



An Introduction to DSP Applications using the AT40K FPGA

**FPGA Application Engineering
Atmel Corporation
San Jose, California**

Overview

The use of SRAM-based FPGAs in digital signal processing is now considered a viable means of offsetting DSP microprocessor performance limitations in applications that require high data rate processing and in designs that need to perform high-speed binary manipulation of data words or signals. In applications such as machine vision, high-speed control, digital communications, and video, standard DSP processors can be ill-suited to perform at the required rates due to the serial nature of their architecture or the lack of certain kinds of instructions.

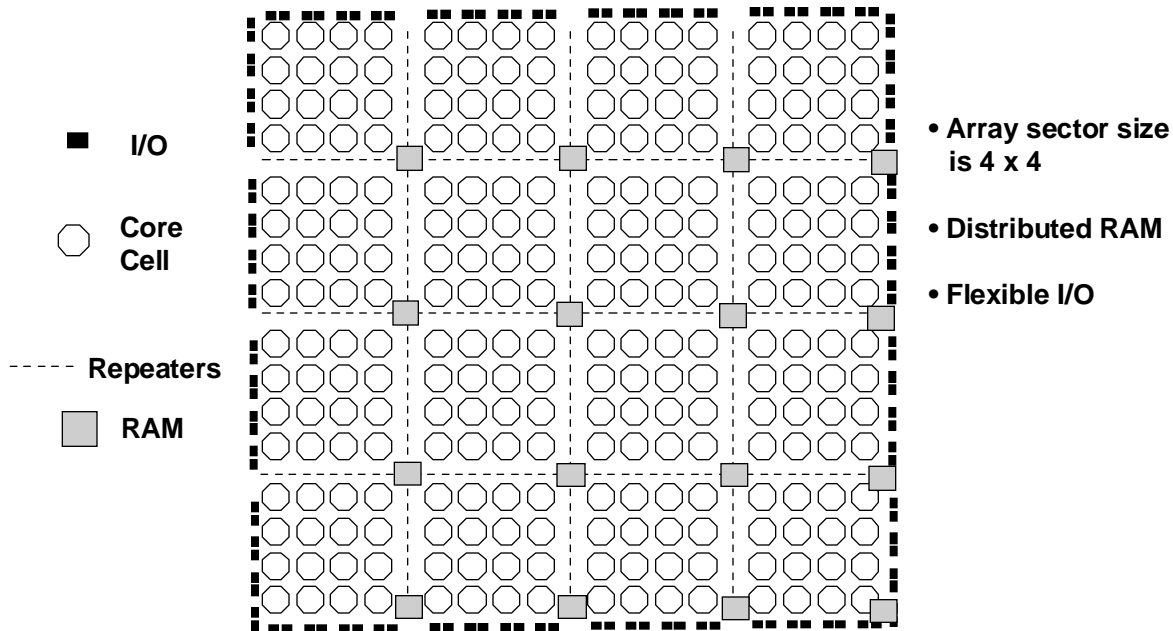
Programmable logic devices have been used successfully to mitigate these problems by performing them in hardware, bypassing the stored-program techniques in favor of dedicated logic functions. Multiplier, adder, accumulator, and comparator functions in FPGAs exploit hardware parallelism to add critical processing power to DSP systems. FPGAs provide the flexibility to meet the demanding specifications in real-time video, imaging, wireless systems, and high-performance digital audio. The new Atmel AT40K FPGA family has been designed with these applications in mind. This document offers an introduction to some of the macrofunctions employed in these applications using the AT40K family of devices.

The Atmel AT40K Coprocessor FPGA Family

The AT40K is a family of fully PCI-compliant, SRAM-based FPGAs with distributed 10ns programmable synchronous/asynchronous dual-port/single-port SRAM, 8 global clocks, Cache Logic ability (partially or fully reconfigurable without data loss) and range in size from 5,000 to 50,000 usable gates. I/O counts range from 128 to 384 in industry standard packages ranging from 84-pin PLCC to 475-pin BGA, and support 3-volt and 5-volt designs.

The Atmel AT40K-series FPGA architecture is tuned for high-speed computing and datapath applications while offering considerable advantages for a variety of applications from networking to reconfigurable computing. Real-time reconfigurable LUT-based logic-cells are organized in a direct-connected systolic array. The logic-cells are also coupled to an extensive multiplane bus routing system that is combined with dedicated, distributed memory blocks and an innovative I/O structure. These features allow the AT40K to provide a synthesis-friendly architecture for today's HDL-based designs while maintaining a compute-intensive focus. Like the Atmel AT6000 FPGA family before it, the AT40K FPGA is dynamically-reconfigurable, implementing an advanced functional mapping of its internal configuration memory. This mapping scheme enables designers of reconfigurable systems or products to exploit the FPGA reconfiguration capability with ease and efficiency by developing software running on a local microprocessor, microcontroller, or DSP processor.

AT40K Device Overview



The AT40K FPGA Architecture is SRAM-based with a symmetrical, medium-grained array of LUT-based logic-cells. The general overview of the device architecture is shown below. Routing within the array is either by one of the five bussing planes (both vertical and horizontal) or by octagonal direct connection (both orthogonal and diagonal) between all eight neighboring cells. Connective units, called repeaters, spaced every four cells, divide each local bus-plane into segments spanning four cells. These busses and cells run in two dimensions. An array of 4x4 cells surrounded by the repeaters is called a sector.

