Imageprocessing with FPGAs

by
Matthias Schaffland
Feith Sensor to Image GmbH, Schongau
mas@feith.de

Imageprocessing in automation and embedded devices is a growing market. To enhance the possibilities of small imaging systems, hardware-based image-processing with reconfigurable devices like FPGAs becomes more important. This presentation should give you a basic-understanding of the technology of FPGAs and an idea, what is possible with FPGAs.

The beginning is an overview over the design and way of function of an FPGA.

Next we have a look at the different fields of applications in vision-systems, where FPGAs can be used. In addition some typical applications are demonstrated, where FPGAs fit in.

How to program this devices? The typical design-flow is discussed in a further part of this presentation.

If we deal with image processing and automation, real-time processing is almost a topic. How FPGAs fulfil real-time requirements is also discussed.

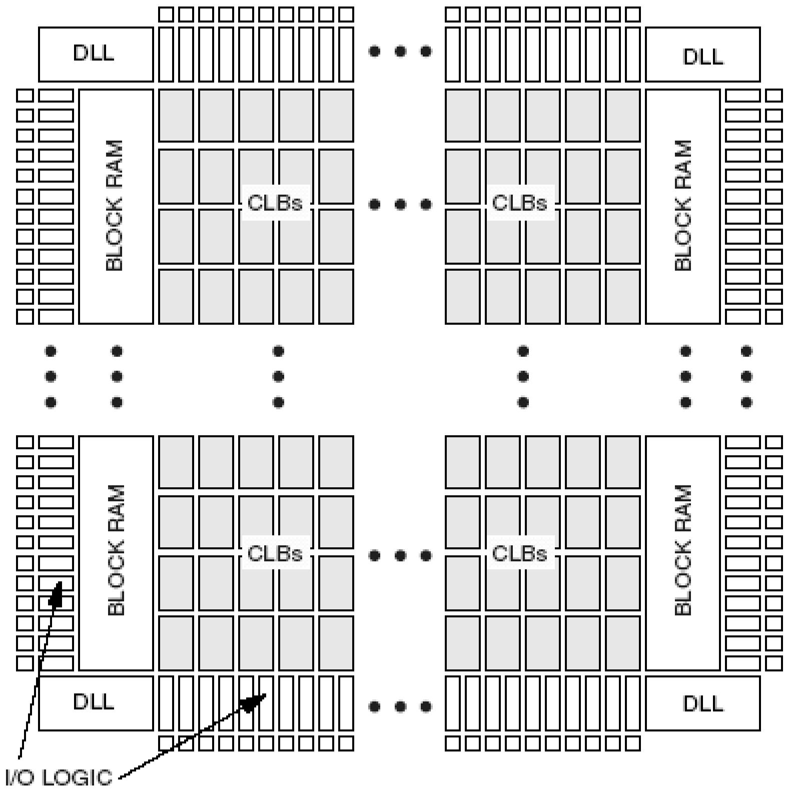
Before a example of successful vision-project is demonstrated, the advantages and disadvantages of FPGAs are summed up, espacially with vision applications.

Last of all to give an idea of the practical using of FPGA-based machine vision components, an application is demonstrated, where a check of pills in blisters is realized.

Design and workings of FPGAs

In many cases the number of gates is mentioned as the criterion to classify FPGAs. These gates are available from some 10000 to several millions. The number of gates is strictly speaking a quite weak feature to rate the performance of an FPGA. In addition the basic-element of FPGA is not a single gate, but a so called logic cell, which number is very smaller than the fictive number of gates. The number of gates is more a feature of the marketing-people and is an estimation how many gates are necessary to build a number of logic cells.

An FPGA is also not a pure grave of gates or logic cells, an FPGA often has additional ready to use components, which are very useful for several applications. Here the structure of an FPGA (Xilinx Spartan2E):



Structure of an FPGA

Beside the logic cells (CLBs) there can be found I/O-Blocks, internal memory (Block RAM), additional function blocks like multiplier or dlls (clock management)

Programming is done with tools of the vendor or third party tools. These translate the function, which is expressed in a hardware description language, into a chip-specific form and output a programming file.

