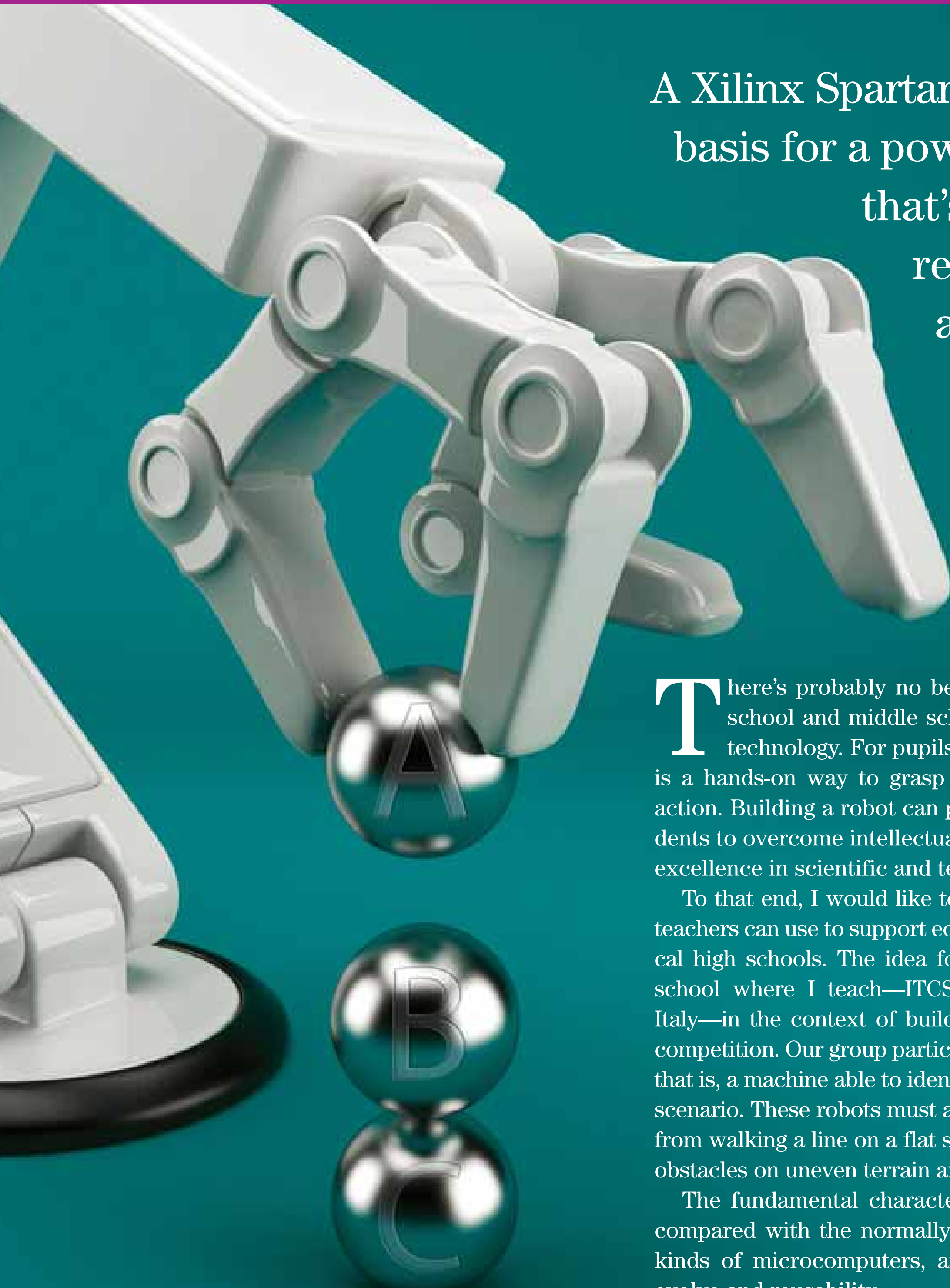
FPGAs Enable Flexible Platform for High School Robotics



