Arm Refocuses DSP Effort

New ARM9E Offers Improved DSP Performance

by Krishna Yarlagadda

EMBEDDED PROCESSOR FORUM

To address the growing need for better signal-processing performance in high-volume system-on-a-chip applications, such as mass-storage devices, cellular phones, and information appliances, Arm has made another attempt to get it right. In 1996, Arm introduced a completely separate DSP core, Piccolo (see MPR 11/18/96, p. 17). With its own instruction set, Piccolo has not become popular, due to the unconventional and complicated mechanism it uses to communicate and coordinate with an ARM processor.

This time, Arm has enhanced its existing ARM9 core (see MPR 12/8/97, p. 10) with a new set of instructions and supporting hardware to improve signal-processing performance. At last month's Embedded Processor Forum, John Rayfield, DSP development manager at Arm, presented the new ARM9E architecture with these added features. The approaches used in the ARM9E are similar to those adopted by Hitachi, Lexra, and ARC for their DSP-enhanced cores. Today, Arm is not actively licensing Piccolo, as it believes that the ARM9E is a better solution for medium-level signal-processing performance.

Targeting a Bigger Role

With its new core, Arm is targeting high-volume applications such as mass storage, voice over IP, cellular phones, modems, networking appliances, automotive control, and audio decoding (Dolby Digital, MP3, etc.). All of these applications require a mix of good DSP and control-code performance. ARM already dominates some of these applications as a microcontroller, and now, with these added DSP capabilities, Arm hopes to replace or complement existing DSPs. As a microcontroller, the ARM9 has reasonable support for digital signal processing—a Harvard memory architecture, a multicycle 32 × 32-bit multiply-accumulate (MAC) instruction, and conditional execution to minimize branch penalties. The ARM9E adds a 32 × 16 MAC unit (two-cycle latency with single-cycle throughput), saturating fractional arithmetic, and SIMD operations on packed 16-bit words to make more efficient use of the 32-bit data paths and ALUs.

New instructions in version 5 of the ARM architecture with Thumb DSP extensions (V5TE) include 32 × 16 and 16 × 16 MAC and multiply instructions as well as four fractional saturating arithmetic instructions. V5TE also includes CLZ, an instruction that supports faster normalization and division by counting leading zeroes. This instruction was first added to the ARM10 (see MPR 11/16/98, p. 14) architecture that was introduced at last year's Embedded Processor Forum. Table 1 shows all of the new instructions in ARM9E.

Instruction	Operation	Comments
SMLAxy{cond}	16 × 16 + 32 → 32	Signed MAC
SMLAWy{cond}	32 × 16 + 32 → 32	Signed MAC wide
SMLALxy{cond}	16 × 16 + 64 → 64	Signed MAC long
SMULxy{cond}	16 × 16 → 32	Signed multiply
SMULWy{cond}	16 × 32 → 32	Signed multiply wide
QADD R_d, R_m, R_s	SAT(R_m + R_d)	Saturating add
QDADD R_d, R_m, R_s	SAT(R_m + SAT(R_s × 2))	Saturating add double
QSUB R_d, R_m, R_s	SAT(R_m – R_d)	Saturating subtract
QDADD R_d, R_m, R_s	SAT(R_m – SAT(R_s × 2))	Saturating sub double
CLZ{cond} R_d, R_m	COUNTZ(R_m)	Count leading zeros

Table 1. ARM9E adds a new set of new instructions to speed up DSP-related tasks.

Multiple Multiply Options

New multiply-related instructions treat a 32-bit general-purpose register as two 16-bit values or as one 32-bit value. Figure 1 illustrates these operations using the SMLAxy instruction as an example. This instruction performs a 16 × 16 + 32 → 32 operation in two cycles with single-cycle throughput. These multiply and MAC operations can choose the top half, bottom half, or the full register as sources (shown as Rm and Rs in Figure 1) for the multiply operation. When 16-bit values are chosen, they are sign-extended to 32 bits before the multiplication is done. MAC-related instructions can choose a single register (32 bits) or a register pair (64 bits) as the accumulator (Rn). The result of the MAC and multiply operations can be either 32 bits or 64 bits long. A signed multiply-accumulate-long instruction takes one extra cycle to compute the results compared with other MAC-related instructions.

