

---

## DSP Acceleration Using a Reconfigurable Coprocessor FPGA

Digital signal processors (DSPs), like their FPGA counterparts, are proliferating into a broad range of compute-intensive applications, including telecommunications, networking, instrumentation and computers. DSP functions include adaptive finite impulse response (FIR) filters, fast Fourier transforms (FFT), interpolators and discrete-cosine transforms (DCT). These functions are required to implement the audio/video compression and decompression, encryption, convolution and other computing functions found in multimedia, digital switches, vision inspection equipment, and many other electronic systems. There are several general purpose DSPs available today, from TI, Motorola, Analog Devices, NEC and others. Recently, specialized processors have been introduced for specific classes of applications, such as compression chips from C-Cube and 3D graphics accelerators from S3. DSPs, like FPGAs, have associated design tools to assist in the development and implementation of DSP-based applications.

Stand-alone digital signal processors (DSPs) support many on-chip functions and are highly optimized for the demands of high-speed computing. Many applications require functions to be implemented that the DSP is not optimized for, or require higher precision or performance than the DSP is capable of. In order to achieve the required functionality and speed, three options are available to the designer: write software code, use a faster, more expensive, or different processor, or design a custom gate array. For option 1, software code must be written and executed in the

processor. The number of instructions required to execute a function is proportional to the decrease in execution time; the more instructions the slower the data output rate. Even though the DSP may be clocked at 50 MHz, the data output is a fraction of that speed. In order to regain this lost speed an additional DSP (either the same or another type), or custom ASIC must be used to achieve the desired function and speed. This results in increased power consumption, added cost for additional design tools, non-recurring engineering expenses (NRE), increased design time (due to learning a new architecture and new tools), and design risk. There is also a loss of design flexibility with ASIC solutions. Processors and ASICs are not able to easily adapt in real-time to different algorithms.

FPGAs can now be used in DSP applications either as a stand-alone, reconfigurable processor or as a coprocessor to the DSP. This 2-chip, embedded DSP/coprocessor solution can effectively eliminate the problems described above. A standard "core" can be developed, with the DSP implementing standard processing functions, and the FPGA used to accelerate specialized functions not easily implemented or quickly executed in the DSP. An example of this is the creation of adaptive digital filters which can be implemented in FPGAs, since the coefficients and the filter structure can be changed through reconfiguration of the FPGA. Register rich FPGAs can implement complex datapath logic functions using similar pipelining and bit-serial processing techniques as those found in today's advanced DSPs.



