Acromag, Incorporated
30765 S Wixom Rd, PO Box 437, Wixom, MI 48393-7037 USA
Tel: 248-295-0310 Fax: 248-624-9234 http://www.acromag.com

A Primer on FPGA-based DSP Applications

Trends, Options, Considerations, and Tools for Using Re-configurable FPGA Platforms as an Alternative to Dedicated DSP Hardware

A White Paper Study
May 14, 2008

FPGAs Arrive to Provide Options for DSP Applications

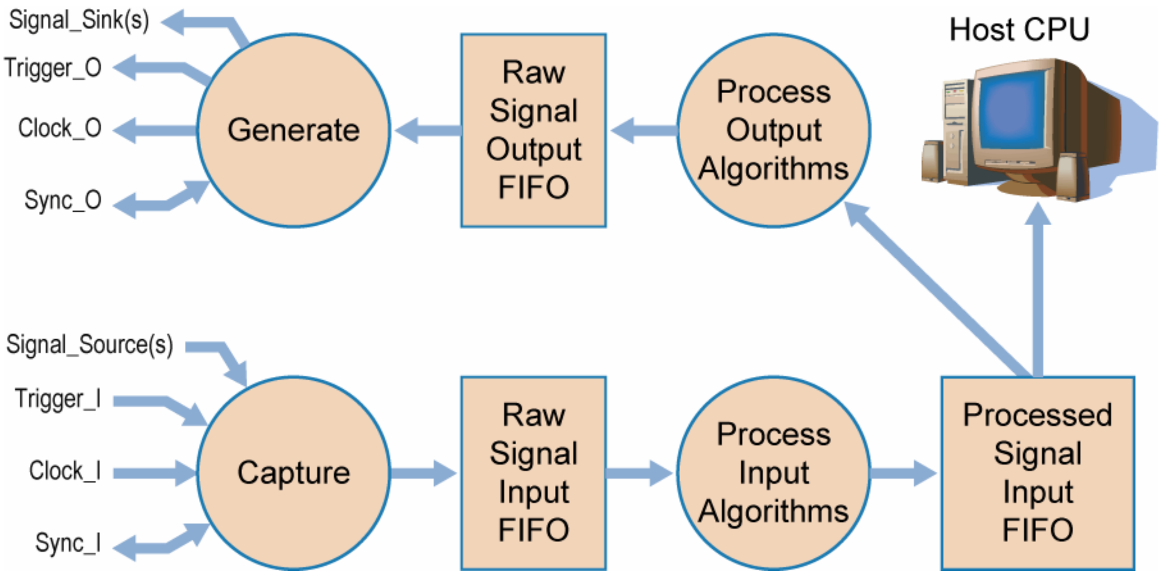
The development time, cost, and ease with which DSP solutions can be deployed has improved considerably since the introduction of high performance FPGA components from companies such as Xilinx® and Altera®. Modern FPGA devices now offer extensive support for DSP applications. In this paper, we will discuss these advances, current trends for implementing DSP applications, and the benefits of using FPGAs for DSP.

Although a variety of FPGA devices now offer DSP support, this article will limit discussion to Xilinx Virtex®-4 and Virtex-5 FPGAs for simplicity. Many of the technologies, tools, and practices mentioned here are likely applicable to FPGAs with specialized DSP blocks from other vendors.

For proper processing of signal information, there are control, synchronization, signal data capture, signal data management, and signal data exchange or transmission activities which make the overall task formidable. The primary reason solutions were so expensive to design, slow to develop and prove, and difficult to re-deploy was that the solutions were fixed in hardware.

We see the task of digital signal processing in Figure 1 as a simplified flow of events found in many common applications.

Figure 1: *DSP flow of events*



Raw signal acquisition and the generation of signals is complicated by the control logic needed to ensure proper timing and sequencing with external devices. Internal processing of signal data, packetizing/de-packetizing, and FIFO storage management are tasks internal to the FPGA module. These tasks also require exacting attention to timing and sequencing.

