in the read / write cycle with memories representing the port, an additional waiting time is introduced.

### 5. CONCLUSIONS

The system is designed for the construction of embedded systems with one or more central units using programmable logic arrays.

The main advantages obtained by applying this method are listed below.

The instruction set optimization allows you to reduce the number of components used to synthesize the central unit. This allows to increase the number of strings made on the programmable logic matrix. Also, the reduction in the number of components increases the reliability of the system.

Another aspect is the effort to develop programs for a given application. Optimizing the set of instructions leads to the simplification of design and to the development of the applications.

One of the main issues of such an approach is the possibility of reusing program sequences. Since the basic logical structure is the same for all the set of instructions, there have been designed applications for converting programs for reuse. In this way, program libraries are built to simplify design.

## REFERENCES

- [1] L. Kalampoukas, A. Varma, D. Stiliadis and Q. Jacobson, "The CPU Design Kit: An Instructional Prototyping Platform for Teaching Processor Design," Workshop on Computer Architecture Education, Int'l Symposium in Computer Architecture, 1995.
- [2] T. Stanley and M. Wang, "An emulated computer with assembler for teaching undergraduate computer architecture," Workshop on Computer Architecture Education, Int'l Symposium in Computer Architecture, 2005.
- [3] L. Udugama and J. Geeganage, "Students' Experimental Processor: A processor integrated with different types of architectures for educational purposes," Workshop on Computer Architecture Education, Int'l Symposium in Computer Architecture, June 2006.
- [4] Krishna Melarkode. Line Associative Registers. Master's Thesis, University of Kentucky, October 2004
- [5] G. Hinton, D. Sagar, M. Upton, D. Boggs, D. Carmean, A. Kyker, and P. Roussel, "The microarchitecture of the Pentium 4 processor." Intel Technology J. Q1 2001.
- [6] C.McNairy, D.Soltis, "Itanium 2 Processor Microarchitecture", IEEE Micro Vol. 23 Issue 2, pp.44-55, March 2003.