Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

ARC ACCELERATES CRYPTOGRAPHY

New Processor Extensions and Software Boost AES and DES

By Tom R. Halfhill {9/22/03-01}

ARC International is offering a new package of microprocessor extensions and security software that can improve the performance of common cryptographic algorithms by an order of magnitude or more. The package, called ARCprotect, includes new instructions, registers, and middleware as licensable intellectual property (IP) for the current ARCtangent-A5 and future ARC microprocessor cores.

The extension instructions focus on the popular AES, DES, and 3DES (triple DES) encryption/decryption algorithms that are the foundation of many network-security protocols, such as IPSec (Internet Protocol Security), IKE (Internet Key Exchange), SSL (Secure Socket Layer), TLS (Transport Layer Security), and Encrypted PPP (Point-to-Point Protocol). They also accelerate the MD-5 and SHA-1 secure hashing algorithms.

ARC's new IPShield security software, an optional part of the ARCprotect package, works with the company's RTCS TCP/IP stack for real-time embedded systems. The IPShield software and TCP/IP stack are also available for the ARM, MIPS, Motorola ColdFire, and IBM/Motorola PowerPC architectures. However, the ARCprotect extensions are specific to the ARCtangent microprocessor cores. All the software and synthesizable IP are shipping now.

Small Extensions, Big Results

One advantage of a configurable microprocessor is that the vendor can introduce optional extensions at any time without disturbing the architecture or burdening it with features that some customers may not want. With a conventional microprocessor, new extensions (such as the MMX or SSE extensions for the x86) become a fixed part of the architecture that all future processors must support for backward compatibility with software. Customers are stuck with the extra hard-wired logic and power consumption of the extensions, whether they want the new features or not. Over time, the opcode map of a conventional architecture fills up with instructions that may become redundant or obsolete.

Not so with a configurable microprocessor architecture. For instance, ARC and archrival Tensilica offer optional DSP extensions for their configurable processor cores. The DSP extensions include additional instructions, registers, and on-chip XY data memories. SoC developers—the customers that license the cores from ARC or Tensilica—can choose to add those features to the processor by clicking a few buttons in a graphical configuration tool. Customers who don't need the DSP extensions don't have to add them.

The new ARCprotect extensions offer similar flexibility. Licensed as soft IP separately from the ARCtangent-A5 processor core, they add four instructions, 13 logical core registers, and 10 auxiliary registers to the programmer's model. To obtain the best performance, ARC also recommends adding the DSP extensions and a barrel shifter—existing options in the company's ARChitect processor-configuration tool. The ARCprotect extensions alone add about 30,000 gates, and the DSP extensions and barrel shifter add another 2,000 gates, plus 30,000 gates for the XY memory. (The minimal base configuration of the ARCtangent-A5 core is about 16,000 gates.) However, developers can omit parts of the extensions not required for their specific application, thereby saving some silicon.

Of the four instructions in the ARCprotect package, only three are actually new. ARC introduced the instruction that accelerates DES cryptography (desrnd) in 2001. The genuinely new instructions are aptly named aes, aessb, and hash, because they accelerate AES cryptography and secure hashing algorithms.

To support the extension instructions, ARCprotect also adds new registers to the ARCtangent processor. Some are core registers, which in ARC nomenclature are peers of the standard ARCtangent register file. Like most RISC architectures, ARCtangent has 32 general-purpose registers, 32 bits wide. The ARCtangent architecture allows a total of 64 core registers, leaving room for SoC developers and ARC's own engineers to add new extension registers that work just like the standard set of core registers. ARCprotect uses five of those extension-register addresses, along with a clever mode-switching scheme that makes them appear as 13 logical registers.

In addition, ARCprotect defines 10 new auxiliary registers. In the ARCtangent architecture, auxiliary registers are optional 32-bit registers accessible in a single clock cycle by special load/store instructions. Their addressable range is 32 bits, so it's theoretically possible to define more than four billion auxiliary registers, although, in practice, the synthesized datapaths would be unable to maintain single-cycle access with that much memory on chip. ARCprotect uses its 10 new auxiliary registers to control the core-register mode switching mentioned above and to store the intermediate results of AES and hashing operations. By saving and restoring the intermediate state information in the auxiliary registers, a multithreaded real-time operating system (RTOS) can switch contexts and encrypt or decrypt multiple datastreams simultaneously.

Table 1 lists all the new ARCprotect instructions and registers. As we'll explain below, many of the core registers perform multiple duties, depending on the mode selected by the auxiliary control registers.