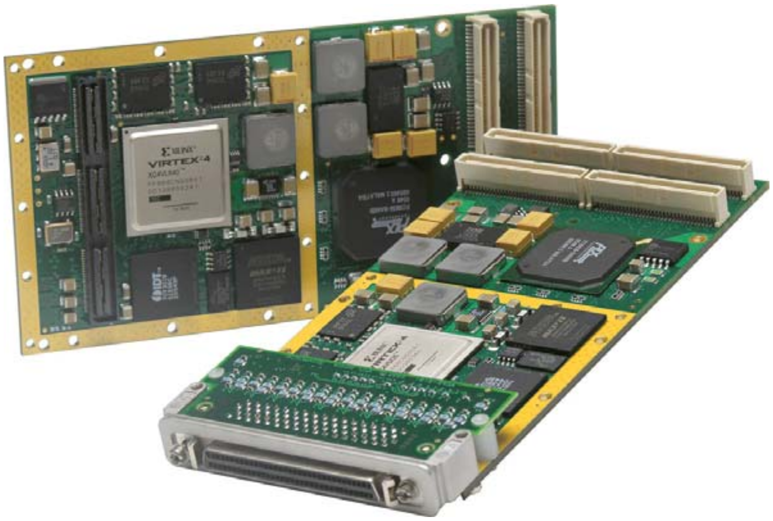
The exchange of information with the Host CPU is often implemented using techniques including–but not limited–to DMA transfers, dual-ported memory exchanges with semaphore synchronization, interrupts, and on occasion time-based or polling techniques. The software system architect makes these decisions based upon performance and resource constraints.

DSP Tasks Migrate from Dedicated Hardware to Re-configurable FPGA Platforms

For years, DSP modules were designed and developed for specific-purpose applications and based upon discrete implementations with fixed hardware interfaces both to the outside “sampled world” and with the internal “algorithmic processing world.” With the advent of FPGA technology, and more specifically the ability of DSP engineers to program the FPGA, a gradual increase in overall flexibility of the DSP solution became a reality. Today, with the latest Xilinx Virtex-4 and Virtex-5 FPGA devices on popular PMC modules, design engineers enjoy great flexibility for the handling of both the “sampled world” and the “algorithmic processing world.”. Significant characteristics of this evolution from discrete hardware DSP to programmable FPGA DSP solutions include the following transitions:

1. From defined inflexible signal capture and synchronization methodologies to freely definable and re-definable interfaces as necessary
2. From a defined data exchange methodology to a re-definable data exchange
3. From custom to off-the-shelf hardware
4. From serial logic processing to parallel logic processing
5. From the highest price per feature to the lowest
6. From defined data/sample rates to configurable rates (within hardware limitation)
7. From requiring completely defined applications to modifiable applications to adapt as requirements change
8. From debugging at the hardware level to the increased use of software debugging tools
9. From relatively inflexible methodologies to increased flexibility and no-risk (aside from lost time) trials.
10. From extensive self-design requirements to frequent use of available IP cores (free and for-sale)
11. From a fixed I/O interface to a replaceable and re-definable I/O interface
12. From long development cycles to much shorter timeframes
13. From high-cost deployment to low-cost deployment