The new fractional saturating arithmetic instruction QADD performs saturating addition of two operands, setting

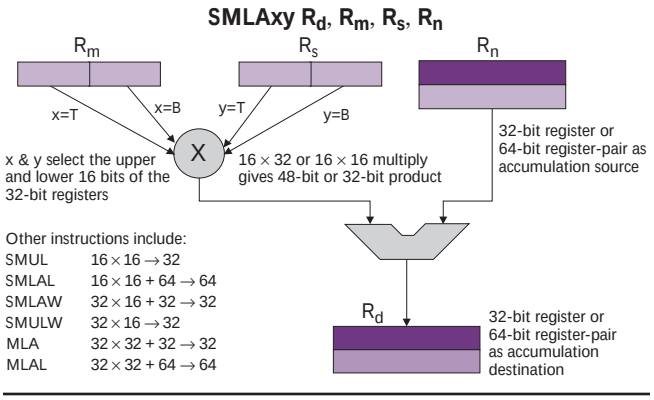

Figure 1. Signed multiply-accumulate instructions can perform 16 × 16 or 32 × 16 multiplies with 32- or 64-bit accumulators.

Price & Availability

Arm will deliver the ARM9E to its lead partners in 3Q99 and expects to produce the first silicon in 4Q99. For more information, go to *www.arm.com*.

the saturate flag if the result overflows or underflows (is too large or too small to be represented in the destination register). QDADD performs a saturating add of one operand to the saturated intermediate result of doubling the second operand, setting the saturate flag if the result overflows or underflows. QSUB and QDSUB perform saturating subtract operations in a similar fashion. These instructions are useful for implementing wireless international-standards algorithms.

Arm Takes Conservative Approach

Arm's approach is conservative compared with more aggressive additions found in Hitachi's SH-DSP (see MPR 12/4/95, p. 10), Lexra's MIPS-like LX5280, and ARC's version 3.0. Arm gave higher priority to keeping things clean than to copying all the architectural features of conventional DSPs. Hence, there is no special support for important DSP features, such as modulo addressing, bit-reversal addressing, and zero-overhead looping. In general, those features add complexity and reduce clock speeds. Not having special hardware support for these features increases the cycle count for DSP inner loops, but it minimizes cycle-time penalties. Arm expects the ARM9E core to run at the same frequency as the ARM9 in an equivalent process technology. Arm has not added any registers or state, and there are no register-usage restrictions.

MICHAEL MUSTACCHI

John Rayfield, Arm's DSP development manager, presents the ARM9E at the Forum.

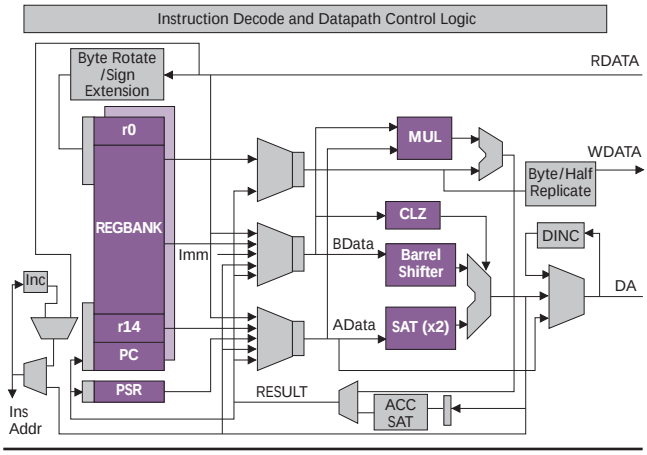

Figure 2. ARM9E datapath with additional resources to support new instructions.

This approach also makes things easier for the compiler. Arm expects developers to write primarily in C, with some assembly-language optimizations. Because the new instructions implement 16-bit saturating fractional arithmetic that is not supported in C, an inline assembler handles those operations in C and C++ programs. This allows the user to create a fractional saturating data type through inline assembly-language functions that use the new instructions. These functions expand as macros during compilation. Arm does not anticipate that anyone will write more than 20% of the code in assembly language, although this figure is very application dependent.

