Chapter 6

Integrated Circuits for Intelligent Systems

The term Intelligent System is being used in a more extensive and inclusive way. It covers systems that perform intelligent functions, understanding by intelligent that it seems to think and decide, based on information it has and information it takes from the environment. An easy example to understand this concept is a Smartphone's. They are called smart because they read the environment and take actions based on what they find: they detect if Wi-Fi networks are available, they use the GPS to know where they are, and so on. In consequence of those readings it can tell the user what to do or use. The decision making process involves hardware and software within the smart device. A robot is also smart, in the way its software and hardware allows him to be. The more sensors it has, and the more complex its program is, it will be more intelligent and can be able to control more output elements and perform more actions.

Nowadays people are more familiar with intelligent systems and, by now, people born in this decade cannot imagine a functioning world without them. We all now count and rely on intelligent systems for everyday activities such as electronic banking, automated parking, electronic document sharing, and permanent communication capabilities, among others.

More recently, connectivity capabilities are becoming the more important feature of an intelligent device or system. Specific functionalities of each device can still grow and innovate, but it is more and more important that a device can connect with other, similar or different, devices and systems.

This is how the concept of smart cities, smart grid, and smart cars has been defined. A smart system consists now on a set of interconnected devices, either they are of the same nature and purpose, or not.

The following examples, although its outside is known by everyone, show that all systems are the same inside: they have a processor to execute the instructions, a memory to store that instructions, sensors to detect what they need from the environment, actuators to perform the functions, and communication capabilities to connect with others.

Smart house: The control system includes light sensors, motion sensors, proximity sensors, temperature sensors, timers, in order to operate the lighting network, the alarm system, the air conditioning system, the access doors, and so on. The more elements it has, the smarter it looks.

Smart building: Same idea than the smart house, and additionally it may include collective access control, separated areas air conditioning, access record, access reports, energy efficiency programs, personnel data base. You can notice that sensor and actuators sound similar than those in smart houses, but processing and storage capabilities need to be larger.

Smart parking: Commercial centers, Corporate buildings and Residential complexes use to have access control, assigned placement, and space optimization. For this, they need a smart control, like the ones mentioned in previous smart systems: sensors, actuators, program, data management and storage, user interface. The administration of space, maintenance cost, users and rates are now common elements in parking systems. And what if you need to know in advance, prior to your arrival to the parking lot, if they have spaces available and what is the current rate? The system should be available through a web page or an app, so users do not discard this parking from their options by not having that information available anytime and anywhere.

Smart grid: As the green movement becomes more important and global impact on energy resources is a regular element in business decisions, also is becoming important the smart grid concept in large cities. The grid that supplies energy to the city (cables, stations, transformers, and measuring devices) can be aware of the user's consumption habits and needs. The energy demand of a city, and of every city district or area, depends on the season, the time of the day, and the day of the week, among other factors. It is useful for the energy provider to know the demand patterns so they can manage the energy distribution, maintenance tasks, rates, and so. A smart greed consists of a regular energy grid plus the needed sensors and measurement devices to know and predict the energy patterns and take decisions for energy optimization and use.

Smart cars: People still use to name the smart system of a car as the "car computer". It was a proper name when the concept began, because the first cars used to have one system that received signals from simple sensors like rpm readers, impact sensors, and proximity detectors. With this signals and a simple program, a central computer decided things like activate the airbags when an impact occurred, activate the ABS when the regulars breaks were not enough for an efficient speed reduction, and to activate a bip signal if the car was too close to the next car, the sidewalk, or an object behind. But, as processes became more complex, the need for independent controls arose and cars had more than one computer. For instance, a state of the art car, these days, has more than 60 independent intelligent controls or "computers": one for the fuel injection system, one for ABS, one for the security tasks, one to collect and store all the info needed for the maintenance procedures (have you noticed that today's mechanics is not about checking under the car to find out what the malfunction comes from, but to download the computer information to analyze the sensor's measurements over time, and what the system is concluding the problem is), one for entertainment, and the list of new needs will never stop. And of course, all the systems need to know what is going on with the other sub-systems, as their decisions depend on the other's decisions. At this point, the need for a local network between subsystems is needed, so all of them require communication or interconnection capabilities, and a central system to coordinate the operation between them.