A Xilinx Spartan FPGA forms the basis for a powerful teaching tool that's able to evolve, reinventing itself according to student needs.

by Giulio Vitale
Professor and FPGA Design Consultant
ITCS Erasmo da Rotterdam, Bollate, Italy
gvitale@tiscali.it

There's probably no better vehicle than a robot to get high school and middle school students hooked on science and technology. For pupils in grades 8-12, tinkering with a robot is a hands-on way to grasp new ideas and to see technology in action. Building a robot can provide a powerful motivation for students to overcome intellectual difficulties so as to achieve levels of excellence in scientific and technical subjects.

To that end, I would like to present an educational platform that teachers can use to support educational robotics activities for technical high schools. The idea for this platform was born at the high school where I teach—ITCS Erasmo da Rotterdam, near Milan, Italy—in the context of building an entry for the RoboCup Junior competition. Our group participated in the category "rescue robot"—that is, a machine able to identify victims within a re-created disaster scenario. These robots must accomplish tasks varying in complexity from walking a line on a flat surface up to negotiating paths through obstacles on uneven terrain and saving some well-defined victims.

The fundamental characteristics of our FPGA-based design, if compared with the normally diffuse platforms based on different kinds of microcomputers, are openness, flexibility, capability to evolve and reusability.

We built the 2011 version of the robot on a Xilinx® Spartan®-3E device, using the Digilent Nexys2 educational board. As of this writing, we are porting this version—which will compete in the next RoboCup Junior Italy in April 2012—to a Spartan-6 FPGA. The next, 2013 version is scheduled to run on a Zynq™-7000 Extensible Processing Platform device.

Working within this paradigm, we were able to design a rescue robot that evolved, year after year, from the first prototype to the current version, named Nessie 2011 (a pun on both Nexys and the Loch Ness monster, whose long neck is reminiscent of our robot's). The flexibility of the FPGA allowed a complete remodeling of the robot's architecture, following the progression of the students' knowledge, while leaving its basic physical structure substantially unaltered and maintaining the same design infrastructure.

THE CHALLENGE

The ITCS Erasmo da Rotterdam is a technical high school located in a suburb of Milan in northern Italy, with a student population that is strongly heterogeneous and, often, not so easy to coax into deepening their scientific and technological knowledge.

Starting four years ago, I decided to activate an open space, which I named the Permanent Laboratory for Didactic Robotics, where students can experiment in a different way from the standard classroom. Here, they approach the technical disciplines in an interactive environment in which pupils can negotiate some unexpected aspects of the subject matter, make choices regarding the subjects they will pursue, organize their own jobs and receive direct feedback from the results of their actions. In other words, they get to experiment in an "active learning space" based on the old and well-known model of "situated cognition" [1], where students collaborate with one another and with their instructor while pursuing a common goal with a shared understanding.

In this learning space, pupils can practice a "cognitive apprenticeship" that uses problem-solving methodologies. The teacher assumes the role of a "professional expert" who proposes specific processes involving authentic tasks and strategies, and allows students to try them independently, coaching only as needed.

Robotics was the natural choice to provide a fertile breeding ground for the convergence of different disciplines and the exchange of knowledge. We decided that the fun of taking part in the RoboCup Junior competition would provide a strong stimulus to incentivize student participation.

THE SOLUTION

I understood that to be effective, I would have to propose topics normally covered in regular classes on digital electronics and informatics, but aimed at more complex applications than those the students would be able to solve alone. Instead, they would need to work in groups or with the support of an experienced teacher who could propose appropriate models.

I knew what to build but I did not know how to build it. Everything had to be born and developed in the laboratory, with students discussing the design and seeking solutions together.

