VI Induscon - Conferência Internacional de Aplicações Industriais
Recife - PE, 09-12 de abril de 2006

METHODS AND GUIDELINES FOR EMBEDDED SYSTEM PROCESSOR SELECTION

Alexandre José da Silva[1] Rômulo Silva de Oliveira[2]

[1]WEG Acionamentos
Departamento Técnico - Engenharia de Produtos
Avenida Prefeito Waldemar Grubba, 3000
CEP: 89256-900 - Jaraguá do Sul/SC

[2] Universidade Federal de Santa Catarina - UFSC
Departamento de Automação e Sistemas - DAS
UFSC/CTC/DAS - Campus Universitário - Trindade
CEP: 88115-400 - Florianópolis/SC

Email: alexandrejs@weg.com.br; romulo@das.ufsc.br

Abstract – The goal of this paper is to survey methods and guidelines proposed for the selection of an embedded system processor. The methods and guidelines described in the literature are organized and presented in a table. The table data is analyzed and it is possible to identify the main aspects of embedded system processor evaluation, such as benchmark, cost and development tools.

***Index Terms*— Digital Signal Processor (DSP), Digital Signal Controller (DSC), embedded systems, microprocessor (MPU), microcontroller (MCU), selection methodology.**

I. INTRODUCTION

Nowadays, there is a huge variety of specific equipments which microprocessor controller is embedded in them. Such computer systems are embedded as components inside of a larger system. For the user of this larger system, the computer is not visible as such. The user only notices the consequences of its operation through the external behavior of the larger system. Such computer systems are known as embedded computer systems or just embedded systems.

Embedded systems are present in practically all human activities and, because of the low current technological costs, there is a trend to increase their presence in the consumer daily life. Examples of such systems are cellular telephones, digital cameras, control systems for cars, general industrial equipments, microwave ovens and countless other electronic equipments.

The project of such computer systems is complex because it involves knowledge not commonly considered in general computer projects. There are issues about portability and limit of consumption of energy without loss of performance, memory restrictions, timing restrictions and the short time available for design.

Most embedded systems are programmed and they include both hardware and software components. Generally, system quality is measured by specific application guidelines that include performance, development and production cost, flexibility, fault tolerance, etc, which will be reached depending on the hardware and software design. Methodology of hardware/software co-design have been developed to support in a effective way embedded systems design [12].

It is very important to have a methodology for the selection of the ideal processing element for the embedded system, in the context of hardware/software co-design. The processor selection methodology should select the processing element for a specific task so that it optimizes cost, performance and it reduces the product time-to-market.

Currently, there are many MCU (microcontroller), DSP (digital signal processor) and DSC (digital signal controller) manufacturers, each one with many families that have several different devices. To select the processing element with better cost/benefit/performance for a special application is a complex task. The task presents many variables that can influence the selection such as specific peripheral needs, required resolution and precision, energy consumption, cost, etc.

The goal of this paper is to survey methods and guidelines proposed for the selection of an embedded system processor. The methods and guidelines described in the literature are organized and presented in a table. It is also included comments on the results.

II. EMBEDDED SYSTEMS

Most current electronic system functions involve some kind of computation and control that is carried through by digital components. In accordance with [12], three basic classes of digital systems exist:

- **Systems emulation and prototyping:** they are based on reprogrammable hardware technologies. Such systems require specialists and they are used for digital systems validation;

- **General purpose computer systems**: they include traditional computers from laptops to supercomputers. Such systems are characterized by the fact that the final user can program the system. They support different applications depending on the type of software used;

- **Dedicated computer system and control**: it describes a specific application where the user can have limited access to the system programming. Such systems are called embedded systems too. This system can be dedicated to the control applications (embedded control), communication, data processing and hybrid applications that include control and data processing.

The embedded system project is complex and expensive for a company, involving multidisciplinary teams such as digital hardware, analog hardware, mechanical interfaces, software and tests, beyond expensive computational tools.