Figure 2: *PMC modules with DSP-capable FPGA devices*

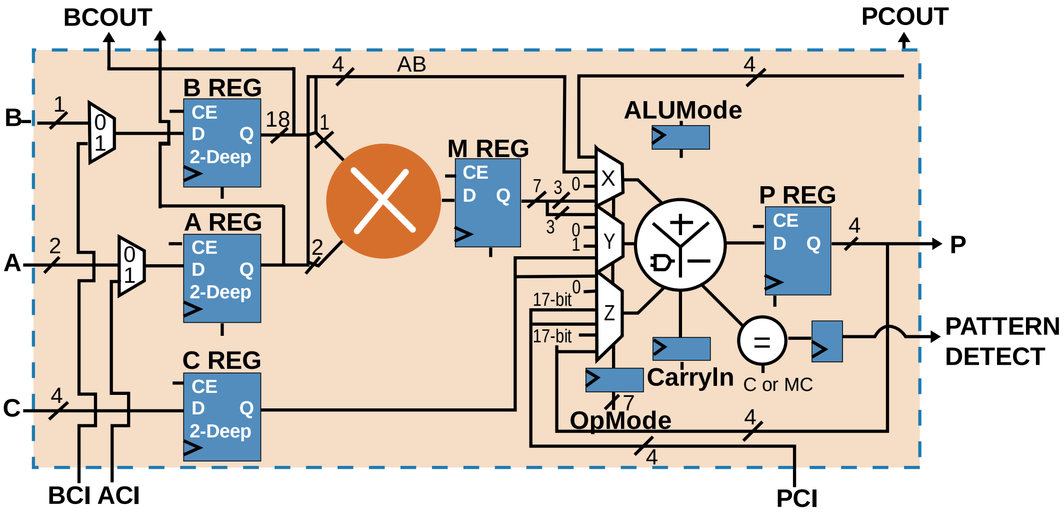


DSP Support on the FPGA

Acromag has chosen Xilinx FPGAs for their line of PMC Modules to provide DSP support for intense signal processing applications. Xilinx, in its product evolution to the Virtex-4 and Virtex-5 series, has achieved some major technology breakthroughs which ultimately result in true DSP performance platforms. To support digital signal algorithmic processing, a sophisticated level of high-capability building blocks known as XtremeDSP™ DSP48 slices on Virtex-4 FPGAs and XtremeDSP DSP48E slices on Virtex-5 FPGAs are made available. These DSP slices enable very high clock rates and very high throughput with multipliers, multiplier-accumulators, multiplier adder/subtractors, three input 48-bit adders, barrel shifters, multiplexers, counters, cascading logic support, and carry functions. The Virtex-5 FPGA’s DSP48E slice improves on the DSP48 slice with a larger 25-bit by 18-bit two’s complement multiplier for full 48-bit results with greater dynamic range and higher precision floating point math. Pattern detection automates identification of convergent rounding. An independent 48-bit C input permits three-operand addition. Additionally, the ability to support cascading results utilizing the new 48-bit Pbus eliminates external and time-consuming logic.

Figures 3a and 3b show simplified block diagrams of XtremeDSP DSP48 and DSP48E slices as found on the Xilinx Virtex-4 and Virtex-5 FPGAs, respectively. These FPGA devices are deployed on Acromag’s PMC-LX/SX and PMC-VLX/VSX Modules.

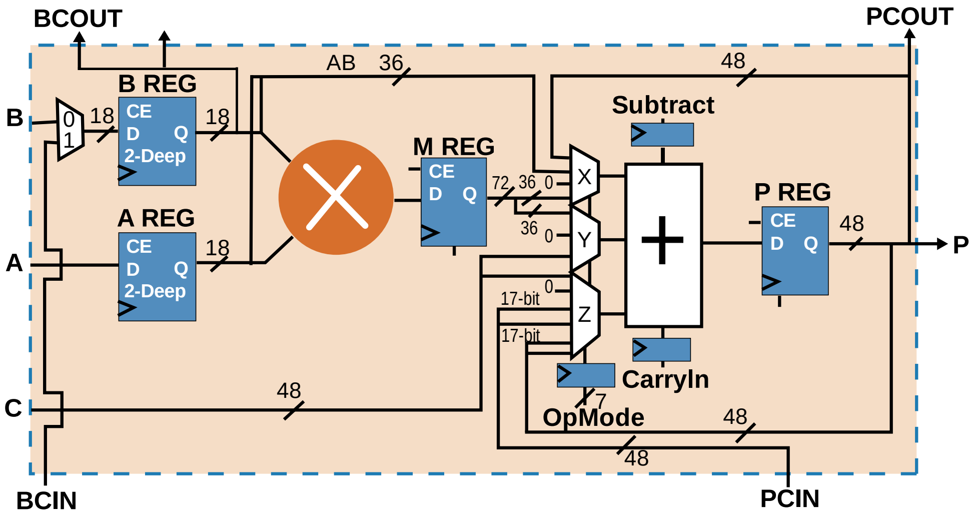
Figure 3a: *Simplified block diagram of Xilinx Virtex-5 FPGA's DSP48E slice*



Features

- Multiplier - 25 x 18
- Cascade inputs - Two
- Cascade output
- Dedicated C input
- Adder - 3 input 48 bit
- ALU logic functions
- Pattern detect - Yes
- SIMD ALU support
- Carry signals - In and out

Figure 3b: *Simplified block diagram of Xilinx Virtex-4 FPGA's DSP48 slice*



Features

- Multiplier: 18 x 18
- Cascade inputs – One
- Cascade output – Yes
- Adder- 3 input 48 bit ALU
- Carry signals - Carry in

Source: Adapted from Xilinx Inc. XtremeDSP Solutions March 2008

DSP Performance on FPGA Modules

Digital signal processing is a very complex and demanding application. Measuring DSP performance is likewise complicated. Performance measures are driven by the application as well as the resources available both within the FPGA device and the module on which it resides. The executable speed of digital signal algorithms depends upon the speed of the processor, the FPGA, the efficiency of the DSP slices, the nature of the application, and the degree of simultaneous versus serial processing. Additional factors include the speed and type of data storage devices both within the FPGA and on the PMC module (e.g. block RAM, DDR SDRAM, dual-port SRAM), the amount of memory available, and the method of data transfer to/from the FPGA and the host CPU. By answering the questions below, it is possible to define the context of execution for your application.