After some deliberation, I came to the conclusion that the most likely solution was one based on a flexible platform, such as an FPGA, rather than on standard microcomputers. That's because an FPGA was the only device able to provide the required characteristics, and to keep pace with the dynamic and evolutionary scope of the laboratory activities.

I chose, initially, to use an educational card based on the Spartan-3E because it could provide the necessary characteristics we were seeking—namely, openness, flexibility, ability to evolve, reusability of the hardware and richness of performance.

- **Openness,** because students must actively participate in the entire design flow, from the sensor interface to the CPU and from this to the actuators.
- **Flexibility,** because the complete architecture of the system and the nature and type of devices should not be fixed in advance, but must emerge from the research process activated within a creative context for learning.
- **Ability to evolve**, because after each RoboCup competition, students must learn the shortcomings of their work and know how to make the appropriate modifications to try to reach more advanced solutions. The system must grow in parallel with the students' expertise.

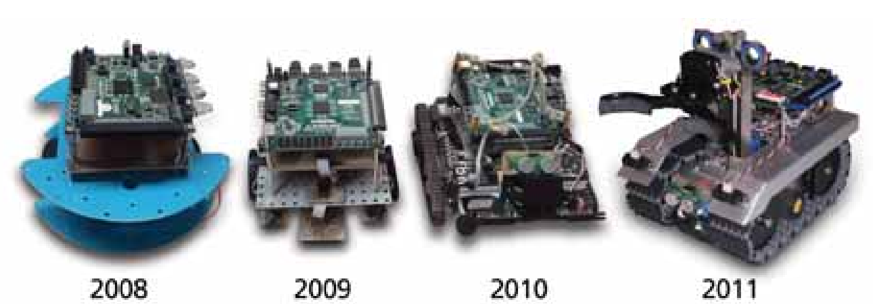


Figure 1 – Thanks to FPGA flexibility, the same platform can evolve in parallel with student understanding, reusing the same hardware and redesigning as needed. Photos show Nessie's evolution from 2008 through 2011.

- **Reusability**, in order to avoid unnecessary waste of the hardware and the school budget.
- **High performance at an affordable cost**. We had to control a large number of devices and peripherals that were not fully defined, but needed to operate with a high degree of parallelism. The CPU should be very powerful but relatively simple in its architecture and easy to interface.

NESSIE 2012: BLOCK DIAGRAM AND DESCRIPTION

We achieved our goal using a Digilent card carrying a Spartan 3E-1200 onboard, which was the common thread of the four-year project development. As you can see in Figure 1, the rescue robot the students designed showed a clear evolution from a machine that barely moved in 2008 to one that, in 2011, made us one of just 15 teams, out of 65 participants, to reach the finals.

The level of student expertise has grown from year to year, laying a foundation for further improvements that we are planning for this year's RoboCup Junior competition in April.

First, we have moved from the Spartan-3E to the Spartan-6 family, converting the bus infrastructure of the standard Processor Local Bus to the AXI4 interface. Second, we have modified some critical sensors for tracking of the reference line and redesigned the motor interface, porting a PID algorithm for automatic speed control directly into the FPGA fabric.

Figure 2 illustrates the complete block diagram of the system, as it appears in the current design. Looking at it, you can clearly see the richness of didactic topics that it enables a teacher to cover in a course on digital control systems. Equally clear is the high level of parallelism that the system can obtain in terms of performance, compared with a robotic platform built on a standard microcomputer.

Moreover, the activity in the robotics laboratory has had an interesting impact on the ordinary educational courses at our school. The FPGA has become a tool for rapid and effective implementation of the theoretical aspects of technology, sparking students' keen interest in the topics covered.