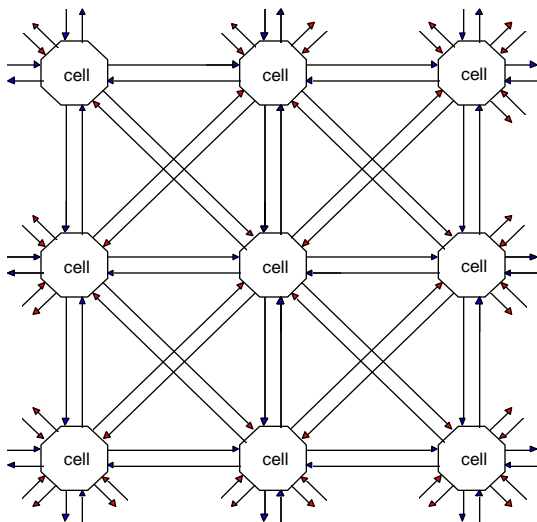
Ten local busses (five vertical and five horizontal) serve each four cell span. The two express busses per bus-plane span two sectors in an alternating leap-frog fashion. Therefore, there are forty local busses (twenty vertical and twenty horizontal) with a direct logic-cell interface and eighty express busses (forty vertical and forty horizontal) passing through each sector, providing ample routing resource for the most demanding designs. The repeaters provide local-to-local bus connections, local-to-express bus transfers, and facilitate the extension of the local-busses when used in tri-state mode.

The AT40K consists of an array of sectors, the number of which will define the total cell count of each individual part. AT40K FPGAs contain small dedicated SRAM blocks that are distributed across the array. The 32x4-bit SRAM blocks are located at the intersection of the sectors and are programmable to perform synchronously or asynchronously and can operate in either a single- or dual-port manner.

The result is an FPGA architecture that is well-suited to DSP, image processing, and digital communications applications. The AT40K can be used as a coprocessor for high-speed DSP/processor designs by implementing a variety of compute-intensive, arithmetic functions.

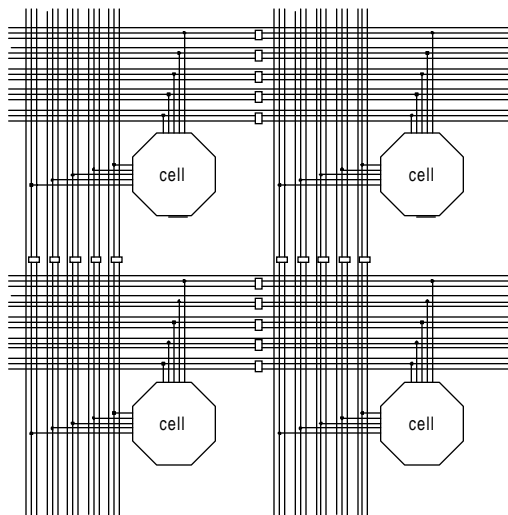
These include digital filters, Fourier transforms (FFT/DFT), convolvers, correlators, and discrete cosine transforms (DCT) that are required for video processing, compression, encryption, multimedia, telecom, and control applications.

Cell to Cell Direct Connections



- Each Cell connects to 8 nearest neighbors
- Each Cell has 4 orthogonal connections
- Each Cell has 4 diagonal connections

AT40K Cell to Bus Connections



- Each Cell input can be connected to a local bus
- Each Cell output can be connected to a local bus
- There are 5 Local buses horizontally per sector row
- There are 5 local buses vertically per sector column

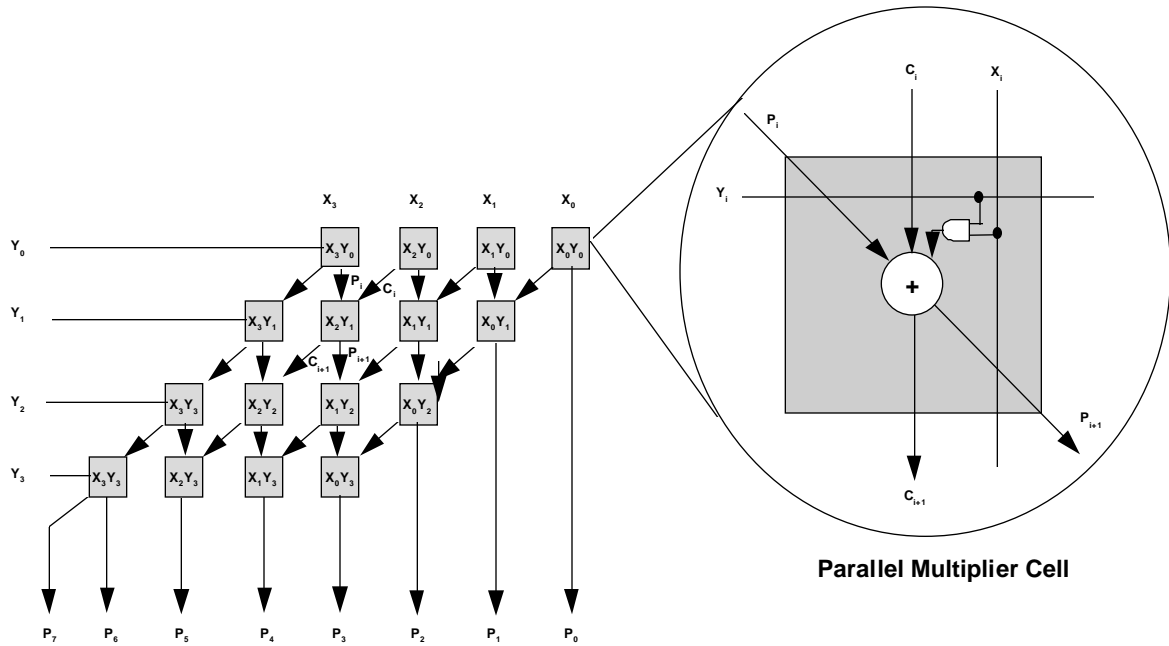
AT40K DSP Building Blocks and Applications

