A COST-PERFORMANCE ANALYSIS OF EMBEDDED SYSTEMS FOR LOW AND MEDIUM-VOLUMES APPLICATIONS

V. Secrieru, assoc.prof.dr. S. Zaporojan, prof.dr.hab. V. Dorogan
Technical University of Moldova

INTRODUCTION

Today's products represent a new generation of capabilities that fuse together sensors, actuators and electronics. Most of manufacturers include embedded systems (EmS) in their products. Because of certain application requirements, engineers have to approach embedded design in a different way than other types of designs.

Depending on the embedded applications, designers need to consider the design constraints. Determining which solution best fits the embedded application often mixes technical and business considerations. Technical ones include design's performance metrics. Business considerations typically address projected volumes, design cost, engineering resources, risk analysis.

This set of challenges is the reality that embedded design engineers face today. As consumer demand increases, manufacturers of embedded devices look for solutions to optimize the cost and performance of their products.

In the literature a number of papers which focuses on the estimation and optimization of the performance of embedded systems for real-time applications can be found [1, 2].

In general, modeling embedded systems constraints, as well as early estimation of embedded systems designs are hot topics. More than ever, embedded systems designers must develop cost-effective systems.

Based on the above points, designers must review available technologies and select which platform will fulfill all the requirements while balancing performance, cost, and time-to-market.

The paper concentrates on the cost-performance analysis of embedded systems for small or medium designs and/or low and medium-volumes applications.

1. OPTIONS IN CHOOSING THE PLATFORMS FOR SYSTEM DESIGN

Generally speaking, an EmS represents an analog-digital electronic system. Usually, embedded systems contain sensors, actuators, and electronics needed for processing the acquired information and controlling the attached actuators. An EmS is designed for operation in the real world. The real world operates in an analog fashion – that is, continuously. For this reason, analog circuits, such as sensors and actuators are used to interface the surrounding environment. On the other hand, the processing electronics is digital. Digital processing units are used for algorithmic control and data manipulation.

While developing EmS, designers need to understand different chips, their capabilities, and their architectures before creating the design. Hardware component selection includes choosing the processor, peripherals, and memories which will compose the EmS. A key decision is choosing the right processing unit [3]. Most embedded applications require highly customized hardware. At the same time, developers want flexible, modular solutions that can be modified to serve new markets or meet future needs.

Depending on the desired flexibility and on the importance of cost (area) and power dissipation, different options exist for the implementation of processing and control algorithms. There are platforms, such as microprocessors (MPU), microcontrollers (MCU), digital signal processors (DSP) in the area of embedded applications.

By taking advantage of MCUs, dedicated peripherals, and predefined software libraries, embedded designers are able to implement their designs quickly with the latest technologies. Depending on the application requirements, embedded designers have a wide range of solutions, from low-cost 8-bit microcontrollers to high-end microcontrollers and top-performing, 32-bit embedded processors families from Atmel, Texas Instruments, Intel, Silicon Labs, to name a few.

On the other hand, programmable logic devices (PLD), application-specific integrated circuits (ASIC), and systems on a chip (SoC) are typical circuits in the area of embedded applications.

Complex programmable logic devices (CPLD) are low-cost devices for any digital control function. Field-programmable gate arrays (FPGA)

are the modern-day PLD technology for building a system or prototype from standard parts. Programmable logic blocks and programmable interconnections allow the same FPGA to be used in different embedded applications [4]. This type of PLD is addressing medium-performance embedded applications. FPGAs have clock rates in the range of 500 MHz. This rate is too slow to meet the operating frequency and throughput requirements in high-performance applications. In fact, FPGAs suffer from post-layout timing issues. After place and route is done, the maximum operating frequency for real designs drops in the range of 250 MHz.

Field-programmable object array (FPOA) is a silicon platform that supports high-speed designs. Unlike FPGAs, which implement functions at the gate level, FPOAs employ higher-order building blocks called “objects” [5]. The timing of both the objects and the communication framework is fixed. This approach allows the FPOA to operate deterministically at frequencies of up to 1 GHz. An FPOA requires some kind of host, such as a RISC MPU or MCU. This requirement puts the FPOA in a gray area between FPGAs and MPU. It is a fully programmable machine, but it needs an external host controller to initially load the bit stream and retrieve the results [6]. Most FPGAs need host controllers, too, although some have hard processor cores on chip, and others have enough capacity to implement a soft processor core in their gate arrays.