1. How fast can the digital signal algorithms be executed?
2. How much DSP computation capability is available?
3. Is signal processing serial or parallel?
4. How much memory is available for raw, intermediate, and final computational data?
5. What communication mechanisms are available and at what speed for data transfer?
6. How easy is the application to deploy?
7. What are the cost/performance tradeoffs?
8. What is the availability of the solution?

A discussion of the above questions follow below.

1. How fast can the digital signal algorithms be executed?

As an example, Acromag’s PMC-LX/SX series features the Xilinx Virtex-4 FPGA which executes at 500MHz and is rated at over 256 billion multiply-accumulate operations (MACs) per second. Acromag’s PMC-VLX/VSX series with Virtex-5 FPGAs executes at 550MHz and is rated at over 350 billion MACs per second. For extreme applications, the PMC modules with a Virtex-5 FPGA more than a 36% increase in raw computation performance compared to modules with the Virtex-4 FPGA.

2. How much DSP compute capability is available?

Not counting the discrete logic component of the FPGA utilized, the following chart summarizes the number of DSP slices available in Acromag's PMC-FPGA family:

Table 1: *DSP resources of Acromag PMC Modules*

Acromag Model	Xilinx Part Number	# of DSP Slices	DSP Slice Type
PMC-LX40	XC4VLX40	64	XtremeDSP DSP48
PMC-LX60	XC4VLX60	64	XtremeDSP DSP48
PMC-SX35	XC4VSX35	192	XtremeDSP DSP48
PMC-VLX85	XC5VLX85T	48	XtremeDSP DSP48E
PMC-VLX110	XC5VLX110T	64	XtremeDSP DSP48E
PMC-VLX155	XC5VLX155T	128	XtremeDSP DSP48E
PMC-VSX95	XC5VSX95T	640	XtremeDSP DSP48E

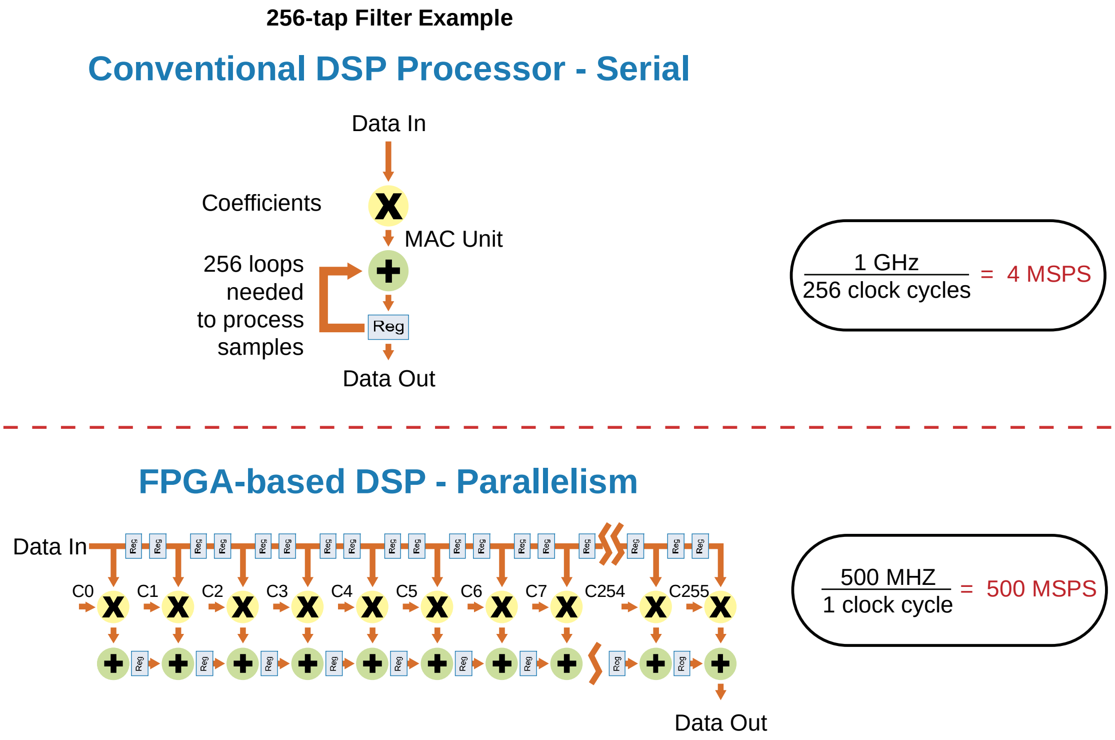
Figure 4: *FPGA modules in a variety of configurations*