MANAGING NESSIE'S WALKING PROBLEM

Armed with the rich supply of resources available in the Spartan-6

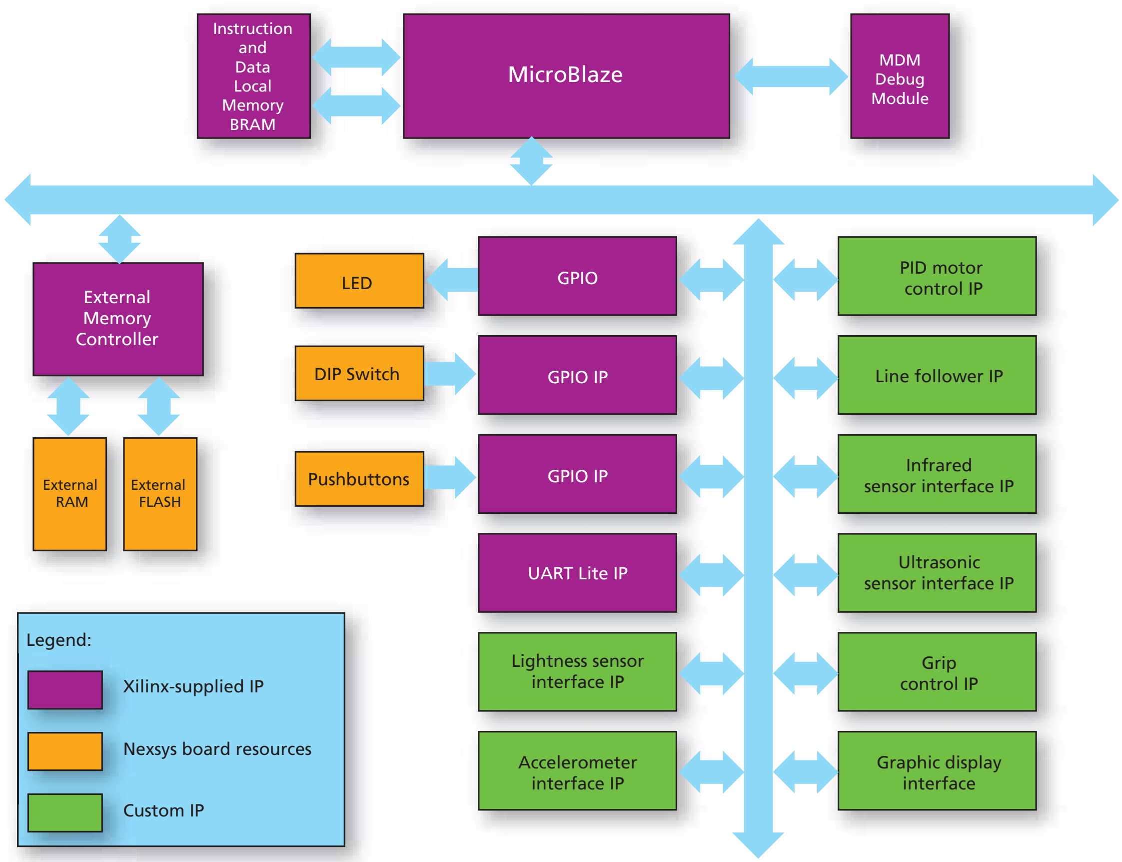


Figure 2 – This block diagram shows the current Spartan-6 reconfiguration, Nessie 2012, which will compete in the RoboCup Junior spring event.