---

### Field Programmable Gate Array

---

### Application Note

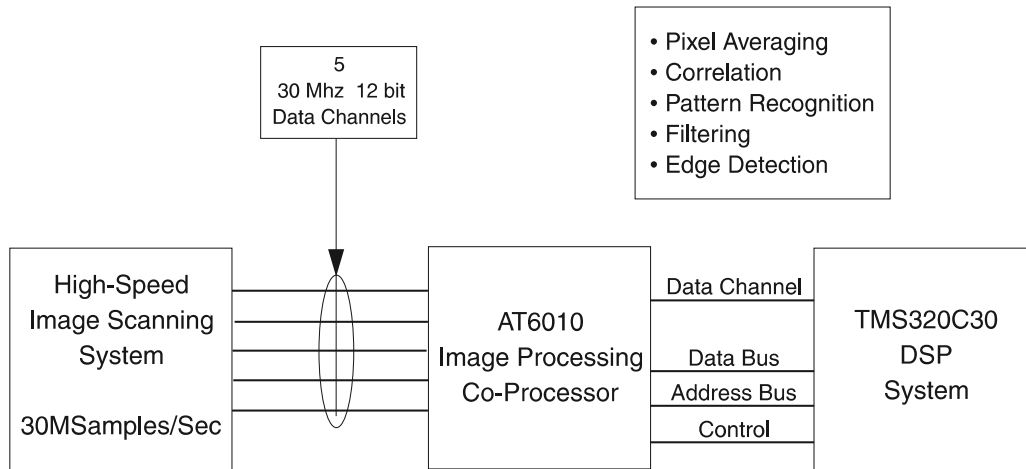




Using the techniques described herein, performance improvements are possible in many compute-intensive applications, including those shown in Figure 3. Image processing systems include scanning, processing and output of data. The TMS320CXX is a popular, low-cost digital signal processor for these types of applications. Typical DSP functions include pixel averaging, pattern recognition, and edge detection. By using a 20,000 gate AT6010 as a pre-processor to the TMS320C30-based system, the data rate can be improved by a factor of 50. For pattern recognition, patterns, or sequences of data can be loaded or programmed directly into the FPGA, giving the ability to

perform hardware searches, instead of software. The XOR gate, or exclusive-OR, is the basic logic element for building compare functions. The AT6010 has 6,400 XOR gates. The combination of the XOR gates, along with dynamic reconfigurability of the FPGA, allows multiple patterns to be loaded into the FPGA as required. This enables searches and comparisons to be done in hardware, instead of software. The result of this capability is a 30 times improvement in speed for pattern recognition. Edge detection, the process of identifying when an object comes into the field of view, can be improved by a factor of 100 by using the AT6010 as a coprocessor.

**Figure 3.** The 20,000-gate AT6010 as a DSP Coprocessor



- **Pre-Processing (i.e... Pixel Averager - 50X reduction of data rate)**
- **Co-Processing (i.e... Pattern Recognition - 30X improvement in response time)**
- **Co-Processing (i.e.. Edge Detection - 100X improvement in throughput)**

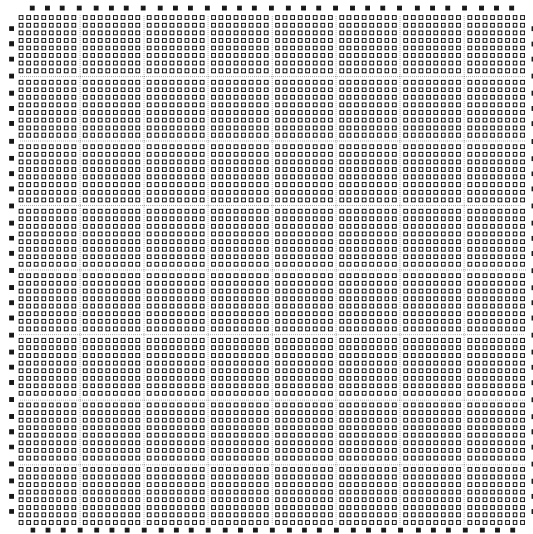
## AT6000 FPGA Architecture

The AT6000 architecture consists of thousands of logic cells as shown in Figure 4. Each cell contains a D-type register and about 12 gates of logic. The specified data shift frequency for the registers is 250 MHz. In the 20,000+ gate AT6010, 6,400 registers can be combined to build a 6,400 bit serial shift register shifting data in excess of 250 MHz. The registers are the building blocks for implementing pipelined functions. By utilizing the thousands of registers, very high speed, pipelined computing functions can be implemented. For example, an 8 x 8 pipelined multiplier can be implemented in the AT6000 and run at 54 MHz. A 16 x 16 multiplier can be executed at 36 MHz using pipelining. The single level logic delays of the AT6000 FPGA are 2 ns, and the input buffer delay is 1.2 ns; the output buffer delay is 3.5 ns. Data can be loaded into, processed and output at extremely fast data rates.

The AT6000 family is organized in rows and columns of cells; the size of an array is determined by the specific number of cells. The smallest member of the AT6000 family, the AT6002, has 32 rows by 32 columns, for a total of 1,024 cells. This device can implement between 2,000 and 4,000 gates of logic. The largest member is the AT6010, organized 80 x 80 cells, or 6,400 total cells. This device can implement in excess of 20,000 gates of logic.

The AT6000 family is completely symmetrical, enabling logic blocks to be placed in the array without regard to chip size or orientation (e.g. large blocks of logic can be placed in a 2,000 gate device or a 20,000 gate device, rotated by 90, 180, 270 degrees or flipped over the X or Y axis without impacting speed or function).