ASIC is an integrated circuit customized for a particular use, rather than intended for general-purpose use. Modern ASICs often include intellectual property (IP) cores, entire MPU/MCU, memory blocks and other large building blocks. ASICs have long been natural choices for high-volume products. While ASICs might have better costs for very high volumes, they also have very high non-recurring engineering (NRE) costs and it is very hard to get the volumes and sizing right. The NRE cost refers to the one-time cost to research, develop, design and test a new product. The NRE is unlike production costs, which must be paid constantly to maintain production.

In the ASIC domain, as the transistor feature size decreases, the NRE costs increase dramatically. This is particularly important in the current economic environment where nobody wants to commit to an order for hundreds of thousands of units and huge NRE cost. For smaller designs and/or lower production volumes, FPGAs may be more cost-effective than an ASIC design.

Platform or structured ASIC design is a relatively new term in the industry, resulting in some variation in its definition. The basic premise of a structured ASIC is that both manufacturing cycle time and design cycle time are reduced compared to cell-based ASICs. Structured ASIC technology is seen as bridging the gap between FPGAs and cell-based ASICs. Because only a few layers of chip must be customized for any given design, the NRE costs are significantly lower than a cell-based ASIC development, where a full mask set is needed to be produced for every design [7, 8].

Platform (structured) ASIC is an intermediate technology between ASIC and FPGA, offering high performance, a characteristic of ASIC, and low NRE costs, a characteristic of FPGA. Using structured ASIC allows embedded products to be introduced quickly to market and to have lower cost.

A system on chip is a circuit that integrates all components of an electronic system into a single chip. A SoC consists of a MPU, MCU or DSP core(s), memory blocks, peripherals, industry standard external interfaces, analog interfaces, such as analog-to-digital converters (ADC) and digital-to-analog converters (DAC). SoCs can be fabricated by several ways, including ASIC-based and FPGA-based technologies. SoC designs usually consume less power and have a lower cost and higher reliability than the multi-chip systems that they replace. However, like most VLSI designs, the total cost is higher for one large chip than for the same functionality distributed over several smaller chips, because of lower yields and higher NRE costs.

Figure 1 shows a typical product volume versus cost analysis for both platform ASIC and cell-based ASIC. The crossover product volume points are fully dependent on the complexity of the design and would differ from one application to another [8].

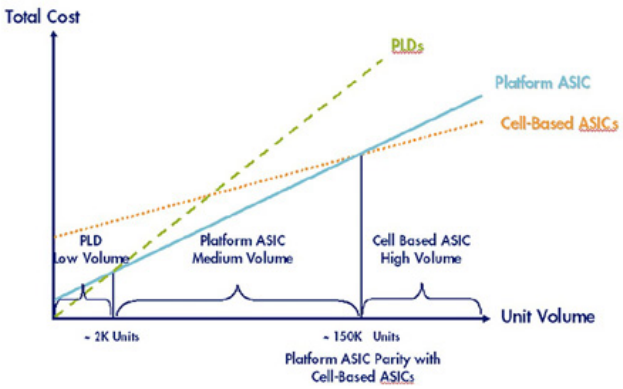

Figure 1. Product volume vs. cost analysis.

At the decision point between the choices of a device/platform, it is important to consider all the

factors involved. These factors typically are: cost analysis (NRE costs, unit cost, design resources, design tools), market pressures, technical feasibility of each solution in achieving end-device target features, engineering resources, and risk analysis.

It is also very important to consider available migration paths from a currently selected device/platform to another device/platform.

If the anticipated volumes of the end-EmS aren't too great, FPGAs could be a less costly alternative to ASICs and SoCs.

With the increasing density and complexity of FPGAs, programmable architectures are becoming more attractive for implementing EmS designs. FPGA manufacturers are continually adding increased logic and memory, I/O standard support, and DSP functionality, which increases system performance and minimize the need for peripheral devices. FPGA companies and their partners are also providing low-cost IP cores that deliver the building blocks necessary to create targeted embedded systems [4].