According to [14], embedded systems can be divided in three groups:

- **Embedded systems for light applications:** such systems requires performance with low cost, small programs generally using assembly language and as little extra hardware as possible. The CPUs (central processing unit) usually have 4 and 8 bits;

- **Embedded systems for common applications:** such systems require medium performance, medium cost embedded CPU that usually have 16 or 32 bits. The program size includes a significant amount of code and it is generally developed with a high level language;

- **Embedded systems with high performance:** such systems require high speed processing and real-time data processing. They are multiprocessor systems where one or more CPUs run handcrafted code.

Until recently, designers have been limited to choose between microprocessors and microcontrollers as the processing element in embedded systems [2].

Recent improvements in semiconductor technology have increased the designers choice by including other solutions for processing such as PLD (programmable logic device), subdivided in CPLD (complex programmable logic device) and FPGA (field-programmable gate array), DSP, DSC or a system-on-a-chip (SoC) implementation as ASIC (application specific integrated circuits).

Initially, embedded systems were implemented based on generic microprocessors that presented reasonable performance in control and signals processing applications. There was the necessity of a large electronic board area to shelter all necessary components for microprocessor system operation, dedicated interfaces and control components.

With the microcontroller advent, which is optimized for control implementation, and the DSP, that is optimized for signal processing, embedded systems became compact with specific hardware and enough performance for the best cost/benefit/performance.

The MCUs and DSPs encapsulate all the necessary components for control system implementation in a single IC (integrated circuit). However, the MCUs and DSPs main manufacturers offer these components in many package options, inputs and outputs pins variants, memory configurations and peripherals features.

PLDs are used with restriction due to its high cost. Generally, they are used to implement logical interfaces and circuits as a complement to microcontroller or DSP based systems.

For each application type there is a processing solution that presents the best cost/benefit/performance relation. However, microcontrollers, DSPs and recently DSCs are the most used in the embedded systems universe. They have flexibility and low cost because these components are standard off-the-shelf in most cases. They have powerful development tools that are even free sometimes. Because of these features, it is possible to develop embedded systems even more powerful, optimized and low cost.

III. PROCESSOR SELECTION METHODOLOGIES

Depending on the application, the project of an embedded system can use one or more processors as its platform. Although this solution seems conservative from the innovation point of view, it brings advantages from the operational side. First there is the scale factor. Processors are found in thousands of projects where its cost is dissolved among many customers. Once that a platform based on a processor is available inside the company, new versions of products can be made by the adaptation of the software platform. System customization is obtained through the application software which takes most of the project time.

For the designer of embedded systems, the first choice is a microcontroller, and that is the most important action to be taken. Many times the delivery of the end product within schedule, respecting budget and specifications, depends on the correct choice of the processing element.

In our research we found in the literature a single methodology that guides the designer step by step for the microcontroller selection. Most surveyed articles bring only important aspects to be observed in the selection process of a processor for embedded systems.

The surveyed guidelines and methods show many important aspects on the processor selection process. Table 1 compiles all the aspects, classifying the methods and guidelines showing the aspects that are most important. For each aspect it is attributed a note as described below:

- 0 (weak): not mentioned;

- 5 (medium): mentioned as consideration;

- 10 (strong): mentioned as important requirement.

By applying notes to the aspects it is possible to identify the method or guideline that presents the biggest number of aspects and the biggest score. This one can be considered the most complete processor selection process. The most important aspects are highlighted many times by the methods and guidelines.

The guidelines found in [1] and the method considered in [11] contemplate the biggest number of aspects (13). However, the method [11] only lists some aspects only as a consideration. It totalizes less points than the guidelines in [1], that totalizes 95 points.

By analyzing table 1, it is possible to identify the most important aspects in the processor selection process. The aspects most cited are:

- benchmarks (5 citations);
- cost (4 citations);
- hardware and software development tools (4 citations).

According to table 1, the aspect most cited in the methods and guidelines surveyed is benchmark, the technique that exposes the processor performance.

The performance evaluation process based on comparisons among processors by measurement is called benchmarking. The workload used in the measurements is called benchmark [7].

A benchmark is composed by a set of programs that tries to emulate the complete application characteristics. It generates a performance estimate.

According to [2], a true benchmark requires a careful balancing of system and variable requirements. The processor performance for a specific application can be different for others applications. It must be considered many aspects to determine what processor has better or worse performance in a benchmark test.