Smart city: What will happen next, after many of the systems living in a city have their own intelligence? The obvious next step is to connect them all together and see what additional intelligence can result from that. The smart grid can know the energy consumption patterns from the smart houses, smart buildings and smart parking. The smart cars can take advantage from the traffic information collected from the City Traffic System. All the smart systems in the city can be accessed using a data center, so any store, service center, weather center, manufacturer and user, can access information in real time and take smart decisions.

Smart manufacturing: When a company already has stable in quality control schemes and lean manufacturing, decides to move towards smart Manufacturing.

We are accustomed to using terms like Smartphone, Smart TV and Smart Cars; soon it will be Smart House, Intelligent Building, Smart City and finally, intelligent planet, meaning that an intelligent system uses its resources to create, manage, and use information to help you make decisions and actions wisely.

Referring to the context of manufacturing, it covers to have information in real time, ensuring its flow and access, and maintain integrated and scalable on which to base all business decisions. This will fundamentally change the way products are invented, manufactured, transported and sold.

When a company decides to join the Smart Manufacturing trend, it will inevitably find other issues when integrating intelligence: Safety and interoperability of data, modeling of production, simulation market, Sustainable Production, Integrated processes, Sensor Networks, Knowledge Management, Zero emissions and, of course, cloud Computing. These terms are not new, but now they must be integrated into a system that includes planning, production, operation, and vision of the company.

Now, if we understood the idea of system intelligence, and we were convinced that we must make the transition, we can classify the next steps into 3 phases:

Integrate into one system all the information from all lines, processes and products of the company. It will take time but it is essential. Since IT resources, sensors, motors, automatic controls, and software to manage production, but each is an efficient island.

Make models and simulations that allow flexible manufacturing, demand production and product customization when markets change rapidly.

When the previous phases progress, create scenarios for innovation, and manage to break the paradigms of today's markets. These breaks are generated by innovative technologies in processes and products. This phase will reverse the traditional chain where the consumer was forced to buy what it was mass produced.

6.1 The Smart Systems and the Integrated Circuits

Having said that, a question arises: What’s the link between Smart systems and Integrated circuit design? The answer is that the core of an intelligent is, mostly for sure, an integrated circuit. Not a generic or over-the-counter circuit, but an Application Specific Integrated Circuit.

If you were able to open a smart phone, or the car computer, or the robot brain, you will find, at the end, an integrated circuit specifically designed for that purpose. More often than not, it will be only one integrated circuits that includes processor, memory, communication ports, and even sensors and actuators.

What is important about this book is that the design procedures described here are universal and non-dependant of the application or need you want to solve.

ASICs for commercial products. ASIC stands for Application Specific Integrated Circuit, so it means somebody detected an opportunity to develop an original circuit to attend a specific need, and then developed a circuit specifically for that purpose. Examples of this are: A commercial brand for refrigerators decides, for the first time, that it will be useful to have internet connection available in its refrigerators, so the customer can connect from any place and check what's in the fridge, or to send a list to the store every weekend of what is missing and have it delivered to your home. The circuit designer starts from the previous circuit (not from scratch) and adds the needed circuitry to complete the monitoring and detection tasks. If successfully designed, it will lead to what is called an intelligent system, because it apparently understood what happened in your refrigerator, decided that you needed more bananas and milk, and ordered them for you. In this example, you design a circuit specifically intended for that application. The understanding and decision making was made by the carefully designed combination of circuit and program. By the way, if you take this same circuit and connect it to your microwave oven it will not know what to do or will do something wrong. So, this is an Application Specific Integrated Circuit.

It may have sounded casual, but the fact that you should not start a design from scratch, but from what has been already designed before, is one of the main rules of circuit design: You should never start a new design assuming nothing like it has ever been done before. If you start from zero, it will take you more time than other designers to get your idea implemented, and by then you will be out of the competition for your idea's market.

In this time and age where there is a solution for almost everything, you may think: what can I design if everything is already done. Nothing more wrong than that. The more complex our environment is, the more opportunities for ideas we have.

6.2 ASIC for Customized Applications

There are design opportunities that may find broader fields of application, meaning that you want to design a circuit that has some of the functionality defined and limited by the cores it has inside, but other functionalities can be defined by the final user. Which is final by design is the hardware, of course, but you can load a small operative system or a program application that allows the user to program his own application, purpose specific, and can load it in the memory Space, you as the designer, left available for that.