Embedded system often uses a MCU to perform all tasks related to the application. However, it would be really hard to solve a computation-intensive application with a low-end or even mid-level MCU. As a result, most of today's designs are implemented using an off-the-shelf standard MCU and a low-cost, off-the-shelf FPGA. This MCU-plus-FPGA combination has effectively replaced ASICs in many embedded applications. Motivations for using commercially available off-the-shelf MCU/FPGA include hopes for reduction of overall EmS development and costs.

Selecting a MCU that integrates commonly used components will help the designers to reduce overall EmS cost, design complexity and development time. Another important design consideration is the flexibility to accommodate change quickly and easily without driving up development cost.

A MCU with the right mix of peripherals implements the most of the application, while the FPGA is used for performing compute-intensive algorithms. The re-programmability of both the MCU and the FPGA offers high flexibility to meet changing market needs. There are no NRE costs and time-to-market is as short as possible.

A problem, which can appear, is the lack of free pins. We can take a larger, faster MCU, if it tackles our problem. An alternative solution is to use a low-cost CPLD as a bridge to perform all tasks related to the I/O space, thereby freeing up our low-cost MCU to perform other operations. Because the CPLDs are non-volatile, single-chip solutions, they are very easy to incorporate into a system to solve board-level issues like insufficient I/O pins on a MCU. For small designs a MCU-plus-CPLD combination can be an excellent solution, where the CPLD is used for custom logic and I/O expansion. Another possibility is to design FPGA/CPLD-based embedded systems. This choice can be done for medium/small embedded designs.

2. EMBEDDED HARDWARE COST: EARLY ESTIMATION

A comparative analysis of digital systems must be based on well defined criteria. For the purpose of cost-performance analysis a crucial criterion is given by the so-called quality metric [9]. The quality of a system is the weighted geometric mean of the performance P and the reciprocal of the cost C:

$$Q = P^{1-q} / C^{q} . \quad (1)$$

The weighting parameter $q \in [0,1]$ determines whether cost or performance has a greater impact on the quality. Therefore, q can be considered as quality parameter [9]. So, at one extreme, we have the case $Q = P$, when only performance counts. On the other extreme, the case $Q = 1/C$ is placed, when costs are strictly limited and must be reduced to some minimal values, according to the application constraints.

In our opinion, for a realistic quality metric in the space of embedded designs, the quality parameter should probably be in the range over 0,5. This is because, usually, cost constraints are more important than performance ones, thus $q \geq 0{,}5$.

Obviously, modeling and estimation mechanisms should be applied to evaluate the cost and performance of a new embedded device. After then, the design can be evaluated according to the criterion (1).

In the above context, this section presents a mechanism for early estimation of the embedded hardware cost.

Because absolute cost measured in currency is changing every year, it is reasonable to define cost in terms of such parameters that influence cost. Our analysis considers the key factors, such as the number of chips, area of the chip packages, the amount of memory required, ADC requirements, dimension of the I/O space, and area of the printed circuit board (PCB). These parameters derive from an implementation on a concrete architecture.

The proposed mechanism cannot account for all the factors which affect cost, but it offers the possibility to isolate the most important ones, especially when comparing two closely related architectures. Our intention is to focus on the differences and discuss the ways they affect the cost. In this way, we can estimate the influence of the design decision on the hardware cost of embedded device.

Generally speaking, a new embedded product cost consists of both software and hardware costs. Each of them has two components:

- NRE - a one-time cost to research, develop, design and test;
- A per-unit manufacturing cost.

High NRE costs are unacceptable for low or medium-volumes applications. In our analysis we consider two closely related architectures whose differences are limited to a few critical design choices. The first is a MCU-based architecture, and the second is a FPGA-based architecture. Each approach has its own advantages and disadvantages, and they in turn influence the cost of the design.

Note that the MCU and FPGA offer high flexibility and meet time-to-market requirements. Besides, there are no NRE costs for selected chips. The software development effort is comparable and not expensive. In conclusion, the MCU-based and FPGA-based architectures are quite suitable for low or medium-volumes applications.

Based on the above points, our goal is to estimate the influence of the design decisions on the hardware cost of MCU-based and FPGA-based embedded products.