The following macrofunctions and building blocks have been generated as reference designs to provide initial characterization of the performance of the AT40K FPGA for DSP applications. In the coming months, Atmel FPGA application engineering will release a series of application notes and briefs providing additional details and specifications as well as reference designs describing the use of these functions in typical applications.

Array Multipliers

Multipliers are essential functions in most DSP applications. The unique direct cell-to-cell interconnection scheme of the AT40K permits highly-efficient parallel array-type multiplier designs. The AT40K logic cell is eight sided and allows direct connections to contiguous cells on eight sides. This effectively doubles the number of cells that can be directly connected to each other. The ability to connect diagonally allows for the efficient implementation of high-speed DSP-type building blocks such as parallel multipliers and barrel shifters. For example, an convolver with nine 8-x-8 parallel multipliers can be constructed in the AT40K and using only 657 logic cells.

Parallel “Array” Multiplier



Parallel Multiplier Specs

	Multiplier I	Multiplier II	Multiplier III	Multiplier IV
Data Width and Format	8-bit unsigned	8-bit signed	8-bit unsigned	4-bit unsigned
Output Width and Format	16-bit unsigned	16-bit signed	16-bit unsigned	8-bit unsigned
Pipelining	NO	NO	FULL	FULL
Core Cells Employed	64	81	350	68
Operating Frequency (-1 speed)	40 MHz	37 MHz	112 MHz	133 MHz

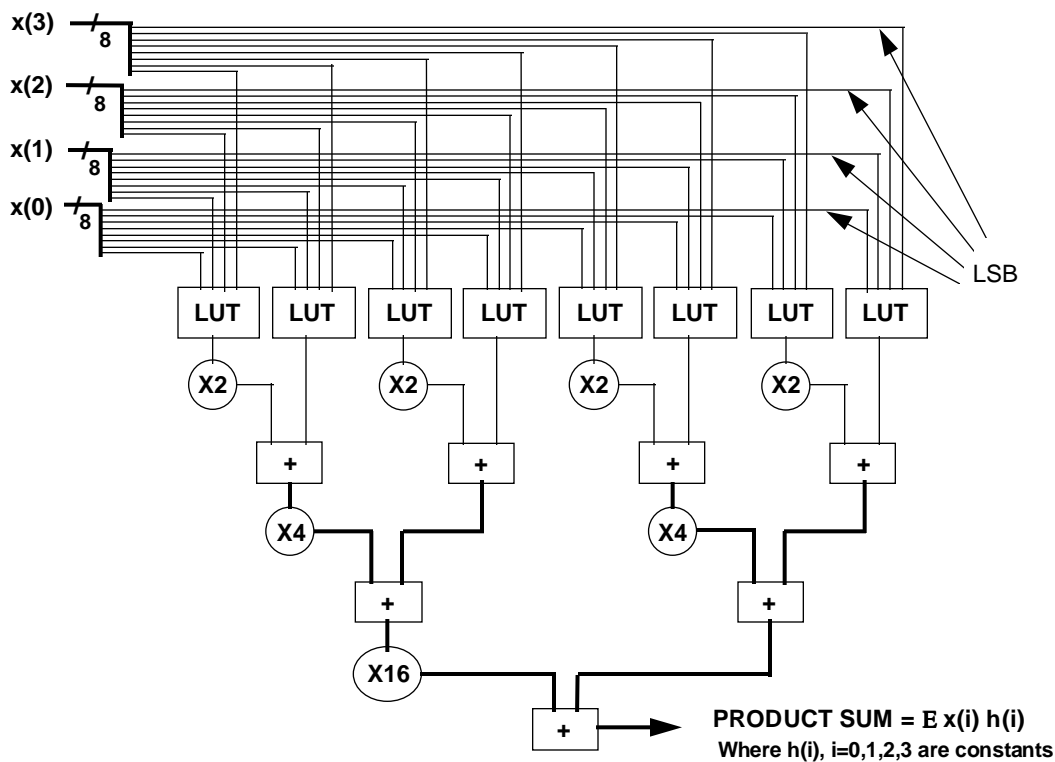
Notes:

- Non-pipelined multipliers all have square layouts.
- 1-stage pipelining also results in square layouts.
- All timing values derived from -1 speed grade.
- Component Generator available for non-pipelined multipliers.

Vector Multipliers

The use of vector multipliers is essential in building compact CCM's (constant coefficient multipliers) used in a variety of DSP and imaging tasks. Vector multipliers have become a hot substitution for regular (parallel) multipliers, especially in applications where multiply-accumulate operations are frequent (e.g. calculate $P = x_{(0)}h_{(0)} + x_{(1)}h_{(1)} + x_{(2)}h_{(2)} + x_{(3)}h_{(3)}$ where $x_{(i)}$ are input variables and $h_{(i)}$ are constants).