3. Is processing serial or parallel?

DSP applications can be implemented in several ways. A serial architecture is where a single input Signal_Source is processed is a single flow, without adjacent simultaneous Signal_Source input influence. Serial architectures may be as demanding upon computer resources as applications of multiple simultaneous signal flows with iterative convergence in-sequence calculations if the intensity of the calculations and the throughput requirements are such. Discretely designed DSP modules for serial architectures are simple in design, however, when multiple simultaneous signal flows and iterative in-sequence calculations are required, separate and distinct Signal_Source processing logic and DSPs become necessary. This type of application is very customized and expensive in design, build and test. With an FPGA implementation, however, it is possible to have multiple simultaneously executing networks of DSP slices and logic yield the benefits of a parallel system architecture. Figure 5 illustrates the differences of using FPGA-based DSP or a standard DSP processor with high computational workloads.

Xilinx FPGAs offer Digital Clock Managers (DCMs) to assist in the orchestration of parallel-executing networks of DSP slices and logic. Each DCM may be configured to generate clock synchronization at a different frequency for each network to which it is assigned. The available clock source frequencies are derived from crystals on the PMC module or from external clock sources injected through FPGA inputs. Virtex-4 FPGAs provide 8 mutually distinct DCMs; while the newer Virtex-5 FPGA provides 12 DCMs.

Figure 5: *Comparison of conventional and FPGA-based DSP processing*



Source: Adapted from Xilinx Inc. XtremeDSP Solutions March 2008

4. **How much memory is available for raw, intermediate, and final computational data?**

Three sources of memory are provided on each Acromag FPGA PMC module:
(i) Block RAM sourced from the FPGA device, (ii) DDR SDRAM sourced on the PMC module, (iii) Dual-ported SRAM sourced on the PMC module. Each of these memory storage areas offer different capacities and speeds. They can be assigned by the FPGA DSP design engineer to best meet the application.

Table 2: *Memory resources of Acromag PMC Modules*

Acromag PMC Modules	**FPGA Block RAM Capacity & Speed**	**DDR SDRAM Capacity[1] & Speed[2]**	**Dual-Ported SRAM Capacity[1] & Speed[2]**
PMC-LX40	1,728 Kbits 500MHz	32M x 32-bit 150MHz	256K x32-bit 66MHz
PMC-LX60	2,880 Kbits 500MHz	32M x 32-bit 150MHz	256K x32-bit 66MHz
PMC-SX35	3,456 Kbits 500MHz	32M x 32-bit 150MHz	256K x32-bit 66MHz
PMC-VLX85	3,888 Kbits 550MHz	32M x 32-bit 150MHz	256K x64-bit 66MHz
PMC-VLX110	5,328 Kbits 550MHz	32M x 32-bit 150MHz	256K x64-bit 66MHz
PMC-VLX155	7,632 Kbits 550MHz	32M x 32-bit 150MHz	256K x64-bit 66MHz
PMC-VSX95	8,784 Kbits 550MHz	32M x 32-bit 150MHz	256K x64-bit 66MHz

Note 1: Capacity figures in chart reflect standard configuration of the Acromag PMC Module; contact Acromag for expansion capacity.

Note 2: Speeds are based upon configurations provided in Acromag’s Engineering Design Kit for each FPGA module. Depending on design criteria, these levels may be modified. Contact Acromag for details.

5. **What communication mechanisms are available and at what speed for data transfer?**

Digital signal processing is an extremely compute intensive function requiring high processing speeds and the ability to move data efficiently and quickly. Typically, PMC FPGA modules support interrupts, DMA transfers, and serial protocols utilizing the FPGA I/O. Acromag’s Virtex-4 FPGA modules support DMA transfers that comply with the PCI r2.0 specification to deliver over 60MB/sec throughput (32 or 64 bit word transfers at up to 66MHz). Acromag’s Virtex-5 FPGA modules feature PCI-X r3.0-compliant DMA transfers to deliver over 600MB/sec throughput (32 or 64 bit word at up to 100MHz). Serial LVDS transfers are verified for over 150MHz per I/O point. This magnitude of data throughput is essential for high speed digital signal processing.