Let a hardware cost is denoted as C_H. This cost is part of the per-unit production cost, and can be written as:

$$C_H = C_H^{fix} + C_H^{\text{var}}, \qquad (2)$$

where the first term represents the cost of the hardware components, such as sensors and actuators of a design. We consider this cost is fixed (common) for each approach.

The second term in formula (2) represents the cost of the hardware components, which can vary for the architectures approaching. For a MCU-based embedded design this cost can be expressed as:

$$C_H^{\text{var}} = C_{MCU} + k_1 C_{CONV} + k_2 C_{MEM} + C_{CPLD} + C_{PCB} \qquad (3)$$

where the following notations are used: C_{MCU} - microcontroller cost; C_{CPLD} - cost of the CPLD; C_{CONV} - analog-to-digital conversion cost; C_{MEM} - memory cost; C_{PCB} - PCB per-unit cost. Coefficient k_1 is equal to zero if the conversion is embedded onto the MCU or not used, and it is an integer if the application needs external ADCs. Coefficient k_2 is equal to zero if the memory is embedded onto the MCU or not used, and it is an integer if the application needs external memories.

For a FPGA-based embedded design the hardware variable cost can be written as:

$$C_H^{\text{var}} = C_{FPGA/CPLD} + k_1^1 C_{CONV} + k_2 C_{MEM} + C_{PCB} \qquad (4)$$

where coefficient k_2 is similar to (3). Coefficient k_1^1 is equal to zero if no conversion is used, and it is an integer if the embedded application needs analog-to-digital conversion.

In order to present how our early estimation mechanism works, it was necessary to take numerical values for $C_{FPGA/CPLD}$, C_{MCU}, C_{CONV}, and C_{MEM}. The presented analysis is based on real data for low-cost MCUs and low-cost FPGAs/CPLDs. Then, a lot of low-cost ADCs and memories were analyzed. As a result, the range of real values was established for the coefficients k_1, k_1^1 and k_2. Finally, our estimation mechanism uses the PCB costs model described in [10].

The PCB cost of a product is shown in figure 2 as a function of the dimension of I/O space of the embedded application.

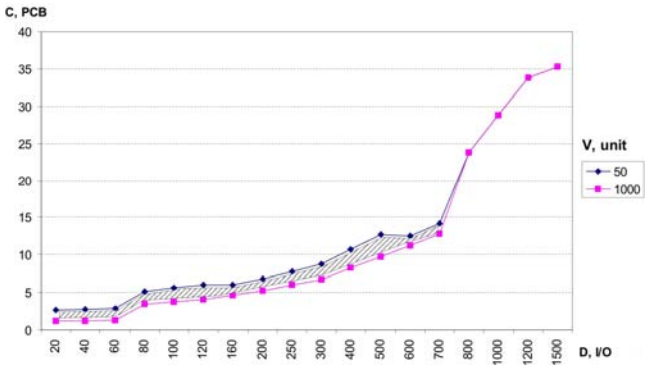

Figure 2. Variation of the PCB cost.

Variation of the PCB cost is given for two different volume of production of the product. Note that the cost of the PCB has been estimated taking into account real prices on the market of the Republic of Moldova.

Figure 3 shows the hardware costs for MCU-based implementation of the embedded designs for one thousand unit volume. For smaller designs (k_1= k_2=0) and for dimension of the application space up to 80 I/O, the cost is at a minimal level. The cost curve moves upward with dimension of the I/O space. This is because the cost is influenced by adding CPLDs. The low-cost CPLD is to be used as

a bridge to perform all tasks related to the I/O space. Each CPLD package increases the area of the PCB which is considered in figure 2.

When designs need extended ADC and memory capabilities (k_1=5 and k_2=2) the hardware cost will increase significantly.

Figure 4 illustrates the cost of the hardware for FPGA-based design. For smaller designs (k_1^1=k_2 =0) and for dimension of the application space up to 80 I/O, the cost is at a minimal level. The cost curve also moves upward with dimension of the I/O space.

Designs with ADCs and extended memory capabilities (k_1^1=5 and k_2 =2) results in more costly hardware.