Vector Multiplier with Four 8-bit Inputs



Vector multipliers are efficient in FPGAs with LUT-based core-cells. The look-up tables store “partial” products based on the constant coefficients, these partial products are “addressed” by the inputs and accumulated in the adder tree, thereby eliminating the need for large parallel multipliers in applications where the coefficients are fixed or are updated infrequently. Application areas for vector multipliers are digital filters, 2-D convolution, and binary correlation. The Cache Logic capability of the AT40K permits unlimited sets of partial products to be programmed into the LUTs, as required, dramatically increasing system functionality.



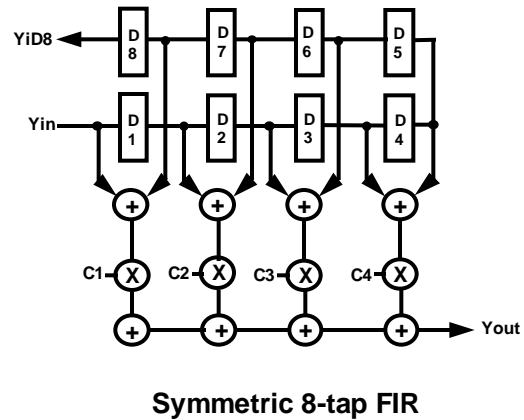
Vector Multiplier Specifications

Sample Implementation	Summation of 4 products
Data Format	8-bit unsigned
Coefficient Format	8-bit unsigned
Output Format	17-bit unsigned
Datapath	Non-pipelined, truncation at final adder stage
Core Operating Frequency (-1 speed)	35 MHz
Device Utilization AT40K10K-2QC-X	33% Logic 0% RAM 0% I/O

FIR Filters

The finite impulse response (FIR) filter is used in many digital signal processing applications to perform a variety of functions, including signal conditioning, anti-aliasing, frequency band selection, signal averaging, decimation, interpolation, and video convolution (2-D filtering of images). Atmel AT40K FPGAs can easily implement FIR filters using a number of techniques. In this case, we use a vector multiplier to replace the usual multiplier-accumulator circuits used in DSP processors. A vector multiplier, as described above, works well in applications where the filter coefficients are relatively fixed. However, utilizing the AT40K dynamic reconfiguration mechanism, real-time update of the vector multiplier makes this approach more attractive and less restrictive than ever before.

- **Advantages**
 - Straightforward to implement
 - Guaranteed stability (all FIRs)
 - Exact linear phase
- **Disadvantage**
 - High order required for narrow transition bands
- **Amendment**
 - Use IIR if linear phase not required

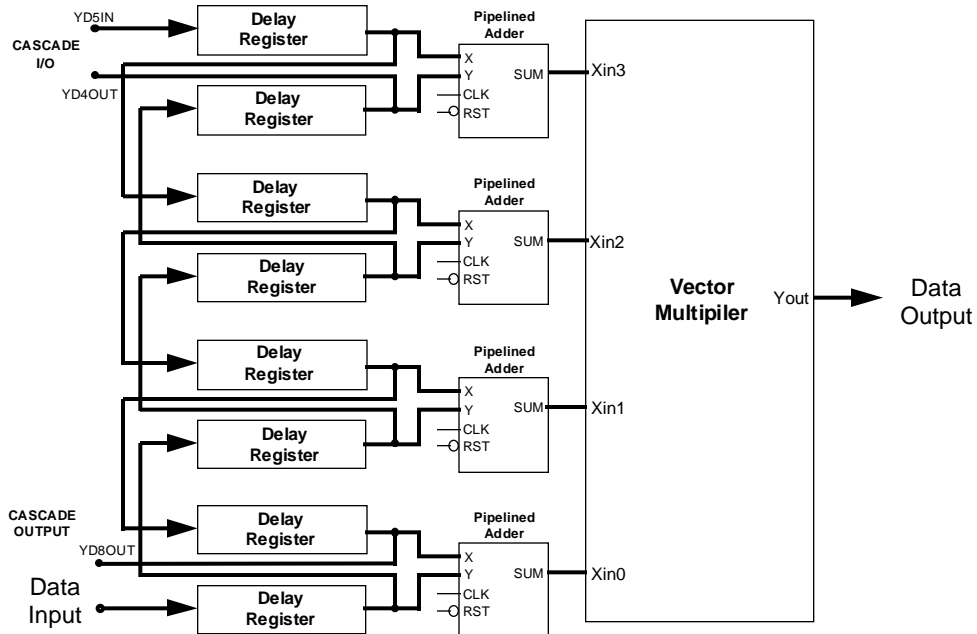


In high sample rate applications, an AT40K can be used in conjunction with a DSP microprocessor. In such applications, the DSP processor can treat the AT40K as a coprocessor, reconfiguring the FPGA for different filter transfer functions or even different processing functions. This allows lower data-rate, complex algorithms to be executed by the DSP processor and the high data-rate signal processing to be performed by the FPGA.

In the figure below, an 8-tap even-symmetrical FIR macrofunction is shown. This configuration is used when the impulse response is symmetrical meaning that the filter coefficients are mirrored along a central axis. This algorithm allows for reduction of the number of multiply operations and consequently a reduction in the size of the vector multiplier. An optimization can occur by pre-adding the input values that correspond to the appropriate coefficient before multiplying them by that coefficient.