6. How easy is the application to deploy?

Many PMC FPGA modules are designed to meet military Commercial-Off-The-Shelf (COTS) requirements. COTS modules are usually supported by design and deployment tools provided by both the FPGA manufacturer and the board vendor. The FPGA manufacturer provides tools covering the application design, programming, and debug. The board vendor provides a Board Support Package (BSP). As an example, Acromag offers an Engineering Design Kit which includes all of the FPGA-to-module interfaces in an operational format that is demonstrable when used with Acromag's OS support software for Windows®, VxWorks®, QNX®, and Linux® environments. With a good BSP, a designer can prove the module and host CPU are co-operational in 15 minutes or less.

Later in this paper, the design and development tools for PMC FPGA deployment will be discussed in more detail.

7. What are the cost/performance tradeoffs?

Compared to traditional discrete DSP hardware module development, use of an FPGA-based DSP module is now often the designer's choice for development time reduction, design/development/debug tool support, and the availability of pre-defined logic (IP cores). The re-configurable nature of FPGA-based modules adds flexibility, upgradeability, and even re-use of the modules in other applications with just simple re-programming. When hardware design becomes a software re-configuration, it is just a matter of developing the initial FPGA platform module to support the external signal interfaces with the proper communication and performance characteristics. In the actual software configuration, limits on the FPGA platform module's capability to support the external signal interfaces can be imposed. This is where PMC FPGA modules can make application deployment very cost effective. Acromag offers a variety of PMC FPGA modules starting at just over $3000, each with different performance levels for DSP processing.

8. What is the availability of the solution?

FPGA manufacturers continue to push the envelope with more and more performance from newer generations of devices. However, system developers must ask themselves if it is worth waiting for the next generation or can the current generation adequately satisfy their application. The time lag between the announcement of a new FPGA device and the release of platform modules that deploy the new chipset can be significant. In many cases, one can begin development on the current generation of modules for proof of concept and then upgrade when the new modules are available with minimal additional efforts.

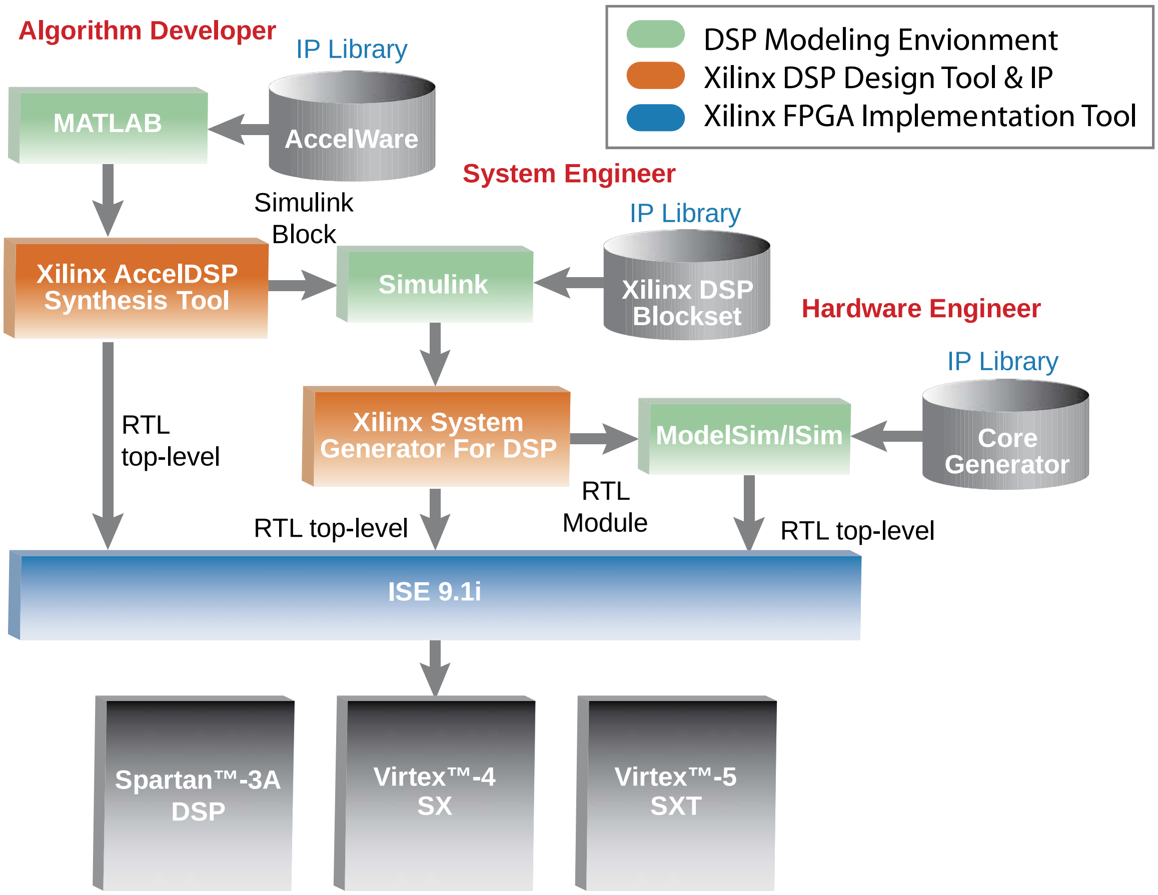
Figure 6: *Modules with Xilinx Virtex-5 FPGAs have recently begun to ship while newer FPGA devices within the series continue to be released over time.*