To descripe the function or the behaviour of an FPGA there exist two common hardware description languages, VHDL in Europe and Verilog in America.
Here an example of a counter in VHDL:

```
entity counter is
port (                                      --port definition
        SYS_CLK : in std_logic;             --clock signal (extern)
        RESET_IN : in std_logic             --reset signal (extern)
);
end counter;

architecture Behavioral of counter is                       --definition of behaviour
SIGNAL counter : STD_LOGIC_VECTOR(31 DOWNTO 0);          --32bit signal
(variable)
begin
        test_counter : process(RESET_IN, SYS_CLK)
         begin
                if RESET_IN = '1' THEN                      -- if reset, set counter to 0
                counter <= x"00";
                ELSIF SYS_CLK 'event and SYS_CLK = '1' then    -- if rising edge
                                                               -- of clock
                counter <= counter + '1';                      -- increment counter
                END IF;
         end process;
End Behavioral;
```

Fields of application in several vision systems

Generally, there are three main scenarios:

a) Classic vision system

An FPGA can be present in all available components, inside of a camera, on the framegrabber or as a seperate FPGA-board as additional card for a PC. It is to check, whether the used FPGA can be used from the user too. Many cameras or framegrabbers have an FPGA-component, but it is not self-evident, that the vendor of the component supports access to the chip. Components of the Feith company explicitly are designed to give the user the possibility to enhance the application with FPGA resources.
For special applications there exist some general purpose dedicated FPGA-board, which can be used for image-processing too.

b) Compact vision-systems

The possibilities to use FPGAs is dependent on the architecture of the vision-system. Either the system is PC-based, then it is a classic vision system even though it is smaller. On other systems, like the Cleopatra-board, framegrabber, FPGA and processor are combined on the pcb. Just for space-saving but also powerful systems, this design is suitable.

c) as further development of compact vision systems we can see intelligent or smart cameras, which also have the imager inside of the device. These components can offer additional FPGA-resources to enhance the power of this limited systems. (e.g. CANCam)

Examples of application

Now some examples of typical jobs, that can be done inside of an FPGA:

- Realising lookup-tables e.g. to realize gamma-correction or a 10 to 8 bit datareduction
- Windowing/datareduction to given AOI
- Bayer-interpolation in case of color sensors
- Binarizing
- Binning
- Histogramm calculation
- Filter
- Edge detection
- AOI-Tracking in case of CMOS-Sensors with windowing feature (including sensor control)

Design Flow

After these examples, the steps to program successfully an FPGA is discussed. Following the typical and proven design flow:

- Development of the algorithmics PC-based, e.g. with C
- Guideline for the implementation
  - no C++ (object-oriented)
  - Level of abstraction of VHDL is like assembler
  - CPUs are working in sequences, FPGAs work parallel
  - Definition of process-order, or identification of dependent processes
- Soft test (in software)
- Port to FPGA (e.g. with VHDL)
- Live test
- Future :Tools to convert C-Code to VHDL/Verilog

Realtime capabilities

Realtime is a popular topic to discuss in the technology of automation and image processing. But what means realtime?

Realtime just means defined (reproducible) response time. This does not automatically imply extremely fast, the response time could be quite long, but the time is reliable. Whether this time is sufficient for the application is an other subject.
Now, what is the response time of an FPGA? Of course, one can't say this in general, but there exist some factors, which have an effect on the response time and which can be calculated exactly.

- FPGA-clock (a chip with 100MHz clock normally works faster than one with a 50MHz clock)
- Algorithmics (essential factor is the implemented algorithm. Like in software the calculation time growths up with complexity of the algorithm. But this calculation time can be calculated or measured exactly.)
- Access to memory (a memory access always takes time, because there are specific access sequences and waitstates, dependent on the used memory technology. But it is also possible to do a worst-case calculation, so this effect is defined too)

There are some other parameter, but the message is, that they are all determinable. So it is possible to get a defined maximal response-time for an FPGA-design, which is normally dramatically shorter, than the time of software-based systems.

