ARM PROCESSORS IMPLEMENT A STORE ARCHITECTURE

Dr. Syed Musthak Ahmed*

Dual processor systems also gain from a general decline in latency. Simply put, while there is no current way to share the current operating system load evenly between two processors, the second processor can step in and keep the system running smoothly while the first is maxed out to 100% burning a CD or encoding a file (or from a software error). Obviously, if dual-core systems become main stream, which it looks like they are going to, future operating systems and applications will be designed with the feature in mind, leading to better functiona4lity down the road. Thirdly, and less obviously, AMD and Intel are desperate. Both companies have run into barriers when it comes to increasing the raw speed of processors, or decreasing the die size. Until these roadblocks are cleared or until the general buying public understands that GHz does not directly translate to performance, both companies will be scrambling to discover any new improvements that will improve processor performance... without actually boosting core speed. This is why the idea of dual-core processors is now a reality.

ARM is the leading provider of 32-bit embedded RISC mi-croprocessors with almost 75% of the market. ARM off ers a wide range of processor cores based on a common architec-ture [9] [4], delivering high performance together with low power consumption and system cost.

ARM processors implement a load/store architecture. De-pending on the processor mode, 15 general purpose registers are visible at a time. Almost all ARM instructions can be executed conditionally on the value of the ALU status flags. Load and store instructions can load or store a 32-bit word or an 8-bit unsigned byte from memory to a register or from a register to memory.

* Professor and HOD, SR Engineering College, Warangal,AP,India

IJMRA

The ARM arithmetic logic unit has a 32-bit barrel shifter that is capable of shift and rotate operations. The sec-ond operand to all ARM data-processing and single register data-transfer instructions can be shifted before data process-ing or data transfer is executed, as part of the instruction. The amount by which the register should be shifted may be contained in an immediate field in the instruction, or in the bottom byte of another register. When the shift amount is specified in the instruction, it may take any value from 0 to 31, without incurring any penalty in the instruction cycle time.

Use of word extended substitution tables in Rijndael im-plementations is unnecessary and inefficient on ARM proces-sors, since the architecture supports load byte instructions. Use of pre-rotated tables cannot improve the performance neither, since the barrel shifter that can be combined with data processing instructions reduces the effective cost of ro-tate instructions to zero. Use of such tables, in fact, in-creases the register pressure and possibility of cache misses, therefore degrading the performance. We will consider only V1, V2 and V1T described in Sect. 2.2 in the rest of this paper.

The Proposed Mix Column Implementation

The MixColumn implementation described by Gladman [17] in V1 requires 4 XORs, 3 rotates and one Xtime opera-tion, incurring 16 XORs, 12 rotates and 4 Xtime operations per AES round.

V1T by Bertoni et al. [12] eliminates the rotations, and requires 16 XORs and 4 Xtime operations per AES round. In fact, the advantage of using a transposed state is more evident in the decryption operation, because the InvMixCol-umn operation sees an important reduction in the number of XORs and Xtime operations.

We describe here a new MixColumn implementation that requires 3 XORs, 3 rotations and one Xtime, incurring 12 XORs, 12 rotations and 4 Xtime operations per AES round. However, using the ARM barrel shifter, the 12 rotations can be combined with 12 XORs without any penalty, resulting in 12 XORs and 4 Xtime operations effectively per round. The proposed MixColumn implementation, in addition to cutting down the number of logical operations, can support all block lengths multiples of 32-bits, unlike the Transposed State Version, which requires a State matrix of 128-bits.

Assuming that b = (b_0, b_1 , b_2, b_3) is the input column to be transformed, s and t are two 32-bit temporary variables, and c = (c_0 , c_1 , c_2 , c_3) is the result, the four steps of the

MixColumn transformation are given as follows, where EOR is the ARM instruction for XOR, and ROL is the rotate left command for the barrel shifter:

EOR s, b, b

1. ROL 8

$s_0 = b_0 \oplus b_1$, $s_1 = b_1 \oplus b_2$

$s_2 = b_2 \oplus b_3$, $s_3 = b_3 \oplus b_0$

EOR t, s, b

2. ROL 16

t = $b_0 \oplus b_1 \oplus$ t

0 b_2 1 = $b_1 \oplus b_2 \oplus b_3$

= $b_2 \oplus b_3 \oplus$

t_2 b_0 $t_3 = b_3 \oplus b_0 \oplus b_1$

3. s = Xtime(s)