Instruction	Description	Comment
aes	Perform half-round AES	3 cycles per round
aessb	AES sub-bytes transform	AES key expansion
desrnd*	Perform DES round	1 cycle per round
hash	Perform half-round SHA-1 or full-round MD-5	1 cycle (SHA-1) or 2 cycles (MD-5)
Core Registers (32b)	**Description**	**Comment**
aes0 (r50)	AES state for matrix column 0	4 x 4-byte matrix
aes1 (r51)	AES state for matrix column 1	4 x 4-byte matrix
aes2 (r52)	AES state for matrix column 2	4 x 4-byte matrix
aes3 (r53)	AES state for matrix column 3	4 x 4-byte matrix
des_c (r50)	1st half of 56-bit DES key	28 bits in 32-bit reg
des_d (r51)	2nd half of 56-bit DES key	28 bits in 32-bit reg
des_l (r52)	1st half of 64-bit DES data	Left 32-bit word
des_r (r53)	2nd half of 64-bit DES data	Right 32-bit word
rA (r50)	Hash context (word 0)	Digest result of hash
rB (r51)	Hash context (word 1)	Digest result of hash
rC (r52)	Hash context (word 2)	Digest result of hash
rD (r53)	Hash context (word 3)	Digest result of hash
rE (r54)	Hash context (word 4)	Digest result of hash
Aux Registers (32b)	**Description**	**Comment**
crypt_mode	Select crypto/hashing mode	Select AES key size
arith_ctl	Select DES/AES-hash mode	Switch reg visibility
aux_aes0	AES engine state	For context switch
aux_aes1	AES engine state	For context switch
aux_aes2	AES engine state	For context switch
aux_aes3	AES engine state	For context switch
aux_aes4	AES engine state	For context switch
des_aux	DES ID register	Extension version
AuxS	Hashing engine state	For context switch
AuxI	Hashing engine state	For context switch

Table 1. The ARCprotect extensions are an efficient example of adding custom instructions and registers to a configurable microprocessor core. Note how ARC maps 13 core-register logical addresses to five physical addresses. *ARC introduced the DES instruction in 2001.

How the Instructions Work

The two AES instructions—aes and aessb—operate on 128-bit blocks of data stored in four 32-bit core registers, named aes0, aes1, aes2, and aes3. ARCprotect supports AES cipher keys that are 128, 192, or 256 bits long. The length of the cipher key determines the number of rounds of calculations required to encrypt or decrypt the data: 11 rounds for 128-bit keys, 13 rounds for 192-bit keys, and 15 rounds for 256-bit keys. (Officially, the AES specifications don't count the initial setup calculation as a round, as ARC does, so the actual encryption or decryption requires 10 rounds, 12 rounds, and 14 rounds, respectively.)

Before performing these rounds, the processor must expand the original cipher keys, using an algorithm in the AES specifications. Because each key must be expanded only once, ARC decided not to implement the whole algorithm in hardware. However, the aessb instruction does accelerate the algorithm by performing a transformation known as the SubWord() function. This function consists of four table lookups, one for each byte of the 32-bit source operand. The aessb instruction stores the 32-bit result of this function in a core register.

Because the expanded AES keys are extremely long—1,408 bits for 128-bit keys, 1,664 bits for 192-bit keys, and 1,920 bits for 256-bit keys—they cannot fit in the core registers, and ARC decided not to allocate any auxiliary register space for them. Instead, ARC recommends adding the optional DSP extensions to the ARCtangent processor so it can store the expanded keys in on-chip XY data memory. With a 64-bit datapath to XY memory, the processor can fetch two 32-bit words of the expanded key per clock cycle. The AES data block stored in the core registers is 128 bits long, so the processor must execute the aes instruction twice to perform one full round of calculations.

The 128-bit data block appears as a virtual 4- x 4-byte matrix in the AES algorithm. Each round performs a series of exclusive-OR transformations on each column in the matrix. ARC's technical documentation specifies a layout for the expanded key that takes advantage of the dual XY banks of memory in the DSP extensions, so the aes instruction can apply the key to the matrix with maximum efficiency.

The new hash instruction accelerates the SHA-1 and MD-5 hashing algorithms frequently used during encryption. One instance of the instruction performs a half-round of SHA-1 hashing in one clock cycle, or a full round of MD-5 hashing in two clock cycles. The hash instruction stores the results of each round—known as the digest results—in five newly defined core registers. Programmers can select between SHA-1 or MD-5 hashing by storing a specified value into the new crypt_mode auxiliary control register. Writing other values into this register also specifies the length of the AES cipher key.