Advantages and Disadvantages of using FPGAs

+ very high processing speed (Hardware!)
+ online-processing possible, i.e. if an image is available the image processing results are available too.
+ costs for components can be reduced (good in case of series production)
(compact vision-system with FPGA has a better price, than a powerful PC with framegrabber)

- complex image processing is costly to realize
- new way of programming (sequential vs. parallel)
- in case of single products/applications often too expensive
- no library available (technology is very dependent on FPGA-type and hardware-platform)

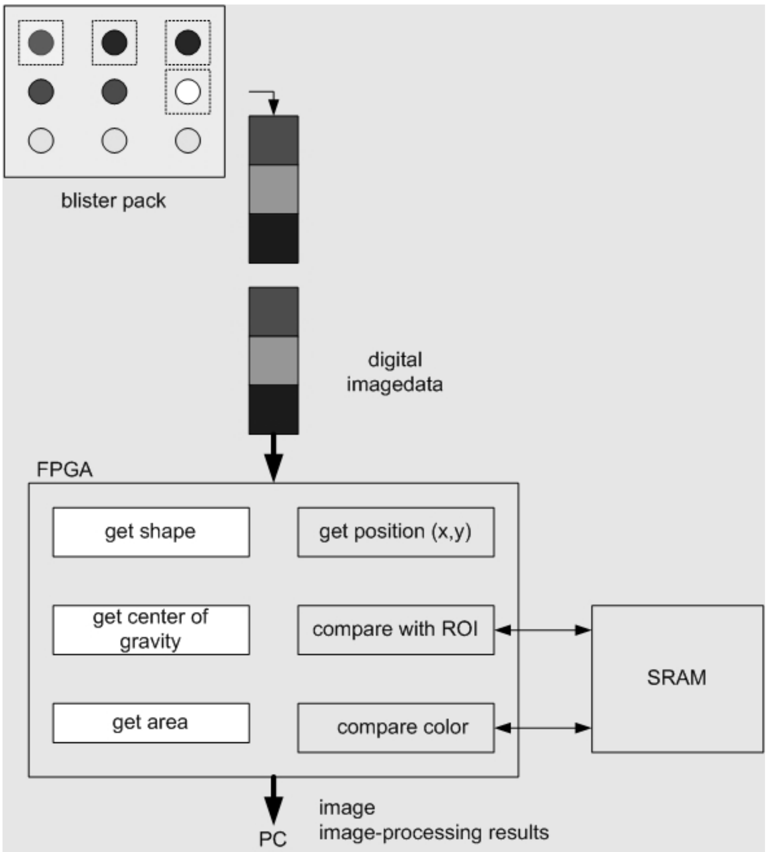
Success Story – Check of pills in blister-pack

Task:
First, pill in its blister-pack are to be tested for presence. Therefore the measured positions of the pills are to be compared with given positions, stored in memory. In addition the calculated color is compared with predefined colors, to detect mixing up of pills. At last the circumference of the pills are checked to detect damages, because normally a damaged pill has an other shape than a non-damaged one.

Constraints:
Previous system works PC-software-based. With new system the processing time per pack is to be reduced. But the performance category of the PC cannot be increased, due to limitations in space and tolerable waste-heat.

Method of resolution:
Of course one tried to optimize the used image processing algorithmic to achieve mentioned goals. This leads to some improvements, but not enough. Therefore is was checked whether it is possible to do some imaging jobs on the framegrabber, to reduce the PC’s load.

Solution:
Do the color-classifing on the framegrabber. For this in predefined regions of interest it is checked on objects with certain color. In addition the object’s shape and the center of gravity is calculated.
If this values are in the normal range, the PC has not do anything. Only in the case of failures the PC must do a second check to verify the result.



Dataflow

Result:
- Speed is quadruplicated.
- PC is only active in problematic cases (center of gravity not in expected range, color mismatch, shape is not like the reference

Conclusion

It was show, that FPGAs are an interesting technology for vision applications. It is recommended that interested parties contact someone, who has broad experience in the field of technology to find the best solution.