Standard FIR filters can also be constructed as well as odd-symmetrical, decimating, and interpolating FIR filters. In this example, we are performing interstage registration of the adder sections. By micropipelining, data-rates approaching 100MSPS (mega-samples per second) are practical.

Symmetrical 8-Tap FIR Filter (FIR8S)



In the table below, we describe the performance specs for an 8-tap FIR macro; using this macro as a slice, the 16-tap and 24-tap versions were produced.

Symmetrical FIR Filters Specifications (Using Vector Multiplier)

Sample Implementation	8-tap symmetric	16-tap symmetric	24-tap symmetric
Data Format	8-bit unsigned	8-bit unsigned	8-bit unsigned
Coefficient Format	8-bit unsigned	8-bit unsigned	8-bit unsigned
Output Format	17-bit unsigned	10-bit unsigned	10-bit unsigned
Pipelining	NO	NO	NO
Speed (-1 speed grade)	31 MHz	31 MHz	31 MHz
Device & Utilization	20% Logic AT40K30	40% Logic AT40K30	44% Logic/Device AT40K40

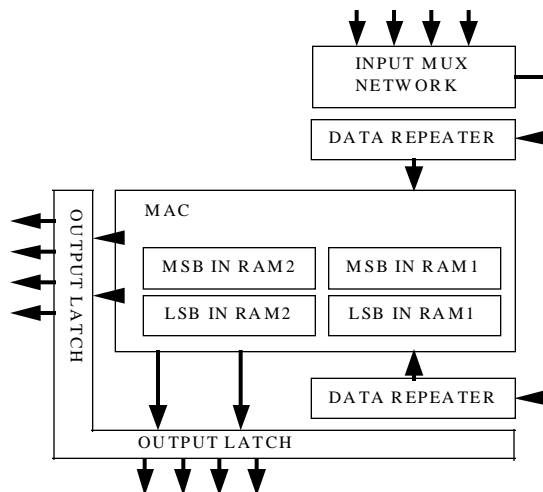


3-TAP FIR System Using High-Speed MAC Core

For the design of high-speed processing units wherein the arguments to the functions must remain variable at all times, a multiply-accumulator (MAC) macrofunction has been developed. The MAC core is a hybrid of a conventional parallel multiplier array and an accumulator chain. Coefficients are stored in dual-port RAM modules that are distributed within the FPGA sectors. As a sample application, the figure below shows the block diagram of a 3-tap FIR system using a MAC core that contains three stages of registration. Two 32x4 RAM modules are used to accommodate the 8-bit coefficients, one for the LSBs and the other for the MSBs. The MAC core is designed to facilitate the construction of various algorithm execution units. Although pipelined, the skewing and deskewing registers have been replaced by a dedicated flushing mechanism. There is a three clock penalty associated with flushing. As convolution cycles are increased, the penalty overhead becomes minimal. However, even for simple video convolvers, full video rates are achieved.

The MAC core and associated storage/control system can run at 85Mhz. This permits its use in real-time video applications. By using three of these blocks, a 2-D 3x3 convolution kernel can be realized. An important aspect of this design is that no optimizations have been made by assuming certain coefficient values or symmetry. This system may be easily upgraded to a 6-tap symmetric FIR system by pre-adding the incoming video information corresponding to the coefficients.

In some applications, the precision of the coefficients can be reduced to as low as four bits and maintain an acceptable filter response. This leads to significant reduction in device utilization as the number of RAM modules, multiplier cells, and accumulator cells decrease. The number of interstage registration also decreases accordingly and the unit still meets the video rate standards (e.g. composite video).



3-Tap FIR example using high-speed MAC

3-TAP FIR System Specifications (using High-Speed MAC Core)

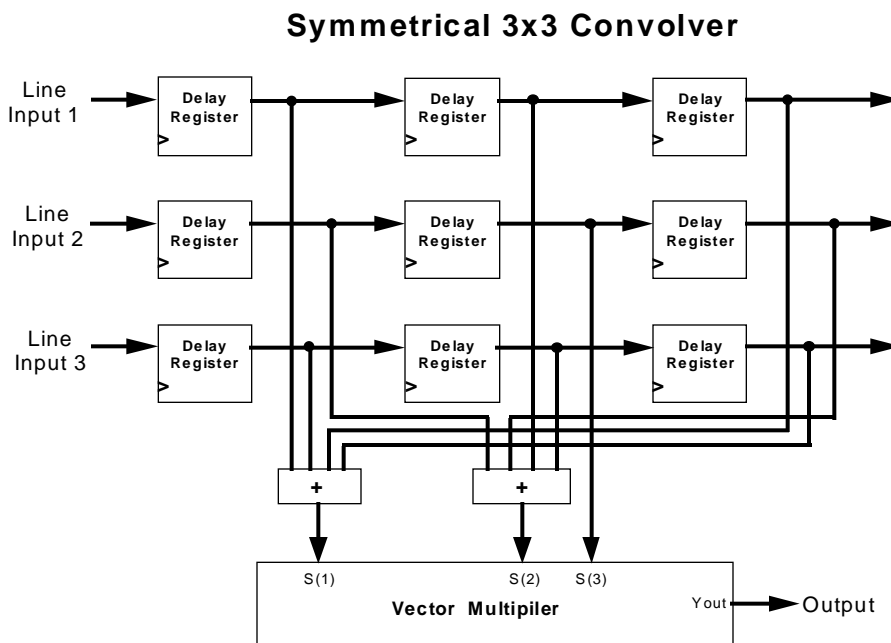
Sample Application	1-D 3-tap FIR
Data Format	8-bit unsigned
Coefficient Format	8-bit unsigned
Output Format	17-bit unsigned
MAC Structure	3-stage registered
Flush Cycle	3
System Clock Frequency (Typical Delay)	84.3 MHz
Throughput	14.05 MSPS
Device Utilization AT40K10-1AC (-1 Preliminary)	40% Logic 11% RAM 24% I/O

Notes:

- The above system meets the 13.5 MHz composite video rate.