Embedded systems execute specific tasks, where the processor performance must be sufficient for the perfect execution. Depending on the task complexity, 4 or 8 bits processors are apt and have enough performance. The problem is the huge variety of devices offered by diverse manufacturers.

In this context, there are some benchmarkings that try to assist the designer in the selection process. However, the performance indexes showed in the process need to be analyzed with caution, observing the benchmark type used in the evaluation. The results should be considered as a reference and cannot be used as a conclusive item in the selection process, except for when all the application or some application critical routine code is used as benchmark.

The cost appears as the second most important aspect. It defines the limits about which processors can be used in determined embedded system. The objective is to select the best processor inside determined price range, or either, the processor with the best cost/benefit relation. Many times, the processor cost can make unfeasible the development of a specific embedded system.

Nowadays, manufacturers offer 16 bits processors with similar or sometimes even the same cost of the 8 bits ones. Of course, it would be better to apply a 16 bits processor with a cost of 8 bits. However, the development tools cost, both hardware and software, has to be analyzed.

The concern with the development tools, both in hardware and software, is the next most important aspect. To choose the software development tool for the processor is not a decision less critical than to select the processor. According to [5], software development cost is around 50 to 75% of the embedded system project cost.

The development tools directly affect the development time and consequently the schedule conclusion and product delivery.

Software tools can be classified as compilers, simulators, writing program software and Integrated Development Environments (IDE).

TABLE 1
Comparison of the method and guidelines surveyed

Important aspects		Method [11]	Guidelines [2]	Guidelines [13]	Guidelines [8]	Guidelines [3]	Guidelines [1]	Citations
Knowledge of system requirement and restriction	Low power	0	0	10	5	0	5	3
	Resolution	5	0	0	5	10	0	3
	Real-time	10	0	5	0	0	5	3
	Cost	0	0	5	5	5	10	4
Memory	Extended	5	0	0	0	5	0	2
	On Chip	5	0	0	0	0	0	1
	Program	10	0	0	0	0	0	1
	Data	10	0	0	0	0	0	1
Family of processors		0	0	0	0	0	10	1
MAC		0	0	0	0	10	0	1
DMA		0	0	0	0	10	0	1
Peripheries		5	0	0	0	5	10	3
Benchmark		10	10	0	5	10	10	5
Operational system		0	10	0	0	0	0	1
Processor Supplier	Technical support	5	0	0	0	0	10	2
	Previous applications	5	0	0	0	0	5	2
	Delivery	5	5	0	0	0	10	3
Development tools	Hardware	5	10	0	5	0	5	4
	Software	5	10	0	5	0	5	4
	Technical support	0	0	0	0	0	5	1
	Previous applications	0	0	0	10	0	0	1
	Upgrades	0	0	0	5	0	5	2
Number of cited aspects		**13**	**5**	**3**	**8**	**6**	**13**	
Total of points		**85**	**45**	**20**	**45**	**45**	**95**	

Integrated Development Environments (IDE) facilitate the embedded system development, because they integrate all necessary tools for the application development, such as, compiler, simulator and writing program software. There are

Integrated Development Environments that offer the same interface for different processors families from the same manufacturer, such as Mplab [9], CodeWarrior [4], HEW [10] and others. There is an Integrated Developed Environment by IAR Systems [6] that keeps the same interface for different processors from different manufacturers.

According to [5], hardware tools can be divided in two groups: invasive and not invasive tools.

The invasive tools can affect the processor functioning, causing small variations in the execution time and problems with peripherals and interruption functioning. The not invasive tools are carefully projected to replicate all processor functions, guaranteeing the functioning in any situation of the emulated system.

IV. CONCLUSION

Based on our study, it is possible to conclude that there is not a complete processor selection methodology in the literature. However, there are many guidelines with some items that must be observed in this process, such as benchmark, the cost of processor and the development tools. Table 1 shows all aspects found in the literature and classifies the methods and guidelines. It identifies common points and indicates a proposal of a processor selection methodology for embedded systems.

Any method or guideline surveyed makes explicit reference to the embedded system specification. It is clear that the processor selection heavily depends on the embedded systems specification to be developed. It is through this specification that the processor features are defined and associated with a cost objective.