Lexra also added DSP extensions, dubbed Radiax, to its instruction set (see MPR 5/10/99, p. 5). Radiax offers 36 new instructions, eight new 39-bit accumulators, and two multiply units that can execute two 16×16-bit multiply operations with 39-bit accumulation in three cycles (two cycles for multiply and one for add) with single-cycle throughput. The ARM9E can execute only one 16×16 or one 32×16 multiply at a time. These 16×16 or 32×16 MAC instructions with 32-bit accumulation execute in two cycles with single-cycle throughput, and instructions with 64-bit accumulation take an extra cycle.

Lexra didn't change its memory architecture significantly, but it did add quadword load/store operations that transfer 64 bits of data (four 16-bit words each) between a register pair and memory, a feature that helps to sustain two 16×16 MAC operations per cycle. Lexra also added three 32-bit start/end address registers to support circular buffers and a new instruction to perform bit-reversed address calculations. A subsequent instruction can use this register for addressing. The ARM9E doesn't have any changes to the memory architecture to support the new instructions, as the memory bandwidth of ARM9 is sufficient to feed the new datapath.

The LX5280 is a superscalar processor that can issue as many as two instructions per cycle, whereas the ARM9E issues only one. This allows the Lexra processor to, for example, issue a load with a MAC or an ALU operation in one cycle, which speeds up many DSP inner loops.

ARC's core (see MPR 5/31/99. p. 16) offers more flexibility than Lexra's or Arm's. In addition to accelerating various DSP operations using min/max, normalize, swap, barrel shift, and zero-overhead loops, ARC's core allows designers to choose either a $24 \times 24 \rightarrow 56$-bit MAC or one or two $16 \times 16 \rightarrow 40$-bit MACs. ARC has also made changes to the memory subsystem and has added X and Y pages, plus an option to change the size and number of the banks.

Lexra and ARC may pay some cycle-time penalties for their extra features, but the Lexra part has a seven-stage pipeline—two stages longer than ARM9's—which should

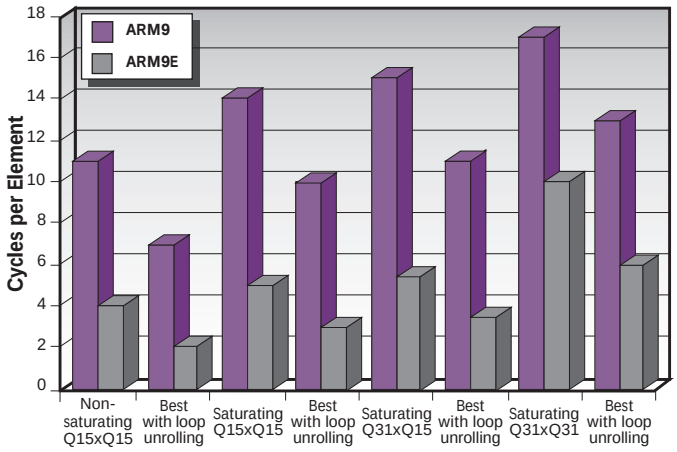

Figure 3. The ARM9E performs dot product significantly faster than the ARM9. (Smaller is better.) (Source: Arm)

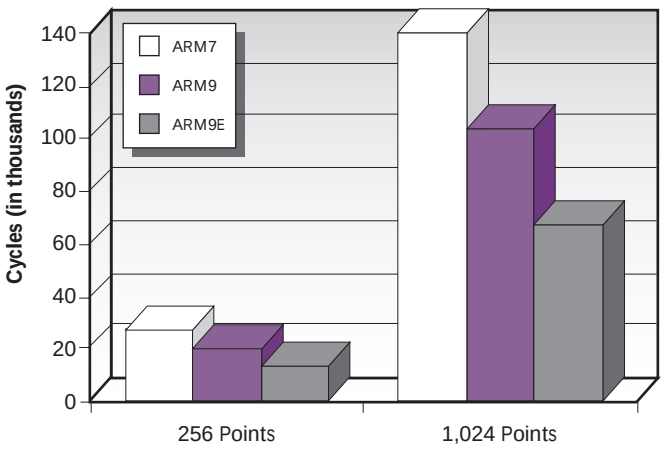

Figure 4. The ARM9E performs FFT (fast Fourier transforms) faster than the ARM9 and ARM7. (Smaller is better.) (Source: Arm)

eliminate any frequency disadvantage compared with ARM9. ARC seems to have lost about 20% in frequency due to its new features, while Arm claims no loss.