**Figure 4.** AT6000 Series Array



## Automatic DSP Component Generators

The automatic component generator is a tool that is integral to the Atmel FPGA design system, which works with tools from Cadence/Verilog, ViewLogic, Synopsys, Mentor, Exemplar, Integraph and others. This tool allows the designer to specify the parameters of hundreds of different complex logic functions, and quickly create fixed functional layouts, as well as schematic library symbols and actual schematics for the created function. The generated function is fully defined, including speed, size, power consumption. These symbols can be called up from the [ViewLogic, Mentor, etc.] schematic capture or synthesis tool, and used just as any other macro library function.

These component generators have been designed to exploit the AT6000's register rich, dynamic reconfigurability by creating reusable, high speed custom logic functions, resulting in significantly improved performance for compute-intensive and other logic applications. These function generators are a powerful design tool because they can quickly (1000 gate multiplier in 3 - 4 minutes) create the schematic and physical layout (shape) of a function, and report the speed, area and power consumption.

This capability is much more useful than providing benchmark data to the designer because with Atmel's component generators the user can create reusable, custom logic functions based on the specific parameters associated with them (e.g. an 8 x 12 multiplier optimized for speed, a 14-bit adder optimized for minimum area, etc.). Until now, no other tools or architectures have supported this unique capability. Some tools exist that allow the user to specify functional parameters (creating a netlist for the function), but they can not report back the speed and area results until the entire design has been physically placed and

routed. Because the AT6000 FPGA architecture is symmetrical, component-generated functions can be created without any layout constraints, resulting in fully predictable delay/speed calculations before place and routing the rest of the design. This is particular useful in DSP design for creating the DSP building blocks and facilitating the generation of hardware coefficients.

## Cache Logic®

Cache logic is a cost-saving way of implementing logic more efficiently. The active functions of an application are performed by a field-programmable gate array (FPGA) that can be reconfigured as it operates, while inactive functions are stored in an inexpensive configuration memory – an EPROM for example. As new functions are required, they are written over old ones, as shown in Figure 6.

Cache logic implementation is similar to cache memory. In cache memory, the highest speed memory (usually SRAM) is used to store active data, while the bulk of data resides in lower-cost storage, such as DRAM, or EPROM, disk etc. Cache logic works in a similar fashion. Only a small fraction of the circuitry is active in a system at any given time. Only those functions which are loaded into the logic cache, while unused functions, or variations reside in lower cost system memory. It is even possible to compile variations in a design in real time. As logic functions are required, they can be loaded into cache logic, replacing, or complementing the logic already present. This is particularly useful for adaptive filter designs, where the algorithm is resident in the FPGA, with the coefficients stored in configuration memory. As a new coefficient is required in the filter, it is loaded into the FPGA.