Developing FPGA-based DSP Applications Using MATLAB®

It is well understood by designers of DSP applications that the complexity of developing DSP systems involves both the accurate modeling of the of the signal processing flow and the precise control/synchronization of the data flow. Designers must accurately control the data flow between device interfaces, processing blocks, sub-DSP systems, and memories interlaced throughout the signal processing flow.

MathWorks® is well known for tools dedicated to handling both of these critical aspects of the DSP application development process. MATLAB aids in the mathematical modeling of the signal processing flow in a high level interactive environment. SIMULINK® assists in the definition and modeling of accurate, timely control and synchronization of the information flow. These two tools combined with the Xilinx AccelDSP™ and Xilinx System Generator for DSP make it possible for engineers to take advantage of the best modeling tools to simulate DSP performance and facilitate transfer of logic into the FPGA. Figure 7 outlines the process.

Figure 7: *Flowchart for development of FPGA-based DSP applications.*



Source: Adapted from Xilinx Inc. XtremeDSP Solutions 2007

DSP application development is a complex process in many aspects. Understanding and modeling the application is the first major hurdle. Traditionally, DSP engineers would use readily available programming tools such as C, C++, Fortran, Visual Basic® and the like to develop and test algorithms to be deployed in the DSP application. Although not insurmountable, the use of productivity tools such as MATLAB make this task much easier and provide a development environment inclusive of:

- a high level modeling language
- access to add-on toolboxes with extensions for signal processing, communications, and wavelet processing
- code management facilities
- interactive tools to explore and aid in the DSP application design
- commonly used mathematical functions (linear algebra, statistics, Fourier analysis, filtering, etc.)
- optimization tools

- graphical visualization and publishing tools to aid in the modeling process and recording of data
- ability to integrate new or existing algorithms from applications written in C, C++, Fortran, Excel®, etc.

The high level MATLAB language supports a wide and increasing range of vector and matrix operations, as well as common arithmetic operators, flow control, data structures, data typing, object-oriented programming, and debug features. The MATLAB Editor, aside from providing a simple to use editor, includes breakpoint and logic single stepping as debug aids. Other performance optimization tools include the M-Lint Code Checker which will analyze the MATLAB code and recommend changes for performance enhancement and maintainability. The MATLAB Profiler enables the developer to measure and rationalize the time spent in the various lines of coded application.

At this point, the DSP engineer will have suitably modularized the DSP application and completed testing for accuracy and performance, but only at the algorithmic modeling level. MathWorks SIMULINK can now be used to combine the algorithmic modeling level product developed with MATLAB into an interactive graphical environment. This environment enables the DSP engineer to design, implement, simulate and test the DSP application within the context of required control, synchronization, and parameterization on a processor based platform.

To accomplish the above tasks, SIMULINK provides extensive pre-defined block libraries, graphical editing and management tools for assembling and managing intuitive block diagrams, model segmenting tools, simulation modes of Normal / Accelerator / Rapid Accelerator, and a number of model analysis and diagnostic tools. Models are built by combining drag & drop blocks and connecting them with lines that determine the relationships between the blocks. The blocks are segmented into design components then simulated and tested either individually or collectively. Additionally, the design components can be saved and used across projects.

