Chapter 1

INTRODUCTION

Embedded systems are prevalent in today's society and promise to be even more pervasive and found in many of the things we interact with in our daily lives in the near future [122]. Applications vary from today's airplane jet or car controllers, and communication devices like cellular phones to the future's autonomous kitchen appliances, and intelligent vehicles. The trend in semiconductor industry is that the Internet and e-commerce will change our lives and impact the semiconductor industry even further. A market forecast [29] of the structure of worldwide electronic production in 2010 is presented in figure 1.1. It shows that 60% of electronic production will be among embedded applications. One sector that stands out of the embedded market is communications (30%), while consumer (15%), industrial (12%) and automotive and defense (14%) have about the same percentage.

A second trend is that in the five key industrial sectors with a high share of microelectronics, software plays a more and more important role [54]. The prediction for the total growth from 2002 to 2015 is to 128% and more than doubles the total research and development (R&D) investment growth. Industries where software played a minor role in 2002 (automotive, medical equipment) will increase their effort to more than a third of their R&D volume and sectors which have already a high software rate (consumer electronics, telecom equipment) will raise the software R&D budget to over 60%. These figures show that the market share for embedded systems is growing and a high portion of industrial research and development activities will be dominated by embedded systems [65]. The complexity of embedded systems will inevitably increase to meet numerous demanding requirements.

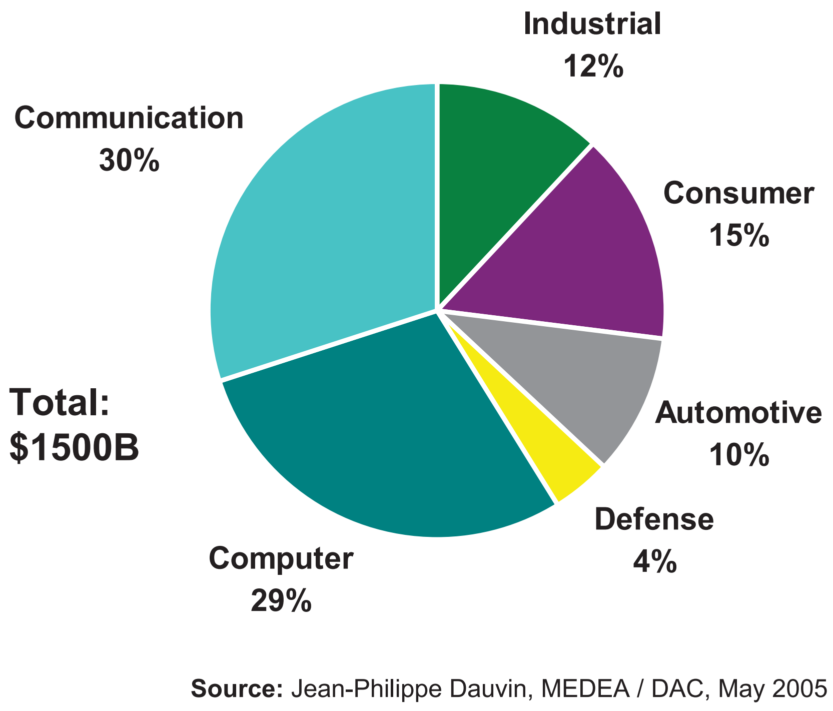

Figure 1.1. Structure of worldwide electronic production in 2010 [29].

1.1 Embedded system requirements

Embedded systems have to satisfy an increasing number of requirements. The *time-to-market* period is getting especially important as product life-cycles are constantly decreasing, such as in multimedia, telecommunication and consumer electronics. A rapid development is crucial for a successful product placement while functional as well as non-functional requirements are essential. The embedded system development process is studied from an economic viewpoint in [65].

Functional requirements

Correctness is a fundamental requirement to guarantee that an embedded system properly operates. Many hardware- and software-tests are carried out to verify functional behavior.

Flexibility is used in two different senses: configurability and re-configurability. The advantage of a configurable system is that the manufacturer can simplify the development for a variety of product lines. Re-configurability gives the customer the ability to use a device for different applications. For example, a firmware update is much less expensive than the exchange of hardware components in case software errors are detected.

Reliability varies strongly for different embedded systems. While a system crash of a cellular phone is acceptable once a year, a similar rate for safety-critical systems like in aerospace applications would be disastrous. Fault tolerance and quality of service (QoS) are terms used to describe the necessary robustness of an embedded system.

Non-functional requirements

Timing behavior denotes the time delay within a software task finishes its computation. For most embedded applications, a calculation has not only to be correct but has to finish before a specified time period. The term timing behavior covers a broad spectrum: from best effort strategies and quality of service, e.g. in networking domain, real-time constraints, e.g. for MPEG video processing, to safety-critical requirements, e.g. airbag in automotive.

