Fast and flexible

Field Programmable Gate Arrays conquer Machine Vision

FPGA´s are proven tools in Electronic Design Automation (EDA) – their parallel architecture cuts them out for the emulation of complex digital circuits.
A more widespread use in imaging is prevented by the present programming tools, originally developed for EDA. Silicon Software, Mannheim, Germany, developed two novel tools which overcome this hurdle by also allowing the non-expert to take advantage of FPGA´s, using object-oriented programming or a graphical user interface: The combination of C++ and the Place&Route tools of the FPGA´s manufacturer provide for an advanced library concept effectively supporting FPGA-based algorithms in machine vision and other demanding image processing applications.
Rapid Prototyping is enabled by Visual Applets Studio, an easy to learn drag and drop approach.
Both software packages run on the microEnable- and microUSB- line of Silicon Software´s CameraLink framegrabbers.

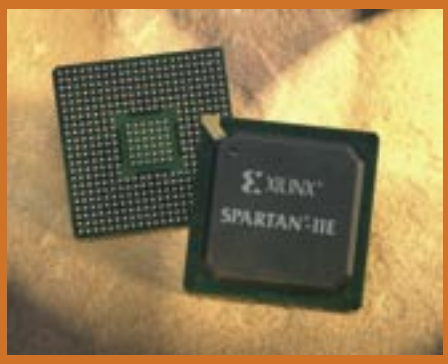

Fig. 1: Xilinx Spartan-IIE FPGA, Source: Xilinx

Fig. 2: microEnable-III frame grabber

What can I do with my frame grabbers IQ?

Since a couple of years the term "IIBP" (Intelligent Image Processing Board) circulates in the industry as a separation to simple framegrabbers. "Intelligence" here stands for the existence of a vision processor on the board, in principal allowing for image pre-processing functionality; however up to now frequently not applied due to difficult and complex programming.

In consequence, the complete application is running on the PC. On the other hand, even the high performance of actual CPU´s reaches its limits taking the more and more demanding machine vision algorithms (often to be computed in real time) into account. Given this, there is a definite need to take advantage of the computing power implemented on the IIBP.

Depending on the product philosophy some companies offer predefined, fixed pre-processing algorithms, others provide code development tools to access and use the board´s intelligence. Clearly, the clue for an efficient usage lies in the simplicity of the programming features delivered by the manufacturer.

A revolutionary way to combine hard- and software

Field Programmable Gate Arrays (FPGA´s) distinguish themselves from other on-board intelligence (DSP, multimedia chips) by their ability to process very large data sets in parallel, and by their possibility to modify the hardware structure – processor functionality can be changed by software. Thus the modern technology of reprogrammable FPGA´s combines the high throughput realized by hardware with the flexibility of software. In essence, both the enduser and the manufacturer benefit from this approach, saving time and money – hardware modifications can be achieved in hours rather than in months, and individual adaptations can be implemented later on at any time with no additional expenses.

microEnable – a reprogrammable IIBP

The concept of Silicon Software´s microEnable line is based on the combination of an up-to-date image transfer design between camera and computer (CameraLink), and a high performance and flexible FPGA. Two cameras of up to 85 MHz can be connected. 96 Mbyte of RAM serve as on-board buffer memory, the 64bit/66MHz PCI interface fits in the bus slot of the PC. The grabber is initialised by means of an especially developed GUI. Loading the appropriate applets, the FPGA can be reprogrammed in a fraction of a second to optimally support linescan or matrix cameras. Furthermore, the readout sequence of the camera sensor, shading correction (1D and 2D), fixed pattern noise reduction, and Bayer filter conversion are other examples of applets readily made available to the user.
Depending on the performance of the FPGA implemented other applets, e.g.

Fig. 3: VisualApplets Studio surface

Fig. 4: Custom specific applets in microDisplay

to support pre-processing functionality (digital filters, look up tables, etc.) or for the configuration of dedicated sensors are offered.

The "classical" way to program FPGA´s

The most common tool to program FPGA´s is the hardware description language VHDL language (VHSIC Hardware Definition Language, where VHSIC stands for "Very High Speed Integrated Circuit"). The typical difficulty for CPU-programmers, used to a sequential sequence with subroutines, is due to the fact that all functions of an algorithm usually are implemented in parallel; at runtime these are then controlled by a dataflow scheme. When this is not taken into account, side effects of the hardware functionality can occur, preventing the expected behaviour of the intended functionality.

Therefore, one hurdle in accepting VHDL is not its syntactic structure, but rather the semantics – the way to implement algorithms and the associated code.

A second barrier lies in the relatively low level of abstraction (from the viewpoint of a CPU-programmer). Examples are, e.g., the problem of very time-consuming coding, the existence of side effects, tedious debugging, missing hardware independence on the processor- and system level, as well as the low reusability of the code generated.

These problems – high performance processor vs. obsolete software tools – challenged Silicon Software to develop an object-oriented high level language based on C++, striving for a much broader acceptance of FPGA´s in the image processing community.
Since five years this tool is internally used for the project- and application-development.

C++ – object orientation also for Gate Arrays

This coding-tool delivers a modern technique, allowing both the FPGA- and the CPU- programmer to easily access the power of an FPGA. A straightforward and intuitive library concept forms the basis, only requiring a C++ – compiler and the Place & Route tools of the FPGA´s manufacturer.

Compared with VHDL, the usage of the standard C++ – language results in a number of substantial advantages. For example, hardware modules can be overloaded and thus be adapted to the users needs; arbitrary levels of abstraction can be realized by implementing different runtime designs(???). Last not least, simulation and synthesis are made in one custing, so that no other, separate tools are necessary.

Important criteria during the planning stage of the of this programming tool lay in placing emphasis on maximum performance of the underlying hardware and extremely short design cycles of the software. Apart from that, the possibility of adapting the tool to a multitude of hardware environments played an important role. This latter issue makes applications to a very high degree independent of the FPGA used.

Drag & Drop – even a simpler solution for Rapid Prototyping

Often only relatively small changes of the framegrabbers functionally make the difference in an imaging project – may it be the implementation of additional digital filters or the possibility to later on modify the system design. Moreover, the enduser often lacks a deep know-how in C++ – programming, or the time constraints are very tight. In other words, here the keyword is "rapid prototyping".

Silicon Software took this challenge by developing VisualApplets Studio, featuring a graphical user interface which can be intuitively operated. The intention was to also enable non-programmers to work out an image processing project by just using building blocks presented on a drag & drop scheme.

VisualApplets Studio comes with pre-defined image processing modules, which can be arranged and parametrised within minutes to form complete imaging applications. In addition, user-specific modules can be integrated, which were written by the tool described above. As a result, this mix & match of own and Silicon Software´s building blocks perfectly addresses the needs of the machine vision community, without compromising the performance and power of the FPGA.

Prototypes of VisualApplets Studio as well as the C++ – programming tool were exhibited at VISION in Stuttgart, they were very well received. This positive feedback clearly demonstrates that Silicon Software´s approach to introduce FPGA´s to the imaging community by making their usage as simple and effective as possible will substantially widen their acceptance.

THE AUTHORS

Dr. Ralf Lay,
Executive manager
SILICON**SOFTWARE**

Dr. Helmut Schwarz
Sales Manager,
Sales office München
Tel. 0621/789507 0
info@silicon-software.de,
schwarz@silicon-software.de,
www.silicon-software.de