Improved DSP/Communications Performance

The data path in the ARM9E is essentially the same as that in the ARM9: a five-stage pipeline and a Harvard memory interface. The new blocks, shown in Figure 2, are the fast multiplier (now 32 × 16, previously 32 × 8), the CLZ block, and the two saturation blocks. One saturation block performs a double-and-saturate, producing a fractional product; the other performs a straight saturation of the accumulated value.

Arm expects the ARM9E's integer performance to be identical to that of the ARM9, i.e., 220 Dhrystone 2.1 MIPS at 200 MHz in a 0.18-micron process. With the new instructions and faster multiply unit, however, inner loops of key DSP algorithms should execute more quickly. As Figure 3 shows, the ARM9E performs a dot product 1.5 to 3 times faster than the ARM9. Dot product is an important function, used heavily in convolution and filtering algorithms. Similarly, as Figure 4 shows, the ARM9E performs FFT (fast Fourier transform) 1.2 to 1.4 times faster than the ARM9. Note that performance improvements of the FFT are not as high as those of the dot product.

According to Arm, a full-duplex G.723.1 speech codec takes 25% of the ARM9E's cycles at 160 MHz, making it about twice as fast as the ARM9. Other speech codecs, such as those used in cellular phones, should see similar speedup. Arm believes the processor can implement a V.34bis modem, an MP3 audio decoder, and Dolby Digital (AC3) audio using about 28%, 11%, and 22% of the ARM9E's cycles, respectively.

Enhanced Debug Capabilities

The ARM9E processor has better features than ARM9 for real-time debugging, such as EmbeddedICE-RT logic, an enhanced version of Arm's EmbeddedICE JTAG interface. Along with new real-time trace and monitor tools, ARM9E provides real-time developers with leading-edge debug capabilities. The real-time trace tool allows nonintrusive instruction and data tracing, and the real-time monitor allows programmers to debug one task while critical interrupt routines continue running in the background.

As with the ARM9, the ARM9E is supported by Arm's development tools, its real-time debug solution, and the Multi-ICE debug interface. The complete ARM9E solution includes the AMBA (Advanced Microcontroller Bus Architecture) on-chip bus, peripheral IP, development tools, application software, electronic design automation tools, Arm training and support, and design services.

Not Much Silicon Needed

According to Arm, the DSP extensions make the ARM9E's die at most 30% larger than that of the ARM9. The ARM9 measures 2.1 mm^2 (just the core area, excluding cache controllers and MMU) in a typical 0.25-micron process. This is small compared with Lexra's LX5280, which measures 3 mm^2 in a 0.18-micron process. Arm expects the ARM9E to run at 160 MHz in a 0.25-micron process and at more than 200 MHz in a typical 0.18-micron process. Although power consumption figures for the ARM9E are not yet available, the ARM9E is not expected to consume significantly more power than the ARM9, which dissipates about 1 mW/MHz at 2.5-V in a 0.25-micron process.

With the ARM9E, Arm took the right approach: adding new instructions to its already popular ARM instruction set. This resulted in a much simpler unified programming model than the complicated microcontroller-plus-DSP programming model. This approach preserved a reasonable core size and power dissipation without impacting speed. The ARM9E, with its added benefits and minimal overhead costs, is well positioned to penetrate new application areas. M

Krishna Yarlagadda is a leading figure in the DSP community, having founded three DSP companies, ZSP, AVAJ, and Hellosoft. He was also a key contributor to the SuperSparc and UltraSparc programs at Sun. Krishna contributes regularly to industry DSP journals and conferences.