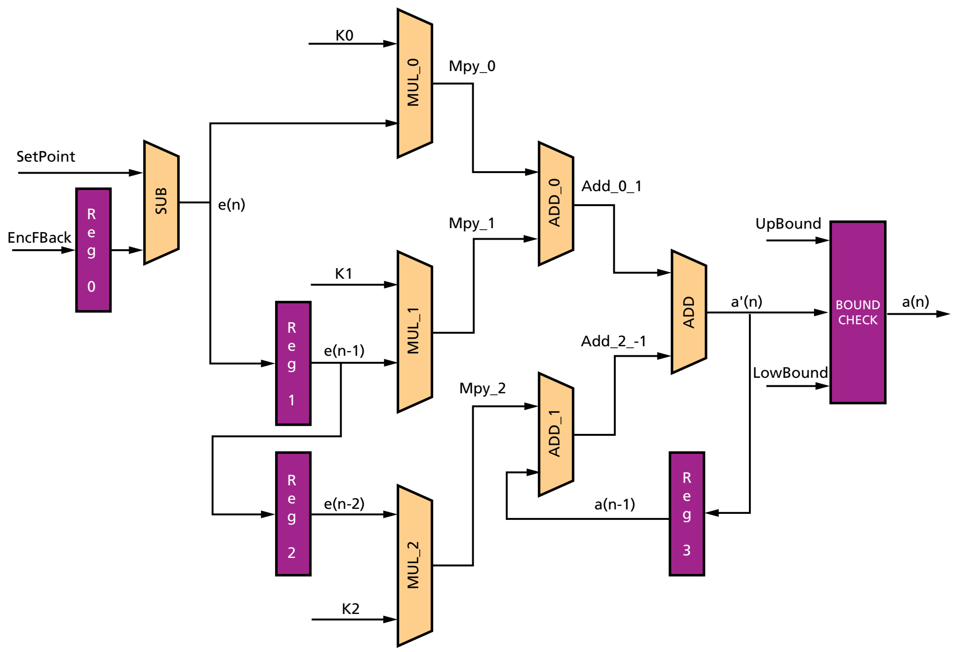


Figure 3 – RTL sketch of the PID algorithm that controls the speed of the two motors of the robot

family, I faced the problem of how to bring PID control from the analysis conducted on continuous systems to an actual realization with digital systems, giving the students the hands-on opportunity to solve the problem of digitizing a system of equations and immediately translate them in terms of the elements of digital electronic circuits.

I started from the classical PID equation:

$$a(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt}$$

And I converted it into another classic finite-difference algorithm,

$$a(n) = K_P \left[e(n) + \frac{T_C}{T_I} \sum_{j=0}^{n} e(j) + \frac{T_D}{T_C} (e(n) - e(n-1)) \right]$$

where K_P is the proportional gain, T_I and T_D the time constant of integrative and derivative actions, and T_C the sampling period. Using suggestions found in the paper "FPGA implementation of a closed-loop control system for a small-scale robot," written by W. Zhao and others, [2] I converted the model into the following very simple iterative algorithm:

$$\Delta a(n)=a(n)-a(n-1)$$

with $\Delta a(n)$ computable as

$$\Delta a(n)=K_0 \cdot e(n)+K_1 \cdot e(n-1)+K_2 \cdot e(n-2)$$

and the iteration step:

$$a(n)=a(n-1)+\Delta a(n)$$

The Ki coefficients are calculable as

$$K_0 = K_P \cdot \left(1 + \frac{T_C}{T_I} + \frac{T_D}{T_C}\right)$$

$$K_1 = -K_P \cdot \left(1 + 2\frac{T_D}{T_C}\right)$$

$$K_2 = K_P \cdot \frac{T_D}{T_C}$$

with the PID parameters empirically tunable using the Ziegler Nichols procedure.

The PID RTL model for the FPGA implementation is outlined in Figure 3. The way we implemented the complete IP for controlling how Nessie will walk into the RoboCup 2012 arena is shown in Figure 4.

What is extremely productive in this approach, from a teacher's point of view, is the linearity of the transformations and the relative transition from concept to its immediate implementation in a physical system that can be easily realized. This process encourages the students to try more experiments and further work with, and learn from, the system.

HOW TO GIVE LIGHT TO NESSIE'S EYES

As a rescue robot, Nessie's mission is to identify victims within an artificially created disaster scenario. The way the robot moves on uneven terrain, overcoming obstacles and debris and looking for the victims to be saved, is to visually follow a black line on a white background.

An efficient tracking system is crucial in determining the level of performance obtained during the RoboCup Junior competitions. A flawless execution of the guided tour is the necessary starting point if the robot is to have enough time to win the rest of the challenges, including avoiding obstacles and rubble, and rescuing victims.

