

# Tech-Tip

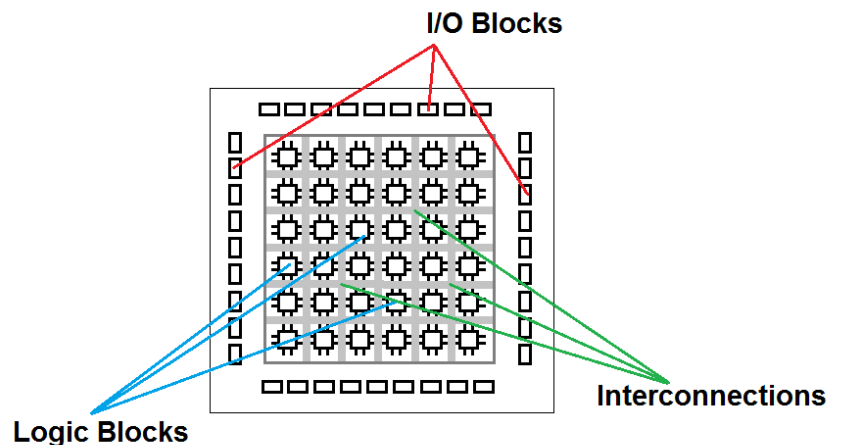


Invaluable Information from the experts...

## Introduction to FPGA acceleration

### What is an FPGA?

An FPGA (Field Programmable Gate Array) is an integrated circuit consisting of a collection of “logic blocks”, “I/O Cells” and “interconnection resources”; this allows the chip to be reconfigured to connect the inputs & outputs (I/O) and logic blocks together in many different ways. Each logic block has traditionally the ability to do a simple logic operation, such as AND or XOR, and generally contains some degree of memory, be it a simple Flip-flop or a more complex block of memory. The



logic blocks have evolved to be more logic function blocks using lookup tables within the blocks to switch the current function; to perform tasks such arithmetic operations. This view of FPGA is only part of what a modern FPGA is; current FPGAs are starting to cross the line of FPGA and on-chip system. But the true power of the FPGA is still the core attribute of having a number of logic blocks that can be connected together to perform massively parallel, real time processing.

### Where are FPGAs in machine vision?

FPGAs are rarely mentioned but they do exist in a lot of equipment that is used within machine vision. They are often found as part of a cameras back end, used for converting the data coming from the sensor to more meaningful data to be transmitted from the camera. As an example of this we can look to the new CVC GigE Cameras from Stemmer Imaging, they use a Sony FCB module as the camera (this in its self-will contain micro controllers and potentially an FPGA), and the Sony module interface with FPGA takes the LVDS signal from the camera and converts it to a data stream to be transmitted using Gigabit Ethernet. As well as implementing the data conversion the FPGA core runs a memory controller used for implementing the data resend mechanism.

The FPGA performs the GigE packet composer and also runs an integrated virtual CPU, in this case used for non-time critical processes, such as the camera control.



WWW.STEMMER-IMAGING.COM • IMAGING IS OUR PASSION

#### GERMANY

**STEMMER IMAGING GmbH**  
Gutenbergstr. 9-13  
82178 Puchheim  
Phone: +49 89 80902-0  
Fax: +49 89 80902-116  
info@stemmer-imaging.de

#### UNITED KINGDOM

**STEMMER IMAGING Ltd.**  
The Old Barn, Grange Court  
Tongham, Surrey GU10 1DW  
Phone: +44 1252 780000  
Fax: +44 1252 780001  
info@stemmer-imaging.co.uk

#### FRANCE

**STEMMER IMAGING S.A.S.**  
23 bis, rue Edouard Nieuport  
92150 Suresnes  
Phone: +33 1 45069560  
Fax: +33 1 40991188  
info@stemmer-imaging.fr

#### SWITZERLAND

**STEMMER IMAGING AG**  
Rietbrunnen 48  
8808 Pfäffikon SZ  
Phone: +41 55 4159090  
Fax: +41 55 4159091  
info@stemmer-imaging.ch

