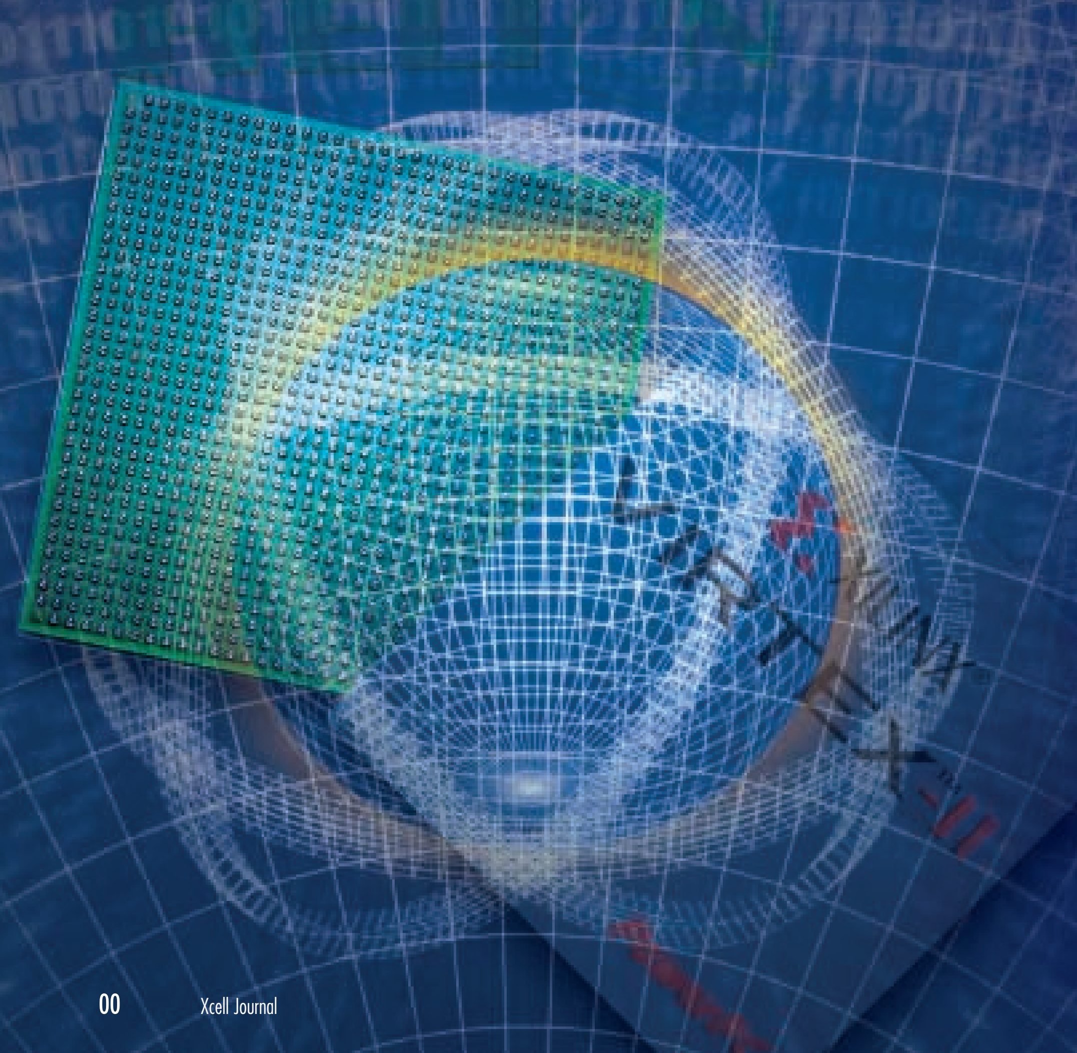
How to Use Virtex-II FPGAs to Deliver Gigabit Advanced Data Security with Helion Encryption Cores

XPERTS
Xilinx Program for Engineering Resources from Third Parties

Virtex-II Platform FPGAs enabled Helion Technology Ltd. not only to deliver on time but to exceed the client's requirements.



by Simon Cocking
Design Consultant
Helion Technology Limited
simon@heliontech.com

As secure communication data rates increase, software implementations of the key data encryption algorithms present a major system bottleneck. To make matters worse, standard off-the-shelf CPUs and DSPs have failed to keep pace with the computational demands of data encryption algorithms. Besides, CPUs and DSPs just have too many other tasks to perform.

The magnitude of the software performance gap for high-speed data encryption is illustrated by the benchmark results shown in Table 1. These figures are for implementations of the new 128-bit Advanced Encryption Standard (AES) algorithm running on standard CPU and DSP architectures – as compared to how a Virtex™-II Platform FPGA solution based on a Fast AES core from Helion Technology Ltd. can provide the multi-gigabit encryption data throughput required by high-speed interconnect technologies.

The Design Brief

Last year, we at Helion were asked by one of our clients to provide a flexible, high performance FPGA-based data security engine to handle encryption of Internet Protocol (IP) packet-based data for their next-generation network security products. Even though their product was based on a high-performance dual 64-bit MIPS network processor, benchmarking had shown a software encryption solution to be somewhat short of achieving the minimum 600 Mbps throughput our client required.