A complete methodology would be one that describes the embedded system with a standard syntax for hardware and software modeling, including all the system restrictions and features. Based on this embedded system model it would be possible to determine the restrictions and features associated with the processor. Also, it must be considered external features that will directly influence the cost and the development time of them. This includes designer experience with some processor family, existence within the company of hardware and software development tools for a processor family and software reutilization.

After the embedded system specification, it must be done the hardware and software partitioning. In accordance with the embedded systems design steps defined by hardware/software co-design methods, processor selection must be done after the definition of what parts will be implemented by hardware and by software, that is, after the partitioning. The parts that will be implemented by software have to use a processor that fulfills the application requirements and that have enough performance to execute it.

A processor selection proposal would be divided in the following steps:

- embedded system modeling;
- processor restrictions and required features identification;
- external features to the development identification;
- survey and processor selection based on restrictions and required features;
- benchmark using system software model.

With the increasing requirement of applications performance, embedded systems are always more complex and demanding. A design methodology is necessary as a design tool.

This design tool would allow the modeling of the hardware and software parts and would have mechanisms for modeling any processor selected in the survey. Also it must execute the benchmark of the processors selected using the system software model as workload and classifying them according to the measured performance.

Once it is found the processor with better cost/ benefit/performance relation, the designer must analyze the offered development tools for the processor and check if there are reference material as manuals, application notes and example codes beyond technical support. We try to prevent with this verification the selection of some processor that have the best cost/benefit relation but which development tools are expensive and there is not reference material or technical support for the development.

Also it must be observed if the selected processor has a family (element group with similar features) that have another pin to pin compatible processor with more memory capability or some improved peripheral that can be used as replacement. The processor replacement brings the possibility of software improvement without hardware modifications of the embedded system.

REFERENCES

[1] BANNATYNE, R.; VIOT, G.; Introduction to microcontrollers. WESCON/97. Conference Proceedings, pp. 564-574, November 1997.

[2] BERGER, Arnold S.; Embeded System Desing – An introdustion to Processs Tolls, & Techniques. CMP Books, Lawrence, Kansas , USA, 2002.

[3] CONTE, Thomas M.; Choosing The Brain(s) of an embedded system. Computer, IEEE, vol. 35, no. 7, pp. 106-107, July 2002.

[4] Freescale, CodeWarrior Development tools, http://www.freescale.com/webapp/sps/site/homepage.jsp?nodeId=012726&tid=FSH (current Feb. 15, 2006).

[5] GILMOUR, P. S.; How To Select tools for microcontroller software. Spectrum, IEEE, vol. 28, no. 2, pp. 37-39, February 1991

[6] IAR Systems, http://www.iar.com (current Feb. 15, 2006).

[7] JAIN, Raj; The Art of Computer Systems Performance Analysis. Techniques for Experimental Design, Measurement, Simulation and Modeling. John Wiley & Sons Inc, New York, 1994.

[8] MARVEN, Craing; EWERS, Gillian; A Simple Approach to Digital Signal Processing. A Wiley-Interscience Publication, New York, USA, 1996.

[9] Microchip, Mplab Integrated Development Environment ,http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002 (current Feb. 15, 2006).
[10] Renesas, High-performance Embedded Workshop, http://america.renesas.com/fmwk.jsp?cnt=ide_hew_tools_product_landing.jsp&fp=/products/tools/ide/ide_hew/ (current Feb. 15,2006).
[11] VACLICA, J.J; GILMOUR, P. S.; How To Select a microcontroller. Spectrum, IEEE, vol. 27, no. 11, November 1990.
[12] VALDERRAMA, Carlos A. ; LIMA, Manuel E. ; CAVALCANTE, Sérgio; BARROS, Edna; Hardware/Software co-design: projetando hardware e software concorrentemente. Escola de computação IME-USP São Paulo, Brasil, 2000.
[13] WOLF, Wayne; How many system architectures? Computer, IEEE, vol. 36, no. 3, pp. 93-95, March 2003.
[14] WOLF, W.; FREY, E.; Tutorial on embedded system design. Computer Design: VLSI in Computers and Processors, 1992. ICCD '92. Proceedings., IEEE 1992 International Conference on, pp. 18-21, October 1992.