Imaging Applications

For many imaging applications, 2-D filtering is an essential processing step. Two basic approaches to 2-D convolution are presented here. First, an optimized version that assumes constant symmetrical convolver masks, such as Sobel masks used in edge detection. The symmetrical 3x3 kernel shown below demonstrates the hardware optimization possible. The symmetry of the convolver masks enable the same kind of optimization possible with 1-D FIR filters.

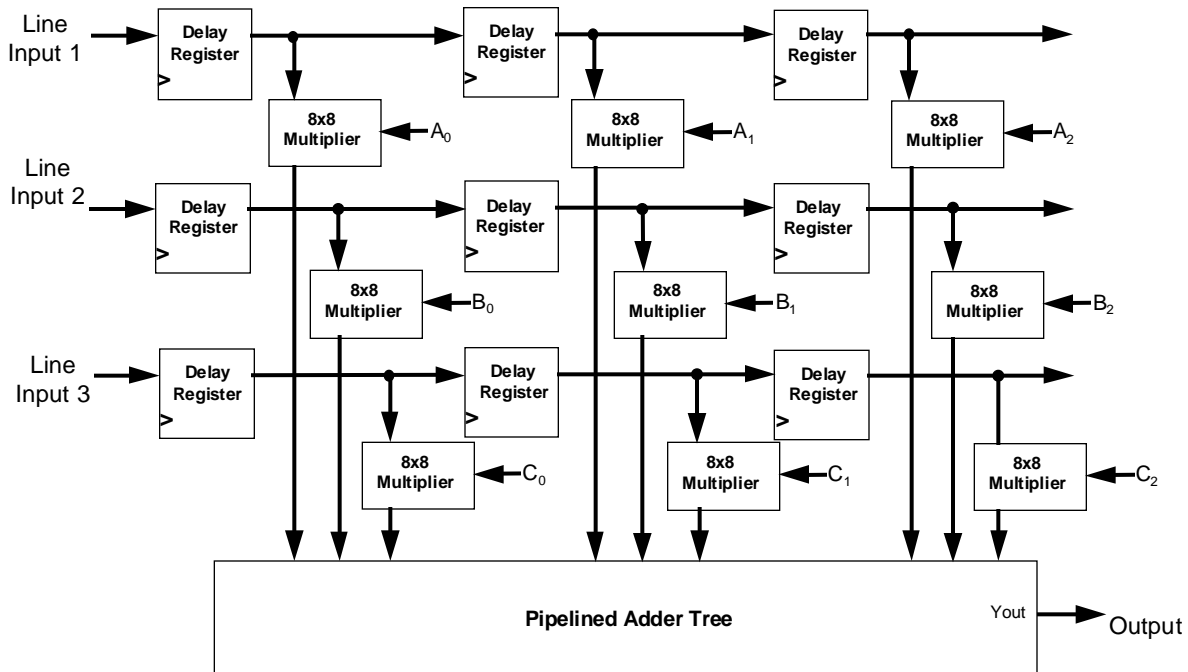


2-D Convolver optimized for convolution mask symmetry and constant coefficients
Real-time video rate performance
8-bit inputs & 8-bit coefficients

The pixel data values associated with the symmetrical mask are pre-added before “multiplication.” Further optimization is achieved by using a vector multiplier instead of real parallel multipliers. The vector multiplier is similar to the type used in the high-speed digital filters discussed above.

While the above approach yields excellent performance and device utilization, it is inflexible from an application perspective. In applications like special effects video processing and real-time graphics manipulation, a fully-programmable 2-D convolver is much more desirable. The AT40K FPGA with its optimized datapath architecture enables the design of fully programmable 2-D convolution kernels that use true multipliers to provide the ultimate flexibility. Shown in the figure below, each stage allows for user programmable coefficients to be modified in real-time.

Fully-Programmable 3x3 Convolver



Fully programmable 2-D convolver using real multipliers for total application flexibility
 Real-time video rate performance
 8-bit inputs & 8-bit coefficients

Applications:

- Noise removal (LPF)
- Radar
- Target Recognition
- Contour mapping

- Edge emphasis (HPF)
- Robotic vision
- MRI
- Special Effects

Performance:

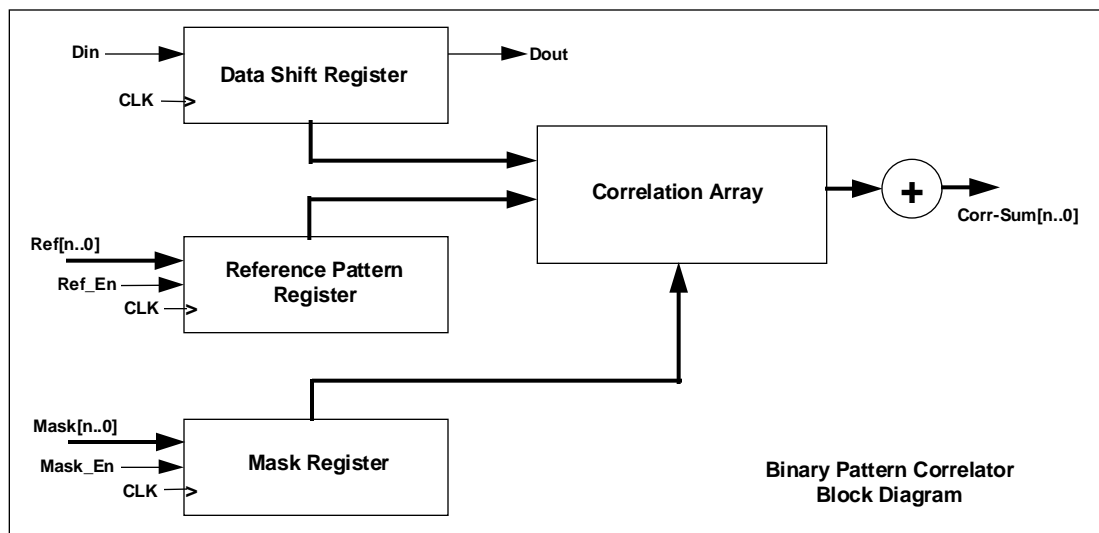
- Real-time video rates

- Computer graphics (VGA) rates

2-D Fully Programmable 3x3 Convolver Specifications

Sample Implementation	3x3 convolver
Data Format	8-bit unsigned
Coefficient Format	8-bit unsigned built-in
Output Format	20-bit unsigned
Pipelining	NO
Core Frequency	20 MHz
Device Utilization AT40K40-1SC	33% Logic 0% RAM 0% I/O

Binary Pattern Correlator



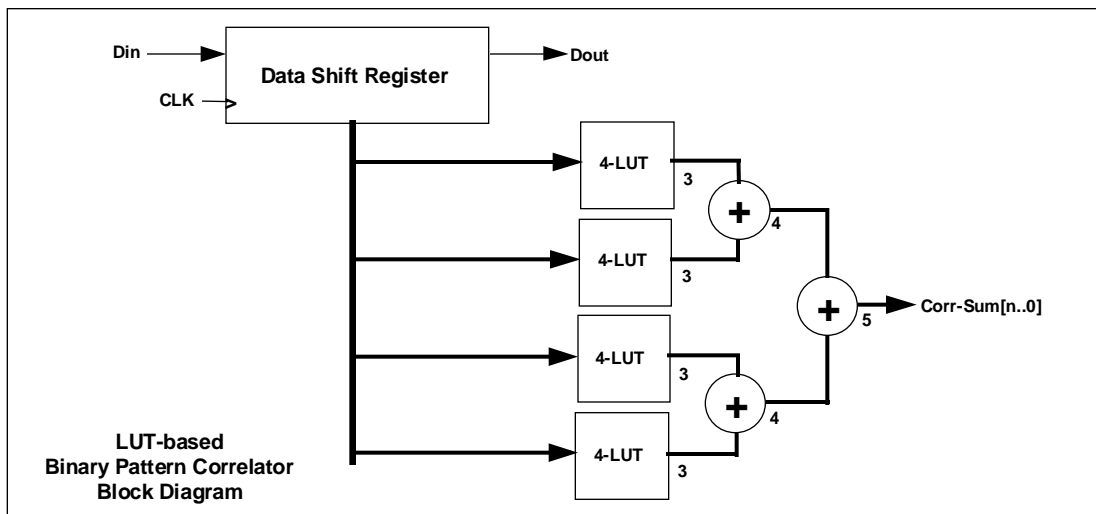
- Optimized for AT40K FPGA device architecture
- Parallel correlation network for high speed
- Programmable reference pattern and mask registers
- Cascadable in length and depth
- Applications Include:
 - Frame synchronization
 - Spread-spectrum receivers
 - Error correction
 - Pattern Matching

Correlator Functional Description

The binary pattern correlator macrofunction is a digital correlator that compares the digital pattern stored in the reference pattern register with the data samples stored in the correlator shift register. The macro contains the following components: a data shift register, a reference pattern register, a mask register, the correlation array and the correlation summing tree.

To determine the number of matches in a data stream, the macrofunction shifts the data samples into the data shift register, where they are compared with the data stored in the reference pattern register. The number of matches is calculated on each rising edge of the clock, the final sum is called the correlation sum. The correlation sum is usually compared to a programmable threshold. The threshold determines the probability of detection and the false alarm rate. A lower threshold increases the probability of a detection but it can also increase the probability of a false alarm. For example, if the shift register and the reference pattern are 32 bits in length, then a threshold of 32 requires a perfect match with no false alarms. However, a threshold of 32 does not permit a single bit error. Most systems are required to be tolerant of a small number of errors. Therefore, a threshold of 31 provides a bit error tolerance of about 3%. As the pattern length is increased, the bit-error detection resolution can be increased as well.