Figure 5. Automatic DSP Component Generators

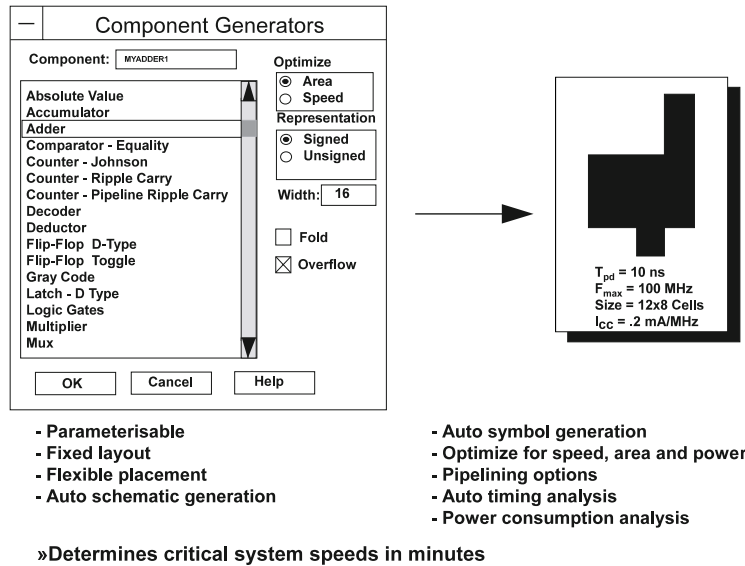
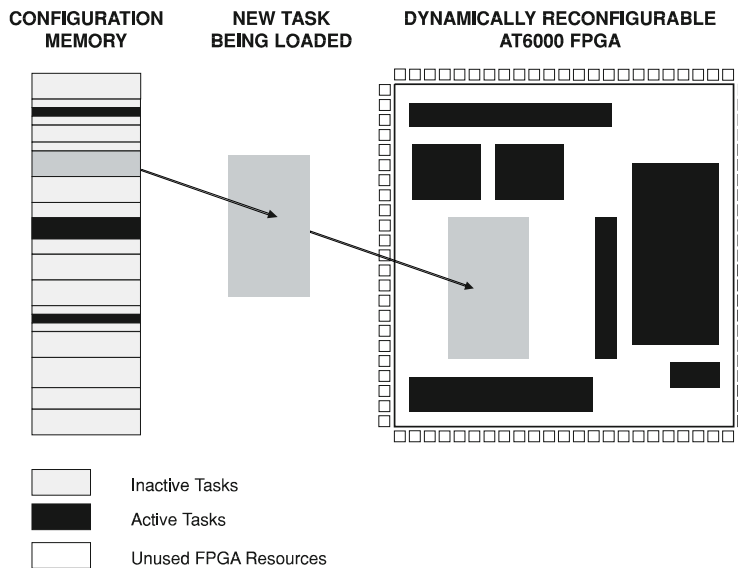


Figure 6. Cache Logic



## Summary

By using cache logic FPGAs, specialized compute-intensive functions can be directly in hardware. Reusable, custom DSP parameterized functions can be created by using the DSP component generator software.

The benefits of this approach are faster processing speeds, reduced board space and power consumption, with shorter time to market. As new algorithms are developed they can be downloaded (remotely) into the configuration memory to update the hardware. This in turn can extend the life of a product, enabling companies to squeeze more life, and money out of older products, without lengthily design cycles and retooling processes.

Atmel AT6000 series FPGAs are the first to enable full Cache Logic design implementation. The ability to fully or partially reconfigure the FPGA in microseconds, without loss of register data and disruption of the operation of the unchanged logic enables real time datapath acceleration. The symmetrical nature of the FPGA allows “standard cell” functions to be placed in any size array, in any orientation with affecting the speed or operation of the function. Compute-intensive DSP applications can be executed in real time hardware without increasing system overhead.



## **Atmel Headquarters**

### ***Corporate Headquarters***

2325 Orchard Parkway  
San Jose, CA 95131  
TEL (408) 441-0311  
FAX (408) 487-2600

### ***Europe***

Atmel U.K., Ltd.  
Coliseum Business Centre  
Riverside Way  
Camberley, Surrey GU15 3YL  
England  
TEL (44) 1276-686-677  
FAX (44) 1276-686-697

### ***Asia***

Atmel Asia, Ltd.  
Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimhatsui  
East Kowloon  
Hong Kong  
TEL (852) 2721-9778  
FAX (852) 2722-1369

### ***Japan***

Atmel Japan K.K.  
9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
TEL (81) 3-3523-3551  
FAX (81) 3-3523-7581

## **Atmel Operations**

### ***Atmel Colorado Springs***

1150 E. Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL (719) 576-3300  
FAX (719) 540-1759

### ***Atmel Rousset***

Zone Industrielle  
13106 Rousset Cedex  
France  
TEL (33) 4-4253-6000  
FAX (33) 4-4253-6001

---

### ***Fax-on-Demand***

North America:  
1-(800) 292-8635  
International:  
1-(408) 441-0732

### ***e-mail***

literature@atmel.com

### ***Web Site***

<http://www.atmel.com>

### ***BBS***

1-(408) 436-4309

#### **© Atmel Corporation 1999.**

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

0724B-09/99/xM