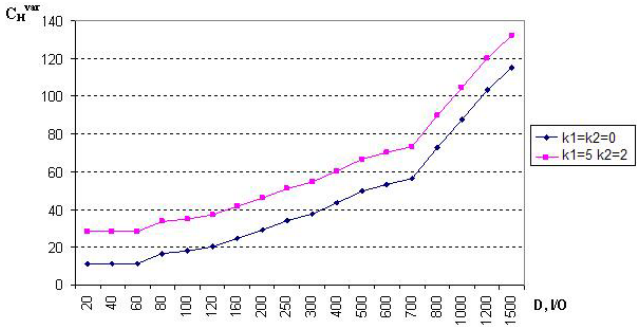

Figure 3. MCU-based hardware costs.

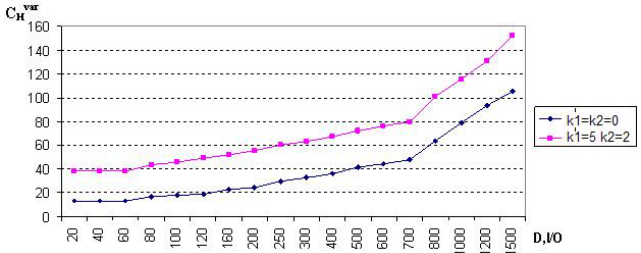

Figure 4. FPGA-based hardware costs.

Figure 5 presents a comparison of the hardware costs of two considered architectures. It can be observed that a FPGA/CPLD implementation takes advantage over a MCU-based architecture for low-performance designs, when I/O space of the application is medium or large.

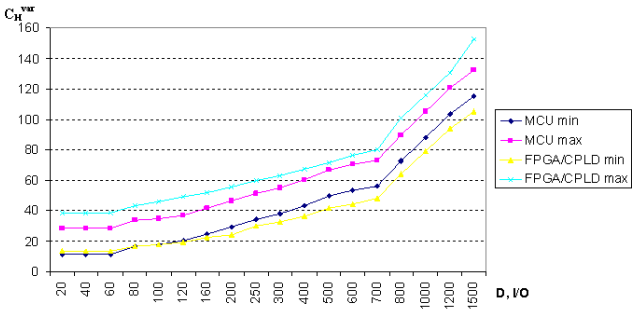

Figure 5. Comparative hardware costs.

On the other hand, for small I/O space the MCU-based approach is less costly alternative. Finally, an MCU-plus-FPGA combination can be considered for medium-performance designs and medium or large I/O space of the application.

CONCLUSIONS

The proposed approach can be used as a method for early estimation of the cost of an embedded device. The size of embedded product can be also estimated at the early stage of the embedded design. In this way, designer engineers can optimize embedded system to meet the design constraints or reduce the size and cost of a new product. The presented approach is suitable for low up to medium-volumes applications.

References:

1. ***Eles P., Kuchcinski K., Peng Z., Doboli A., and Pop P.*** *Process scheduling for performance estimation and synthesis of embedded systems.* In *Proceedings of the CONTI'98. Timisoara, pp.246-257, 1998.*

2. ***Altenbernd P.*** *Deadline-monotonic software scheduling for the co-synthesis of parallel hard real-time systems.* In *Proceedings of the ED&TC'95. Paris, pp.190-195, 1995.*

3. ***Wolf W.*** *Computers as Components. Principles of Embedded computing system design. Morgan Kaufmann Publishers, 2nd edition, 2008.*

4. *http://www.altera.com*

5. *http://www.mathstar.com*

6. ***Halfhill T.*** *MATHSTAR CHALLENGES FPGAs: New reconfigurable-logic chips have massively parallel arrays. Microprocessor Report, July 24, 2006.*

7. ***Johnson J.*** *Using customizable MCUs to bridge the gap between dedicated SoC ASSPs, ASICs and FPGAs. Retrieved from http://www.embedded.com*

8. ***Khalilollahi Y.*** *What platform ASICs are and when to use them. Retrieved from http://www.eetimes.com*

9. ***Mueller S.M., Paul W.J.*** *Computer Architecture: Complexity and Correctness.* Springer, 2000.

10. ***Secrieru V., Zaporojan S., Dorogan V.*** *The study of the variation of costs of the printed circuit board in embedded systems design. Meridian Ingineresc. Nr.1, pp.28-31, Chisinau, 2012.*

Recommended for publication: 15.02.2012.