In the same sense, a designer can provide external access to the modules inside the integrated circuit, by having the in-chip address bus, available on external pins. This will increase the pin out of the IC, the packaging and will move the place-and-route, but the benefit of an open system supersedes the design difficulties. The sample application presented in this book shows how this can be done. It illustrates concepts like:

Open architecture: the integrated circuit has the address bus available in the external pin out, so other devices using the same bus can be connected to it. As long as address assignment stays compatible, the interconnection of other devices has no limit through this bus.

Programmable: a memory space is reserved to download a different application program, or additional functions to those already loaded.

Configurable: Many functions are preloaded in the memory chip, by design, but in the user interface of the application some of those functions can be disabled, so they do not take execution time.

6.3 Design and Market Trends

The integrated circuit market have been revolving around developing faster, smaller, and less power consuming components, and it will continue to do so unless a completely different technology is developed.

Each of the 3 variables depends on two factors: the technology and tools available at the moment, and the designer technical capabilities and knowledge. None of them completely compensates the other, so both, the technology and the developer, need to be good to achieve a usable product.

Faster: As silicon transistor based technologies became smaller, they are also faster. That way we went from tenths of nanometers to a single digit figure. As for edition time of this book, smaller has always been more expensive to fabricate, so older machines working with larger transistor sizes are cheaper but no big companies want cheaper and slower circuits. That way, the cheaper fabrication options are good for beginners or universities on small budgets. The developer competences come in play when designing the HDL program: a designer needs to keep in mind that loops, variable assignment, variable sizes, data transfers and so, will finally translate into circuits. And circuits can be efficient in their implementation or not. A simple example to understand this idea is to think on a simple adder being implemented in a protoboard by 2 students; each of them can have a different idea on how to do it, then use different gates or array of gates, or use multiplexers. At the end, the two circuits will be different in appearance, in size, in gate count and, consequently, in response time. This illustrates how your programming style impacts your design size and speed. Besides, your programming language is also a factor: designers who prefer to program in C see this idea clearly: when a C program is transferred to HDL the sizes are completely different meaning there is no optimization possible when you program in a high level language that does not allow you to see how your program will look when in the final language.

Smaller: As mentioned in the paragraph above, silicon based technologies became smaller over the years. There is a size limit as connections and transistors need to transport electrons, and electrons have dimensions. Under this idea, the smallest a wire or transistor can be, is related to the electron size, so it can freely transit through it without reducing its speed or overheating the wire. About the designer skills, core size and placement are the main issues in circuit size. Core size comes from the synthesis process, where HDL design is translated into circuits, and the programming style impacts the resulting circuit. Once each core is size optimized, the pin placement determines where each core is going to be placed within the integrated circuit; space between cores is needed for interconnections, so you want to be safe and leave more space than needed, but external pin-out may demand that cores be placed differently than interconnections suggest. There is no single solution for a good route and placement, which is one of the more careful design processes, other than the cores design.

Less power: Power consumption by an integrated circuit is separated in two types: active and passive consumption. Passive is the power it takes to keep the circuit ON even if it is not running the application or any part of the program; this state could be named as Stand-by. Active consumption is when the circuit is operating or running. Of course active consumption is larger than passive, but active is not a fixed or constant figure: Not all parts of the hardware and software architecture are being used in every function of the system, so the power consumption rate depends on the function currently executed or performed by the circuit. In a simple integrated circuit, as a 4-NAND gate circuit, it is easy to estimate the amount of power being consumed if one, or two, or four gates are ON. For an integrated circuit that has a running processor executing a complex application program, a power simulation is needed to determine the low and high consumption peaks.

Modularity is also a trend, meaning that a complete system is built over interchangeable blocks that can define capabilities and functionality using a common platform. Modular systems are upgradeable by definition, as the user can change the processor, the memory or storage capabilities, the communication components, and so on.

Modularity in hardware, for circuit designers, is a constant in any design, as no one starts a new design from scratch, but from the previous product or from something similar. Design teams work by developing independent and coherent blocks that will finally complete the hardware architecture. But for the final user of a product, like a cell phone, a computer or a tablet, the product is a closed hardware architecture where the user chooses the architecture when buying a product, and can do nothing or just a little to add or upgrade circuits or blocks in it.