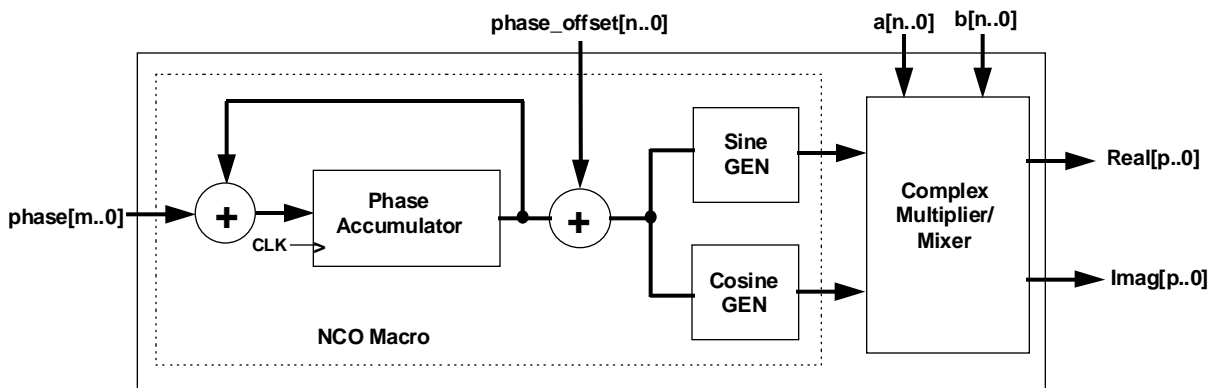
LUT-based Binary Pattern Correlator



- Optimized for AT40K FPGA device architecture
- Parallel LUT-based correlation network for maximum speed
- Reference pattern and thresholds determined by LUT contents
- LUT contents modifiable by dynamic reconfiguration
- Cascadable in length and depth
- Applications Include:
 - Frame synchronization
 - Error correction
- Spread-spectrum receivers
- Pattern Matching

The LUT-based correlator works in a similar manner to the fully programmable version described above. However the reference pattern and partial correlation sum values are stored in the LUT. With this type of correlator, the bit-error detection thresholds can be scaled within the look-up table. To update the reference pattern or change the threshold, new values are loaded into the LUTs. This can occur while the device is in operation through the dynamic reconfiguration mechanism of the AT40K.

Digital Modulator



Component Blocks:

- Configurable NCO (Numerically-Controlled Oscillator)
- Parameterizable Sine/Cosine Generators
- Parameterizable Multiplier/Mixer
- Two's Complement Arithmetic Format

Applications:

- Amplitude Modulation
- Frequency Modulation
- Phase Modulation
- Down Converters
- Direct Digital Synthesis

Digital Modulator Functional Description

The digital modulator macrofunction consists of a numerically-controlled oscillator (NCO) with quadrature outputs and a parameterized complex multiplier/mixer. The NCO uses an accumulator-divider that can range from 2 to 32 bits in resolution. The phase offset is achieved by using an adder of any required resolution. The sine and cosine generators consist of look-up tables and process logic, which are parameterized in terms of resolution and precision. The complex multiplier/mixer is built from standard AT40K multiplier and adder components and computes real and imaginary outputs from the two user inputs and the NCO outputs. All these modules are parametrically-generated by Atmel's Component Generator tool for maximum user-flexibility.

This macrofunction can generate a variety of modulations from the basic types: AM (amplitude modulation), FM (frequency modulation), and PM (phase modulation). For AM, the NCO is set to the desired frequency, the modulating signal is applied to the a-input, and the b-input is set to zero. Binary data can be applied to the MSB of the a-input, creating ASK (amplitude shift keying). Quadrature amplitude modulation (QAM) is attained by using the a and b inputs to apply a complex vector to be modulated. FM is achieved by varying the phase input to the NCO. The modulating source can be added to the center frequency data at the phase input. The frequency modulation range can be controlled by scaling the input data. Binary data controlling a multiplexer that switches between two phase words permits frequency shift keying (FSK).



Conclusion

Unlike DSP processor chips, FPGAs can compute signals in parallel. Many designers choose programmable devices over ASIC because of their adaptability. In an evolving field like DSP, algorithms and processing techniques are changing continuously. With a programmable part on the board, a change can be made in hours, and the design can be recompiled in minutes, and downloaded in seconds. By allying a DSP with a reconfigurable FPGA, the convenience of firmware can be combined with the flexibility of reconfigurable hardware yielding a powerful resource that can be targeted for a variety of applications.

Reconfigurable FPGAs provide new approaches to used in signal processing systems, products, and board designs. Reconfiguration is emerging as an integral system concept enabling a new class of "virtual products" wherein "soft" hardware updates are now possible along with software/firmware updates. Reconfigurable FPGAs can perform multiple DSP tasks as needed, instead of merely being used for fixed functions. Product life cycles can be extended long past what is normally considered possible by adding new features in the FPGA hardware via configuration updates. By employing DSP macrofunctions, engineers can meet performance specifications, shrink design cycles, and decrease time to market. These unique abilities allow AT40K FPGAs to bring exciting possibilities for the world of DSP.

FOR MORE INFORMATION:

Click here for a 25-page summary of the AT40K Data Sheet or to download the full 52-page data sheet

If your Reader does not support direct links, use your web browser and go to one of the URLs shown below.:

(<http://www.atmel.com/acrobat/doc0896.pdf>) [Summary Data Sheet](#)
(<ftp://www.atmel.com/pub/atmel/at40K.exe>) [Full Data Sheet](#)