Although the desrnd instruction isn't new, it's a logical addition to the ARCprotect extension package. The DES algorithm requires 16 rounds of calculations for each pass through the data that must be encrypted or decrypted, and the desrnd instruction performs one round in a single clock cycle. The more-secure 3DES algorithm requires multiple passes. DES data blocks are 64 bits long, only half the length of AES data blocks, and the DES key is only 56 bits long. Therefore, the processor needs only two 32-bit core registers to store the data block and two more registers to store each half of the key—four new registers in all.

Mode-Switching Saves Resources

As Table 1 shows, the ARCprotect extensions define 13 new core registers, but all are mapped to the same five register addresses, r50–r54. To conserve register slots and silicon, ARC decided to share the same physical registers among the four instructions in the package. This makes sense, because it's unlikely that a program would perform these different cryptographic operations at the same time in the same thread of execution. (As mentioned above, a multithreaded RTOS could perform multiple operations simultaneously by saving and restoring the new register state.)

To prevent register conflicts, programmers choose which logical registers are visible by writing specified values into the control registers, which ARCprotect implements as auxiliary registers. For example, the aforementioned crypt_mode register selects between SHA-1 and MD-5 hashing and specifies the AES key size. It also determines whether the visible core registers are the four AES registers (aes0–aes3) or the five hash registers (rA–rE). Other bits in crypt_mode indicate whether the processor supports the AES instructions in hardware and, if so, the version of that hardware.

The crypt_mode register works in concert with the other auxiliary control register, arith_ctl. Writing specified values into this register also determines whether the visible core registers are the AES registers, the hash registers, or the DES registers (des_c, des_d, des_l, des_r). Another auxiliary register, des_aux, indicates whether the processor supports the DES extensions in hardware and, if so, the version of the hardware.

Custom Extensions Boost Performance

Adding a few custom instructions to a configurable processor often results in eye-popping performance. For instance, the desrnd instruction accelerates DES processing 47 times and 3DES processing 92 times. To get the same improvement by applying brute force without the desrnd instruction, an SoC designer would have to crank up the clock frequency of an ARCtangent-A5 processor to about 14GHz—clearly unattainable with existing technology.

A major reason for the dramatic performance improvement is that the custom instructions greatly reduce the number of memory accesses by carrying out more operations in the registers. For example, a DES encryption routine that performs 16 rounds using the standard ARCtangent instruction set would normally access memory 176 times (168 reads and 8 writes), not counting the additional accesses required to set up the 16-element key table. With the desrnd instruction, the processor must access memory only four times (two reads and two writes) to perform the same 16 rounds. Other operations drop from as many as 3,677 cycles to 16 cycles.

Unfortunately, ARC hasn't released specific performance estimates for the other instructions in the ARCprotect package, other than to say that IPSec security algorithms will run "an order of magnitude" faster—an impressive (10×) improvement.

Another happy consequence of using custom instructions is denser code. A single new instruction might do the same work as dozens of standard instructions. For example, implementing the low-level DES routine on the base configuration of an ARCtangent processor normally requires about 6.5K of executable code, using standard instructions. With the desrnd instruction, the same routine collapses into as little as 2.4K of code—a reduction of about 63%.

There is a price to pay in silicon for the extra hardware, of course, but it's relatively small in view of the dramatic performance improvement and the transistor densities of deep-submicron fabrication technology. Even after bolting on the ARCprotect and recommended DSP extensions, the ARCtangent-A5 core weighs in at perhaps one square millimeter of silicon in a 0.18- or 0.13-micron process. If the SoC contains any significant amount of on-chip memory and peripheral logic, the processor core will occupy only a small corner of the die.

Encapsulating an algorithm in a custom instruction is also a way of adding a measure of security. The algorithm is hard-wired into silicon that can't be disassembled like

Price & Availability

The ARCprotect extensions and IPShield software are available now. A single-product license for the IPShield software alone is $30,000. ARC does not publicly disclose up-front license fees and royalties for its synthesizable microprocessor IP. For more information, see *www.arc.com*.

software and is virtually opaque to hackers. The entire algorithm might appear in the software as a single opcode that is not part of the standard instruction set. However, that advantage doesn't matter in this case, because the AES, DES, SHA-1, and MD-5 algorithms are public information, not proprietary IP. Indeed, a principle of good cryptography is that an encrypted message should remain secure even if the encryption method is known. Strong encryption relies on mathematical security and key integrity, not obscurity.

The ARCprotect extensions—and the IPShield software that uses the extensions to accelerate the IPSec protocol in ARC's TCP/IP stack—are worthy additions to the ARC IP library. They expand the options available to SoC designers who need fast, secure networking in embedded systems. ◇