= $\{02\}_3$ $(b_0 \oplus b_1)$= $\{02\}_3$ $(b_1$

s_0 s_1 $\oplus b_2)$

s = $\{02\}_3$ $(b_2 \oplus b_3)$= $\{02\}_3$ $(b_3$

2 s_3 $\oplus b_0)$

4. EOR c, s, t ROL 8

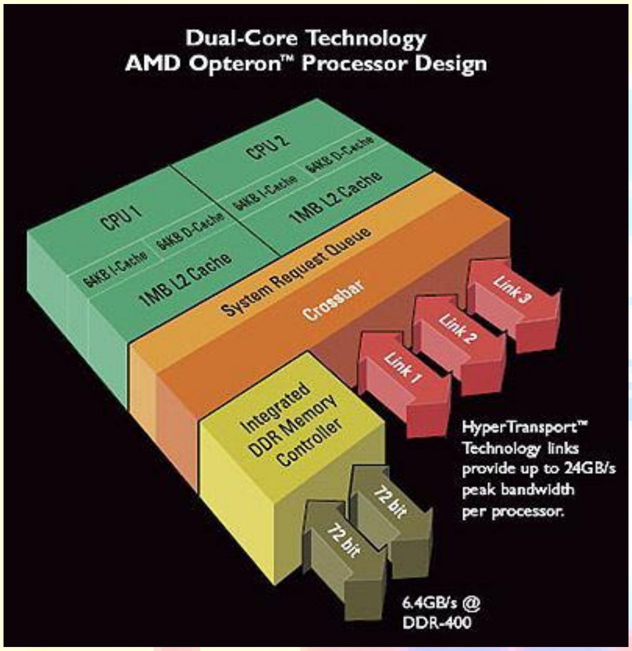
$c_0 = \{02\}_3 \ (b_0 \oplus b_1) \oplus b_1 \oplus b_2 \oplus b_3$ $c_1 = \{02\}_3 \ (b_1 \oplus b_2) \oplus b_2 \oplus b_3 \oplus b_0$

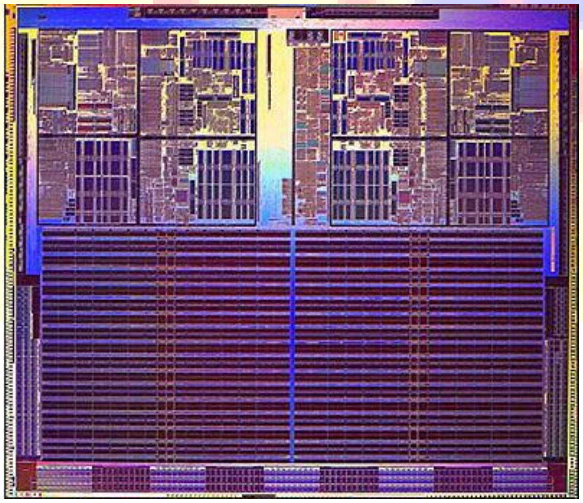
$c_2 = \{02\}_3 \ (b_2 \oplus b_3) \oplus b_3 \oplus b_0 \oplus b_1$ $c_3 = \{02\}_3 \ (b_3 \oplus b_0) \oplus b_0 \oplus b_1 \oplus b_2$

The final result is equivalent to (1).

Dual Single-Core vs. Single Dual- Core

AMD's Opteron chip is capable of SMP due to its multiple hyper transport links, so which is faster; a single dual-core chip or two single-core chips? On paper, dual Opterons should be faster than a single dual-core Opteron at equivalent clock speed for one major reason: Due to the built-in memory controller, each Opteron has exclusive access to its own set of system memory.





The dual-core designs have to share the memory controller, leading to competition for resources that will inevitably drag down comparative performance. Intel SMP systems do not gain this advantage over dual-core siblings since they already share a single memory controller over the front-side bus of the motherboard. It's difficult to tell whether either design has any performance advantage in Intel's implementation. The data has a shorter path to travel with the dual core chips, but not so much as to make a radical difference. Certainly Intel dual-core chips should have a pricing advantage over SMP solutions, especially when you factor in the price premium that dual-socket motherboards demand. It's time to talk money. At first glance, basic economics suggests that dual-core processors should be more affordable than buying a pair of single core processors. After all, the companies are integrating two cores into a single die, saving manufacturing effort. Besides, there would be no point in charging extra money for the second core of a dual-core chip; no one would buy it, right? Maybe, but let's not forget what dual-core chips have to offer besides convenience. The picture is quite different for Intel as opposed to AMD, so let's run through each company's pricing strategies for these chips.

Quad-Core Processors

In November 2006, Intel introduced the first quad core microprocessors for the volume x86 markets. The quad-core chips were designed to offer better performance compared with the previous generation of single- and dual-core processors. A little less than a year later Advanced Micro Devices brought its quad-core Opteron to the market, showing that all four cores could be placed on a single piece of silicon. While chipmakers figure out their next chip movies, here are 10 things you should know about quad core processors. While Intel came to market first with a quad-core processor, it did so essentially by tying two dual core processors together on the same silicon package. AMD took more time to bring its quad-core chip to market but developed a manufacturing process that placed all four cores on the same piece of silicon.

Bibliography

[1] International Technology Roadmap for Semiconductors, “International technology road map forsemiconductors—System drivers,”2007[Online].Available:http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_SystemDrivers.pdf

[2] Intel Corp., “Intel core i7-940 processor,” Intel Product Information, 2009 [Online].Available: http://ark.intel.com/cpu.aspx?groupId=37148

[3] Element CXI Inc., “ECA-64 elemental computing array,” Element CXI Product Brief, 2008 [Online]. Available: http://www.elementcxi.com/downloads/ ECA64ProductBrief.doc

[4] ITU-T, “H.264.1: Conformance specification for h.264 advanced video coding,” Tech. Rep., June 2008.

[5] “Intel 64 and IA-32 Architectures Software Developer’s Manual,” Intel Developer Manuals, vol. 3A, Nov. 2008.

[6] ARM Ltd., “The ARM Cortex-A9 Processors,” ARM Ltd. White Paper, Sept. 2007 [Online].Available:http://www.arm.com/pdfs/ARMCortexA-9Processors.pdf

IJMRA