The idea we came up with to solve the problem of Nessie's vision capabilities was to use a 128 x 1 linear sensor array with an internal light integration and holding circuit, namely, the Taos 1401R-LH.

The linear array is composed of a block of 128 photodiodes and an analog circuit that uses two capacitors for integration and maintenance of the charge generated by the photodiodes. The first capacitor gathers the current and the second copies it back and keeps it during the scanning and capture of the next load. The sensor, in effect, performs two operations simultaneously, superimposing the phase of integration and acquisition of a new measure with that of reading, by scanning, the charge accumulated in the previous cycle. The charge integrated in one scan cycle is transferred at the end into the holding capacitors.

Each cycle begins on the rising edge of a signal of Start Integration. At the first clock cycle, this signal throws an internal switch in order to isolate the holding capacitor and, in parallel, to delete the stored charge contained in the integration capacitor. This

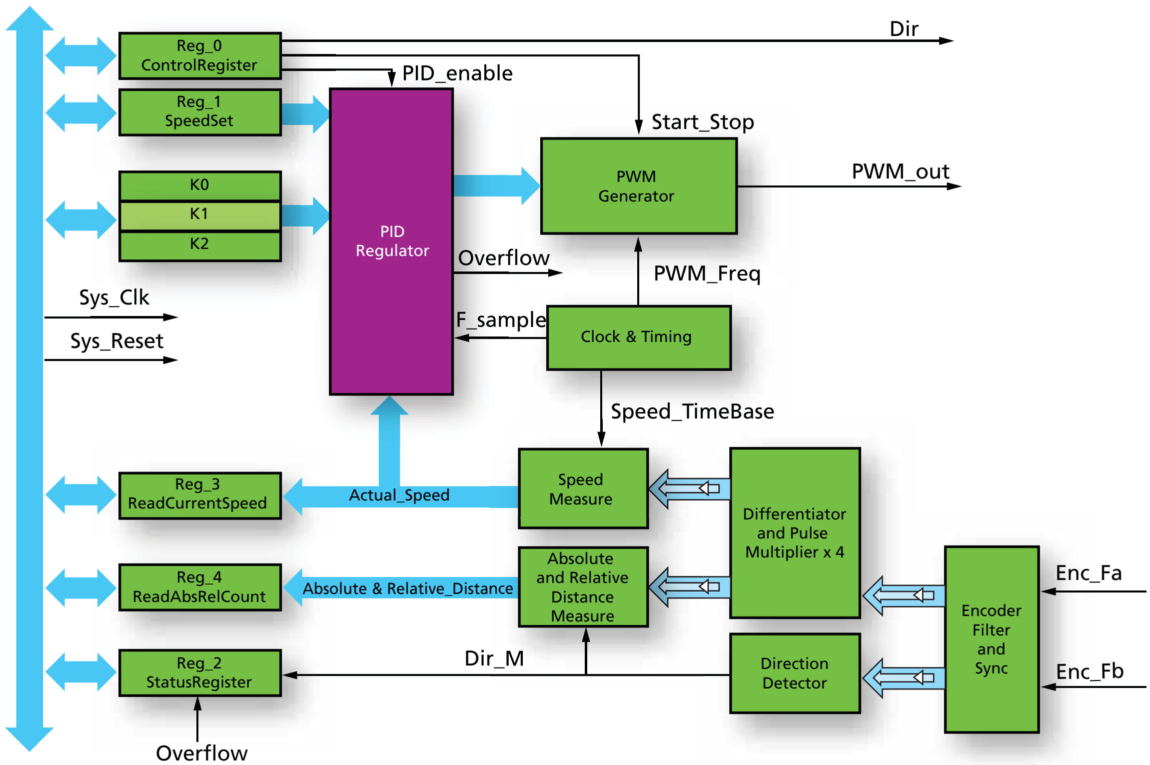


Figure 4 – The complete block diagram of the automatic motor control IP we designed for Nessie 2012