After the model is built in SIMULINK, it becomes possible to simulate the timing, synchronization, and algorithmic processing afforded by the DSP design. Diagnostic tools are available to compute system dynamics, evaluate timing and synchronization, and diagnose the overall behavior of the model – all from a graphical user interface. Simulation is configurable to execute in normal or accelerated timeframes. Information can be collected on performance bottlenecks and documentation added. Using the HDL Coder, it is possible to generate FPGA target VHDL and test benches.

Once the DSP engineer has successfully demonstrated that algorithmically and structurally the model meets application requirements, it is necessary to implement the MATLAB models in the appropriate form factor for execution on the Xilinx FPGA platform of choice. The Xilinx System Generator for DSP is a SIMULINK-like graphical environment designed to create DSP designs in FPGAs. Access to a pre-defined block of Xilinx cores facilitates interfacing with Xilinx FPGAs.

The Xilinx AccelDSP synthesis tool enables DSP algorithm designers to incorporate their MATLAB M-files to perform stimulus definition and generation, algorithm evaluation, and results post-processing in the context of the Xilinx FPGA. Remember, SIMULINK enables the DSP engineer to design, implement, simulate and test the DSP application within the context of required control, synchronization, and parameterization on a processor based platform…not on the FPGA itself. The Xilinx AccelDSP synthesis tool creates, based upon the MATLAB floating point models, System Generator IP blocks in cycle-accurate fixed point. The fixed point design can be simulated with options for saturation and rounding applied. Precision growth through arithmetic operations is automatically propagated through the design under user-controlled override options until the designer determines the suitability of the performance against the accuracy.

Fine Tuning the FPGA Design

At this point, the AccelDSP synthesis tool is used to generate the Register Transfer Level (RTL) for the target FPGA device and to apply optimizations as might be applicable and permissible by boundary conditions (performance requirements, memory cells available, block RAM availability, etc.). Some of the optimizations and their effect are summarized in Table 3:

Table 3: *DSP optimizations and their effects*

DSP Synthesis Directive	Effect on Generated Hardware
Rolling / unrolling of loops	Improves input sampling rate by reducing throughput
Expansion of vector and matrix additions and multiplications	Improves input sampling rate by reducing throughput
RAM / ROM memory mapping of arrays	Improves FPGA utilization by mapping arrays into dedicated Xilinx Block RAM resources
Pipeline insertion	Improves input sampling rate by improving clock frequency performance
Shift register mapping	Improves FPGA utilization by mapping shift register logic into SRL16s

Utilizing the synthesis directives produces a very hardware-specific design composition RTL. The AccelDSP synthesis tool will permit evaluation of the entire algorithm based upon the RTL and perform boundary optimizations when possible. Additionally, it will report throughput and latency which are necessary for the DSP designer to judge performance of the design prior to generating a cycle-accurate System Generator model.

Final Simulation and RTL Code Generation

Once the RTL is successfully generated and optimized to perform at required specifications, the AccelDSP synthesis tool can be used to generate a System Generator IP block which will support both simulation activities and RTL code generation. This new IP block appears in the SIMULINK library browser. To incorporate the new AccelDSP IP block into a model, it is only necessary to select the IP Block and drag it into the destination model. To compile your design for installation on the Xilinx FPGA target, use System Generator for DSP to generate the appropriate bit stream file(s) which can then be converted into .mcs files for PROM load using the Xilinx iMPACT utility.

Summary and Conclusions

Using the well-known mathematical modeling of MATLAB together with the conversions and optimizations available through the Xilinx AccelDSP synthesis tool and Xilinx System Generator for DSP, it is possible to enjoy the benefits of desktop simulation, test and the final Xilinx FPGA hardware-specific targeting of DSP applications. Acromag provides a large family of Xilinx-based PMC FPGA products which augment these capabilities. A variety of models offer many performance levels with a range of available logic cells and DSP Blocks, memory capacities, and many I/O options. The combination of tools from MathWorks and Xilinx, plus the flexibility and breadth of the Xilinx FPGA line available on many PMC modules, provides a tremendous development time and cost advantage to DSP engineers.

For more information, contact:

Joe Primeau	P: 248-624-1541 x1823	E: jprimeau@acromag.com
Rowland S. Demko	P: 248-624-1541 x1825	E: rdemko@acromag.com
Ronald Moquin	P: 248-624-1541 x1824	E: rmoquin@acromag.com