The system requirement was for a PCI-based engine that could accommodate multiple encryption algorithms: AES and a proprietary algorithm for legacy purposes. The solution had to be low cost, capable of supporting gigabit data rates, and flexible enough to allow for future algorithm changes.

After evaluating the available options, the only solution that met all of the requirements was a PCI card containing the one million-gate Virtex-II XC2V1000 Platform FPGA. The device incorporated a Xilinx LogiCORE™ 32-bit/66-MHz PCI core and Helion-designed encryption cores.

Why Use a Virtex-II FPGA?

From our many years of design experience with leading edge technology, we were well aware of the potential pitfalls, such as inadequate design tool support and unavailable parts. Fortunately, Virtex-II Platform FPGAs use the same EDA toolset as earlier technologies, and the XC2V1000 device was already in production. The reasons for choosing the XC2V1000 instead of the more established Virtex-E family were compelling:

- Cost – The Virtex-II XC2V1000 cost less than half the nearest equivalent Virtex-E(-8 speedgrade) device.
- Performance – Benchmarking with the Helion AES core showed the Virtex-II(-4) FPGA to be 30% faster than the nearest equivalent Virtex-E(-8) device.
- Bigger and wider block RAM – We found 32-bit wide data buffers were more efficiently implemented in Virtex-II block RAM. For example, a 512x32 buffer required only a single Virtex-II block RAM rather than four in Virtex-E technology.
- Enhanced CLBs – The MUXF7 and MUXF8 primitives allowed up to four slices to be combined for fast implementations of any logic function up to eight inputs wide. This led to fewer logic levels and faster critical paths for the very wide logic functions typical of encryption algorithms.
- Digital Clock Manager (DCM) – True clock frequency synthesis allowed the encryption data clock to be tuned to closely match worst-case PAR (place-and-route) timing, optimizing encryption throughput.
- Easier Design Implementation – The raw speed of Virtex-II devices meant no PAR guide file was required for the LogiCORE 66-MHz PCI bus logic.

Encryption Engine Overview

A block diagram of the encryption engine is shown in Figure 1. The only external interface is the LogiCORE 32-bit/66-MHz PCI bus.

All encryption key information and data to be encrypted (plaintext) are stored in PCI system memory and transferred into the FPGA by the direct memory access (DMA) controller. This configuration allows the encryption keys to be easily changed by software – an important security feature.

The DMA read-and-write buffers use block RAM to support the long PCI burst read-and-write transfers needed to achieve optimum performance. Checksum calculation and buffering is also provided for the encrypted data (ciphertext) to off-load this task from the system software, which reads the checksum(s) at the end of each DMA transfer.

The main data path between the DMA buffers is through the encryption cores. The Helion 128-bit fast AES encryption core had already been developed, so it was simply a case of dropping it into the design. However, it was still necessary to develop a high-performance core for the proprietary algorithm.

Encryption Engine Operation

While the encryption engine is inactive (no DMA encrypt transfer in progress), the PCI interface defaults to "target" mode and waits for the system CPU to initiate a DMA transfer. Once a DMA operation is requested, the PCI interface is switched to "master" mode, and the DMA controller initiates the required bus transfers.

Upon completion, the encryption engine asserts the PCI interrupt request and returns the interface to "target" mode so that the system CPU can read the checksums and/or start the next transfer.

Solution	Clock (MHz)	Encryption Data Rate	Comments
TMS320C62XX 32-bit fixed-point DSP	200	112 Mbps	
MIPS-based 64-bit RISC processor	250	392 Mbps	Assumes fully primed cache and CPU fully dedicated to encryption
Pentium III	1,000	464 Mbps	Assumes fully primed cache and CPU fully dedicated to encryption
Helion Fast AES Core	132	1536 Mbps	Virtex-II 2V1000-4 target

Table 1 - Typical achievable data rates for 128-bit Advanced Encryption Standard solutions

Because the plaintext data stream written into PCI system memory can be fragmented into multiple IP packets, the system software is responsible for creating a control data structure that details the size, location, and number of fragments, as well as the encryption type to use. This structure is transferred from memory into a block RAM in the DMA controller at the start of each encrypt transfer.

The encryption engine can then read in all plaintext fragments, encrypt them using the selected algorithm, and reassemble the ciphertext fragments in memory. In the process of writing ciphertext back to memory, a checksum is calculated for each encrypted fragment, and this value is added to a total checksum of all fragments. These checksums are stored in a small distributed RAM in the FPGA, and then read back by the system CPU at the end of each transfer.

Encryption Engine Performance

Our client was delighted to learn that not only was the final encryption engine design delivered on time, but that it also exceeded the required data throughput for both encryption algorithms by a significant margin.

The proprietary algorithm core is clocked at 66 MHz to achieve a raw encryption throughput of 990 Mbps. The AES core is clocked at 132 MHz to yield a raw encryption throughput of 1.536 Gbps.