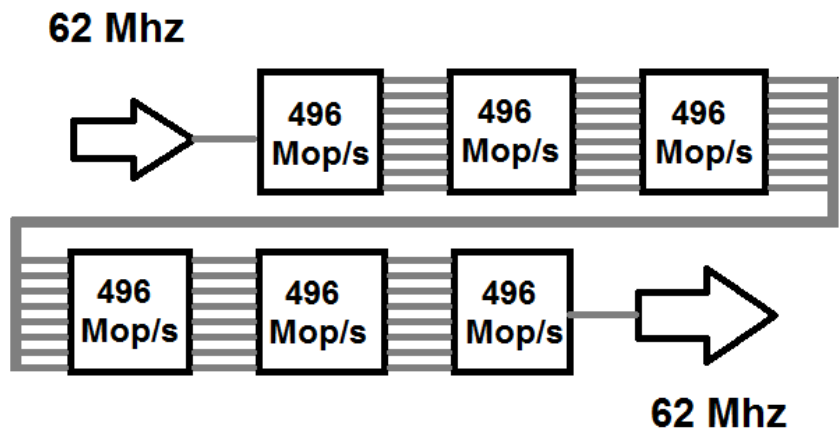
A second area where FPGAs are heavily used is in frame grabbers, again this is often for data conversion or image enhancement, for example the Dalsa Xcelera frame grabbers generally have several different firmwares that can be uploaded to perform different function, be it to convert a grey scale Bayer pattern to RGB data, or for image enhancement such as flat field correction.

A more advanced use of FPGA's in frame grabbers can be seen on the Silicon Software ME4 boards, the architecture here uses two FPGAs, one for controlling data acquisition and transfer, virtual serial ports and communication with the camera, and a second FPGA, programmable by the user via a graphical design tool, to perform time critical, high bandwidth and more complex operations such as sobel filters, simple JPEG compression, high speed data conversion, blob location or laser triangulation.

### Speed comparison!

It is difficult to compare CPU performance against FPGAs. CPU benchmarking is usually specified as how long functions take to execute, and how fast they run whereas FPGA performance is based on data throughput. Modern CPUs run in the order of GHz, as an example a modern CPU might have four cores that run at 2.4 GHz each. Comparatively, speed wise an FPGA is very slow running at just 62 MHz. This looks like a startling difference and might make one wonder, how an FPGA can be fast enough to be of any worth. This is where the parallel processing nature of an FPGA comes in.

Rather than being concerned with how fast an FPGA can run, we are interested in the data rate it can handle. If we take a look at the above image as an example flow of data, we can see how the speed is only part of the equation. Data comes in to the FPGA, it then gets split up to be parallel processed, the data then feeds into different functions (set up using the logic blocks), in the example we have 6 of these functions. The FPGA then outputs the processed data. This all happens in real time (i.e. a predefined amount of time with only a tiny amount of jitter).



If we look at that in more detail, data comes in at the rate the camera output, (62 MHz)

This is then split in to 8 parallel processes, so now the data rate is 496MOPS (Million operations per second)

This then passes through the different stages of the processing. Now processing 2480 MOPS

The parallelism is then removed and output to the memory at a steady rate of 62 MHz



For vision applications where this processing happens on the frame grabber it is all happening in real time, with no overhead on the CPU, leaving the CPU to perform other functions.

### What can and can't an FPGA do well?

FPGAs can perform some function very efficiently and very fast but there are limitations to what an FPGA can do (well). The sweet spot for FPGA processing is in-line processing, functions that can be performed on a stream of data without knowing what the rest of the data looks like. This includes Fourier transforms and matrix filters, be it high pass, low pass, Sobel, erode, dilatation. Anything that can be done on a small section of data, this can be very useful for managing high data rates, if for example a threshold needs to be applied to an image, this is very easily done by the FPGA. And has the advantage of presenting the CPU with far less data to processes.