The term *real-time* is often used when the embedded application reacts to signals from its environment. Such systems are further distinguished into *soft real-time* and *hard real-time* systems. In soft real-time systems, timing behavior is considered an important aspect yet is not essential to correct functional behavior. The quality can be reduced in case of timing bottlenecks and it is considered as correct functional behavior. As an alternative, the software task could be switched to a different processing mode, in which less accurate results are computed within a shorter time. In hard real-time systems, the application must finish before a pre-defined deadline. The term deadline denotes the longest acceptable time period before the computation has to finish. Examples are engine control software in automotive, flight control software in avionics systems and pacemakers.

Micro-systems with real-time behavior have been developed to make driving more secure. Micro-systems are embedded systems, in which electronic components are combined with micro-mechanical, micro-optical or micro-fluidic components. The automotive supplier Continental-Temic developed a lane-keeping system that automatically controls the car navigation. Another product by Ibeo is the Alasca XT laser, which observes the area in front of a vehicle from 30cm to 200 meters and can compute up to seven different functions simultaneously [33] [41].

A low *power consumption* is essential for mobile devices. A lower power dissipation allows lighter and smaller products as well as longer operation times. Other important product requirements are size, weight and design.

In order to meet these requirements, an appropriate hardware and software architecture has to be chosen. Often requirements have opposite goals: a short time-to-market window versus an efficient hardware and software design. This thesis focuses on timing requirements and discusses the challenges of advanced hardware architectures.

1.2 Architecture and application properties

In this section we survey properties of modern embedded architectures including processors, memory hierarchy and operation systems. Then, we discuss their impact on the timing behavior of software applications.

Architectures

Embedded architectures consist of one or several processors with several memory devices and peripheral units. An example of an embedded architecture is shown in figure 1.2. The TriCore 1796 processor [51] is used in the automotive domain for engine control units (ECU)s. It consists of three cores, an instruction cache, scratch-pad memory, other memory units, several IP components, for example a CAN bus interface and several busses. The scratch-pad RAM, which is a SRAM memory, holds frequently executed instructions, e.g. of the operating system, to prevent cache replacements.

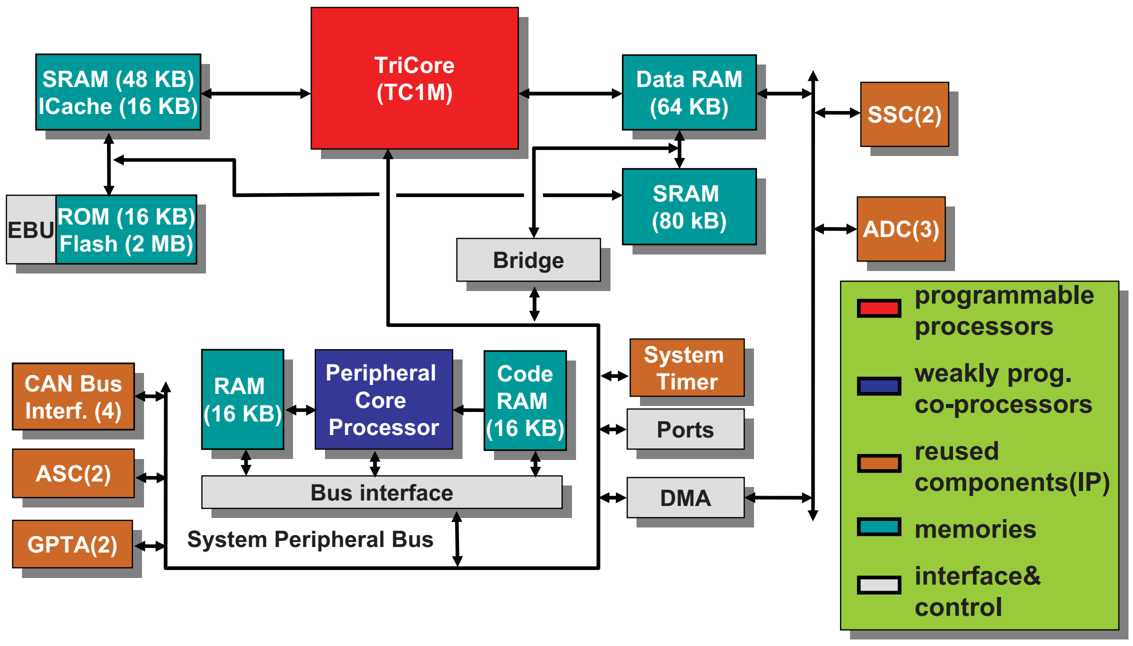

Figure 1.2. TriCore 1796 Architecture [51].

Such highly integrated systems are also called system-on-chip (SoC) because many components are integrated on a single chip.