However, these raw throughput figures are degraded somewhat in the final system due to bus arbitration overhead in the PCI bridge. In fact, the data throughput of the Helion fast AES core is so high that the maximum available bandwidth on the 32-bit/66-MHz PCI bus acts as a ceiling on its performance.

Frequent changing of the encryption keys will also lead to a reduction in overall data encryption throughput due to the increased transfer load on the PCI bus.

Conclusion

The encryption engine project we've described is a great example of how a combination of third-party intellectual property cores and Virtex-II Platform FPGA technology can yield flexible, high-performance security products in record time.

A large reduction in the effort required to implement such a design relies on the ready availability of intellectual property like the Xilinx LogiCORE PCI products and the range of advanced encryption cores we have developed at Helion. Design blocks like these enable a single engineer to attack a million-gate design and achieve results in a matter of weeks.

Future plans for our encryption engine include the use of a higher bandwidth bus interface, such as HyperTransport™ technology in place of PCI. We also plan the addition of multiple AES cores to provide a full-duplex (encrypt and decrypt), higher performance solution.

The Helion encryption cores allow for even higher throughput: Eight similar Fast AES cores in a larger Virtex-II Platform FPGA could provide encryption at rates in excess of 12 Gbps. This is fast enough to support emerging high-speed interconnect standards, such as OC-192, 10 Gigabit Ethernet, POS-PHY L4, and RapidIO™ Phy.

For more detailed information on Helion and our products and services, please visit the Helion website at *www.heliontech.com.*

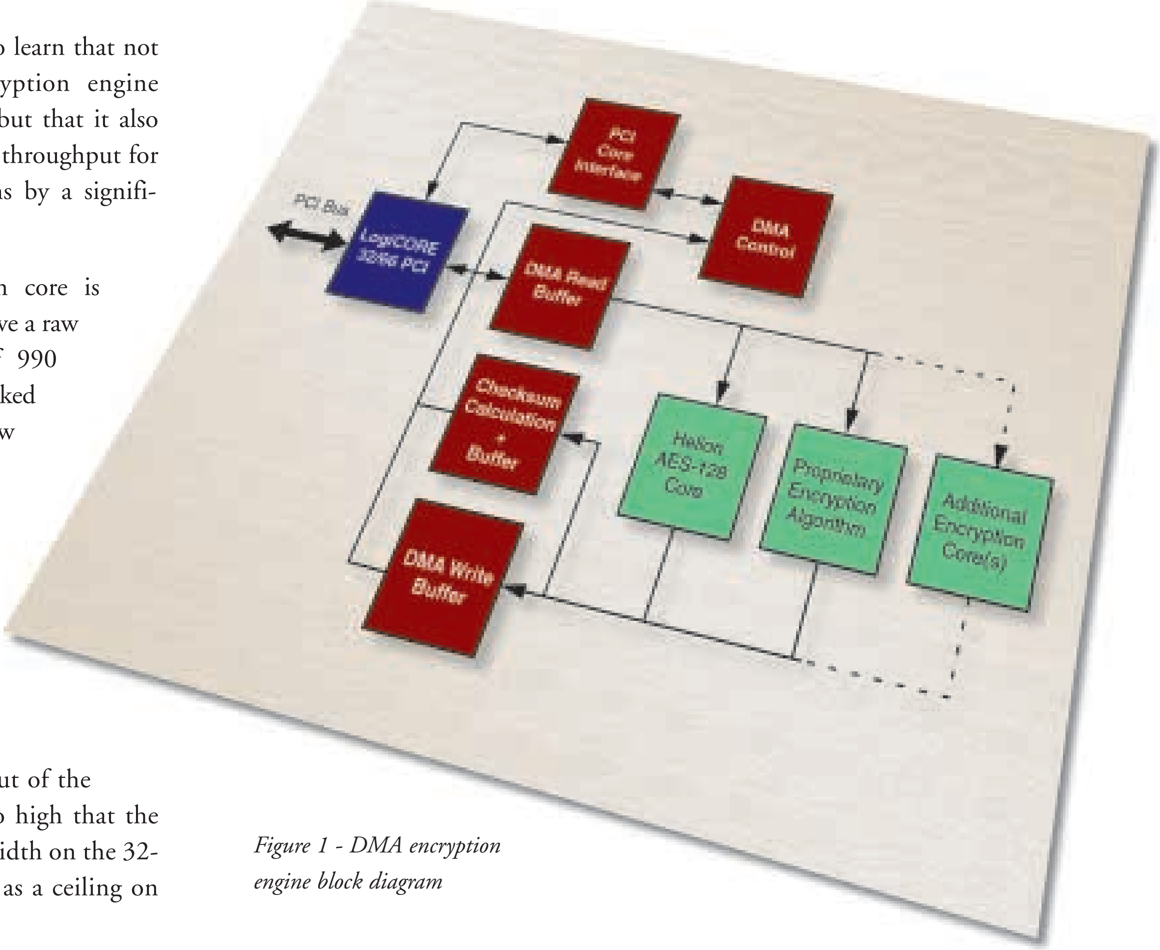


Figure 1 - DMA encryption engine block diagram