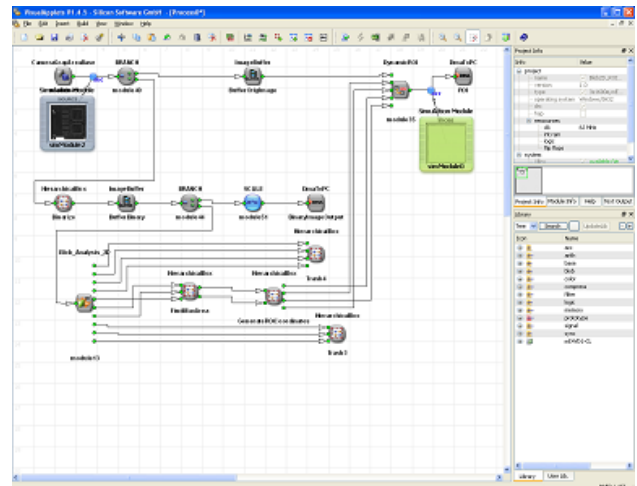
Where FPGAs are weak is when they have to deal with the whole image, FPGAs generally have little memory compared to CPU's, they don't have banks of registers to use for addressing, and they can't juggle data around. The implication is that running algorithms that require random access to the data are not practical; also functions where iterative operations have to be performed again are not ideal for FPGA's.

This is not to say they can't technically be implemented, if we take the fact that an FPGA can have a virtual CPU created on it, we can at this point perform the same kind of functions used on a CPU based system, but this requires extra hardware for the FPGA to access.

### FPGA Programming; help is at hand

With the ever increasing data rate achieved by modern cameras, turning to FPGAs for the extra processing power needed can solve a host of problems. To overcome the complexity of programming FPGAs, products like Silicon Software's Visual Applets, a graphical interface for the programming of the FPGA on the ME4 frame grabbers, can be used Simplifying the world of VHDL to a GUI based programming environment.

For the more adventurous OEM users Stemmer Imaging offer a GigE Vision compliant IP-core for Xilinx and Altera FPGAs enabling quick entry to FPGA based technologies and GigE Interface.



### A list of key differences between FPGA, DSP, GPU and CPU:



FPGAs and CPUs are not the only method of processing. It is now very common to use the graphics processor (GPU) to offload some processing and this can be a very cost effective solution. Digital Signal Processors (DSP) another method for processing data in real time.

Below is a broad overview of the differences between these processing methods

FPGA processing	CPU Processing	GPU processing	DSP processing
Massive parallel processing	Sequential processing	Limited parallel processing	Limited parallel processing
Implementation is hardware	Implementation is software	Implementation is software (Imaging Lib's)	Implementation is software - proprietary
No or limited legacy due to "pipelined" process	Many I/O operations, load time etc.	Many I/O operations, load time etc.	Many I/O operations, load time etc.
Extremely fast execution Real – Time processing	Execution non-deterministic, varies with many dependencies	Execution fast Real – Time processing	Execution fast in certain areas, limited in others
Limited space	Nearly unlimited space for operations (programs)	Dependent on GPU chip (Limited)	
Not good at certain operations (floating point etc.)	Extremely versatile	Extremely strong in certain operations	Extremely strong in certain operations
Complicated to program (VHDL etc.)	With languages such as C++ very easy to program	With imaging libraries very easy to program	Not easy to program, proprietary development environment
Test / Debug difficult	Wide range of tools for debug	Some tools for debug available	Some tools for debug available
Need to interface to software	No need for special interfacing	No need for special interfacing	Needs to interface to software

## Conclusion

FPGAs used for data conversion is wide-spread and generally unseen by the user but when they are brought to the forefront of processing, they have the ability to offload processing power form the CPU and can enable extremely high bandwidths to be managed; they are not suitable for all applications and not needed for many. But what they offer can make them an invaluable tool in machine vision.

*Mr William Clifford, STEMMER IMAGING*