process occurs in parallel on all 128 photodiodes. The cycle continues with a new integration phase, during which the sensor will read all the 128 values of the new brightness data. Meanwhile, in parallel, on each rising edge, the charge contained in the holding capacitors is exposed to an output amplifier that allows the voltage across it to be read on an analog output pin. The voltage is readable as a shift register. In this way, it is possible to simultaneously read the voltage due to charges accumulated during previous exposure to light radiation, while the current acquisition value is accumulated in the capacitor integration.

Here again the FPGA proves its worth, freeing the CPU of the load of refreshing the accumulated current and saving the converted value to make it available for further handling. Figure 5 depicts the IP we designed for this purpose. It also illustrates the "double buffer" mechanism that internal BRAM, available on the Spartan devices, allows in order to overlap the acquisition of a new frame with the manipulation of the last one acquired. In this way we optimize the time needed for processing the images and to extract all the information for guiding the robot.

DIGITAL PROCESSING AT THE HIGH SCHOOL LEVEL

True to the didactic spirit of this robotic activity, I also used this design to introduce the students to some simple elements of digital image processing.

As can be seen from Figure 2, we added a small graphics display to the design in order to show what the Taos sensor was viewing and to understand how to process it for extracting the information necessary to guide Nessie to achieve its purpose. Figure 6 shows the raw image captured in one frame.

In working with Nessie, students must discover how to manipulate the scanned line so as to discern, between possible debris and garbage, the location of the black line and to define an algorithm for controlling the speed of the two motors to accordingly follow it. In this way the students grasp simple issues of how pixels are processed and learn the results of different mathematical operations in pursuit of that task.

In particular, to locate the black line, they will discover some fundamental operations: thresholding, salt-and-pepper filtering, edge detection and line segmentation.

The scanned line needs to be organized as a collection of segments of different levels of light and different lengths, and all these objects must be correctly ordered to identify the direction and guide the robot. This represents a second control loop, managed via software, outside the hardware PID loop, that allows an optimal movement during the task of line following, adjusting the relative speed of the two motors according to the angle between the sensor axis and the black line, and the relative dimension of the lateral white spaces.

MOTIVATIONAL TOOL

This level of system complexity has turned out to be age-appropriate for my students and has proven to be a