Modularity in software has been here for a long time, as it is the concept behind Apps: the main operative system is the foundation, and the apps are the added blocks to complete a different software system based on the preferences of each user.

Figures 6.1 to 6.4 show examples of applications for embedded and intelligent systems, in every day uses.

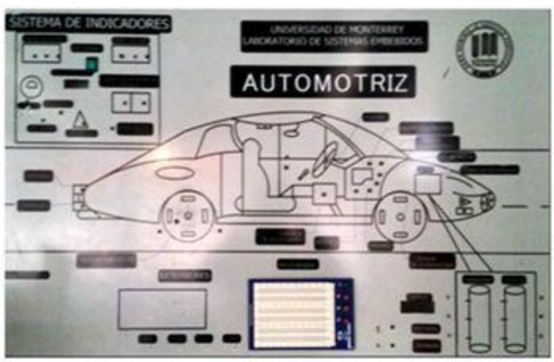

Figure 6.1 *Example of embedded systems in automotive.*

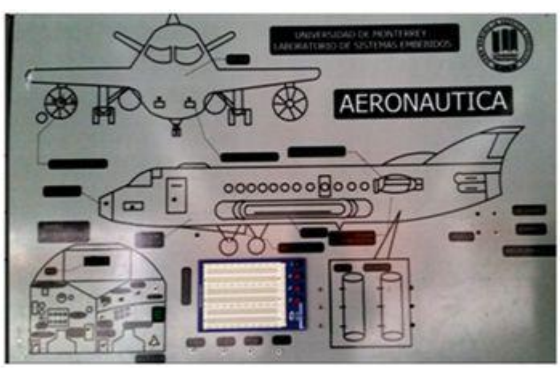

Figure 6.2 *Example of embedded systems in aeronautics.*

Figure 6.3 *Example of embedded systems in safety monitoring.*

Figure 6.4 *Example of embedded systems in home appliances.*

Glossary

ASIC: An integrated circuit designed for one particular use, such as substituting many small integrated circuits with a larger but specific one.

Address bus: A unidirectional set of signals used by a processor to point to memory locations in which it interested, in a certain device or circuit.

Analog: A continuous value that most closely resembles the real world and can be as precise as the measuring technique allows.

Analog circuit: A collection of components used to process or generate analog signals.

Bit: A zero or one value or representation in the binary language of computers.

Byte: a package of 8 bits.

Clock tree: This refers to the way in which a clock signal is routed throughout a chip. This structure is used to ensure that all of the flip flops see the clock signal as close together as possible.

Custom circuit: An Integrated circuit designed and manufactured for a particular customer.

Data Bus: A bidirectional set of signals used by a computer to convey information from a memory location to the central processing unit and vice versa.

Design flow: Design flows are the explicit and graphic combination of electronic design automation tools and representation to accomplish the design of an integrated circuit.

Die: The small piece of the wafer on which an individual semiconductor device has been formed.

Digital Circuit: A collection of logic gates used to process or generate digital signals.

Diode: A two terminal device that conducts electricity in only one direction.

EDA: Electronic design automation is a category of software tools for designing electronic system such as printed circuit boards and integrated circuits.

Hardware: Generally understood to refer to any of the physical portions constituting an electronic system, circuit boards, power supplies and monitors.

Hertz: Unit of frequency. One hertz equals one cycle or one oscillation per second.

IC layout: Also known as mask design, it is the representation of an IC in terms of planar geometric, so components can be visualized and placed.

Integrated circuit: A complete electronic circuits composed of interconnected diodes and transistors on a single semiconductor substrate.

IP Core: Reusable unit of cell or chip layout. IP cores are used as building blocks within chip designs.

Micros: A micrometer, or one-millionth of a meter.

RAM: A data storage device from which data can be read out and into, which new data can be written on.

Semiconductor: A material (silicon or germanium) that has four electrons in its outer ring and is a poor conductor of electricity.

Silicon: The basic material used to make the majority of semiconductor wafer.

SRAM: A type of RAM that has self contained memory circuitry. Memories are categorized by speed and by storage capacity.

Transistor: A three terminal semiconductor device used mainly to amplify.

Via: A hole filled or lined with a conducting material, which is used to link two or more conducting layers in a substrate.

Wafer: A thin disk, from 3 to 8 inches in diameter from silicon or other semiconductor material. The same or different integrated circuits can be printed in one wafer.