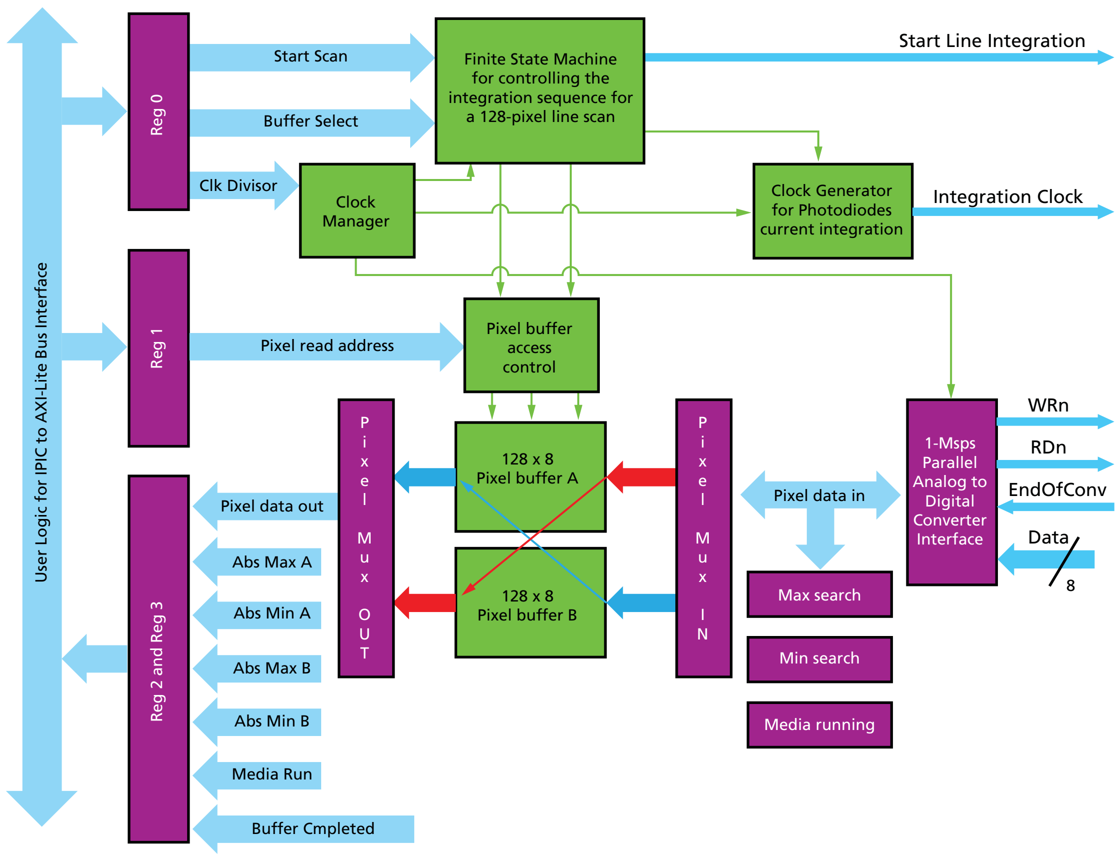


Figure 5 – Block diagram of the interface between the MicroBlaze® and the 128 x 1 Taos 1401 linear-array light sensor

good way to introduce them to higher levels of difficulty, such as two-dimensional image processing. These kinds of starting problems are more accessible and can serve to motivate the students to continue improving their technical and scientific knowledge and gaining new skills as they find solutions to more complex issues. With my long experience in teaching in high school, I can safely say that after this kind of experience, students acquire greater interest and motivation to continue their technical studies and can address both university study and the search for a job in their specific sector with greater security.

In my three years of teaching robotics using this flexible FPGA-based platform, I have seen my students attain a greater perception of the internal organization of "real-time" computer systems. They also acquire a knowledge of how to design the inner peripherals along with a greater sense of authorship over the finished product, an enhanced ability to work in a team and a better capacity to address problems and find appropriate solutions on their own.

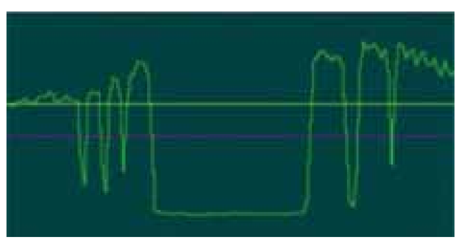


Figure 6 – Raw pixel line captured with a black line over a white background and some debris near it, before the thresholding and the salt-and-pepper filtering

Given the educational purpose of this activity, our group at the ITCS Erasmo da Rotterdam is encouraging students to use their work on Nessie as the basis for passing their final State Exam, a necessity for graduation in Italy. The exam has an individualized component in that students must discuss a personal research path. Nessie was, in fact, the inspiration for some students to achieve an electronics technician diploma, with honors.

References

1. J. S. Brown, A. Collins and P. Duguid, "Situated cognition and the culture of learning," Educational Researcher, Vol. 18, No. 1. (Jan. - Feb. 1989), http://people.ucsc.edu/~gwells/Files/Courses_Folder/ED%20261%20Papers/Situated%20Cognition.pdf

2. W. Zhao, B. H. Kim, A. C. Larson and R. M. Voyles, "FPGA implementation of a closed-loop control system for small-scale robot," Proceedings from 12th International Conference on Advanced Robotics (ICAR 05), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.81.8719