

## **ARM DSP-Enhanced Extensions**

**By Hedley Francis**  
**Senior Systems Engineer, ARM Ltd.**

Emerging standards for algorithms in many application areas have put further demands on the ability of processing platforms to deliver efficient control capability. The signal processing content of algorithms, whilst still calling for high-performance peak-processing capacity, has tended to diminish as a proportion of the total algorithm requirement.

Design teams targeting high-volume applications with high-performance algorithms have a seemingly wide range of design alternatives from which to choose. Competitive pressures in many markets mean that it is critical that the selected platform should be of sufficient performance to implement the chosen functionality *efficiently*. Selecting a design platform of much higher performance than required for the application in hand can result in unnecessary cost and power dissipation, leading to an uncompetitive product specification.

ARM's approach has been to design RISC core architectures with instruction sets that provide efficient support for particular applications, with an optimal balance between hardware and software implementation. In addition, applications that appear to require a DSP-oriented processor because of their high signal processing content can be implemented efficiently with an ARM core. One such example is the MP3 audio algorithm. Analysis of the processing stages in the MP3 algorithm shows that in the critical front-end steps, which include reading of the bit stream, Huffman decoding and inverse quantization, the ARM RISC architecture has a performance advantage over many DSP implementations. In addition, the core can also be used to implement all of the complex control tasks.

To accelerate signal-processing algorithms ARM introduced the V5TE architecture, which adds new DSP instructions to the ARM instruction set. ARM's DSP extensions broaden the suitability of the ARM CPU family to applications that require intensive signal processing, whilst at the same time retaining the power and efficiency of a high-performance RISC microcontroller. The ARM DSP extensions have already been implemented in the ARM946E-S™ and ARM966E-S™ cores and are used in the forthcoming ARM926EJ-S™ processor. Intel has also adopted the 'E' extensions in the ARM core-compliant XScale Microarchitecture, which offers implementation clock frequencies of up to 1GHz.

The ARM approach is to balance the increased performance needs of the target application with the necessity to minimize the core area and power dissipation. A single processor solution such as the ARM9E™, supporting both control and signal processing requirements, has many advantages over traditional solutions based on separate DSP and control processors, both in terms of the efficiency of the final solution and the development process.

### **Target Applications**

ARM has produced successful software implementations of CD-quality audio algorithms such as MP3, including the WMA and MPEG AAC standards, targeting various ARM platforms.

In general, the DSP-enhanced cores are best suited for applications that require a blend of high-quality DSP performance and efficient control implementation. This includes high-volume applications such as in mass storage devices, speech coders, speech recognition and synthesis, networking applications, automotive control solutions, smartphones and communicators, and modems.

### DSP Enhancements

The DSP-enhanced extensions are listed in Table 1. They include a single-cycle 16x16 and 32x16 MAC implementation, and saturation extensions to existing arithmetic instructions. These are intended for use in the design of stable control loops and bit-exact algorithms. The CLZ instruction improves normalization in arithmetic and floating-point operations and enhances performance of the division operation. The DSP-enhanced extensions are fully compatible with the ARMv5TE architecture.

Instruction	Operation	Purpose
SMLAxy{cond}	$16 \times 16 + 32 \rightarrow 32$	Signed MAC
SMLAWy{cond}	$32 \times 16 + 32 \rightarrow 32$	Signed MAC wide
SMLALxy{cond}	$16 \times 16 + 64 \rightarrow 64$	Signed MAC long
SMULxy{cond}	$16 \times 16 \rightarrow 32$	Signed multiply
SMULWy{cond}	$16 \times 32 \rightarrow 32$	Signed multiply long
QADD Rd, Rm, Rs	$\text{SAT}(\text{Rm} + \text{Rd})$	Saturating add
QDADD Rd, Rm, Rs	$\text{SAT}(\text{Rm} + \text{SAT}(\text{Rs} \times 2))$	Saturating add double
QSUB Rd, Rm, Rs	$\text{SAT}(\text{Rm} - \text{Rd})$	Saturating subtract
QDSUB Rd, Rm, Rs	$\text{SAT}(\text{Rm} - \text{SAT}(\text{Rs} \times 2))$	Saturating subtract double
CLZ{cond} Rd, Rm	COUNTZ(Rm)	Count leading zeros

Table 1. ARM9E DSP-enhanced extensions

The hardware architecture to support the DSP-enhanced extensions is based on the existing ARM9TDMI™ RISC core, that is, a five-stage pipeline and Harvard memory architecture. The architectural modifications were carefully considered so that support for the enhanced instructions has minimal impact on the hardware overhead. There are no register or state additions, and no restrictions on register usage. The datapath in the ARM9E contains a limited number of new blocks (Figure 1) – a fast 32 x 16 multiplier, CLZ block, and two saturation blocks.

Because support for the DSP-enhanced extensions is possible without major architectural modifications, the ARM9E compares favourably with the ARM9™. The ARM9E will operate at similar frequencies to the ARM9 core, up to 195 MHz on 0.18µm. It has a die area of only 1.0mm<sup>2</sup> on 0.18µm, and dissipates an estimated 0.5mW/MHz.

The DSP-enhanced extensions do not include special hardware support for instructions such as modulo addressing, bit-wise reversal addressing and zero-overhead looping. Whilst the

hardware to support these functions is significant, they can all be implemented in software using alternative instructions and resources with little performance penalty.

Bit-wise reversal addressing is a common requirement in performing the fast Fourier transform (FFT) operation, a function fundamental to many DSP algorithms. The barrel shifter provides an alternative mechanism for implementing bit-wise reversal, with insignificant overhead. For example, a 512-point FFT takes 29k cycles on the ARM9E processor, of which the total cost of simulating bit-reversed addressing is just 300 cycles, around 1% of the total FFT operation.

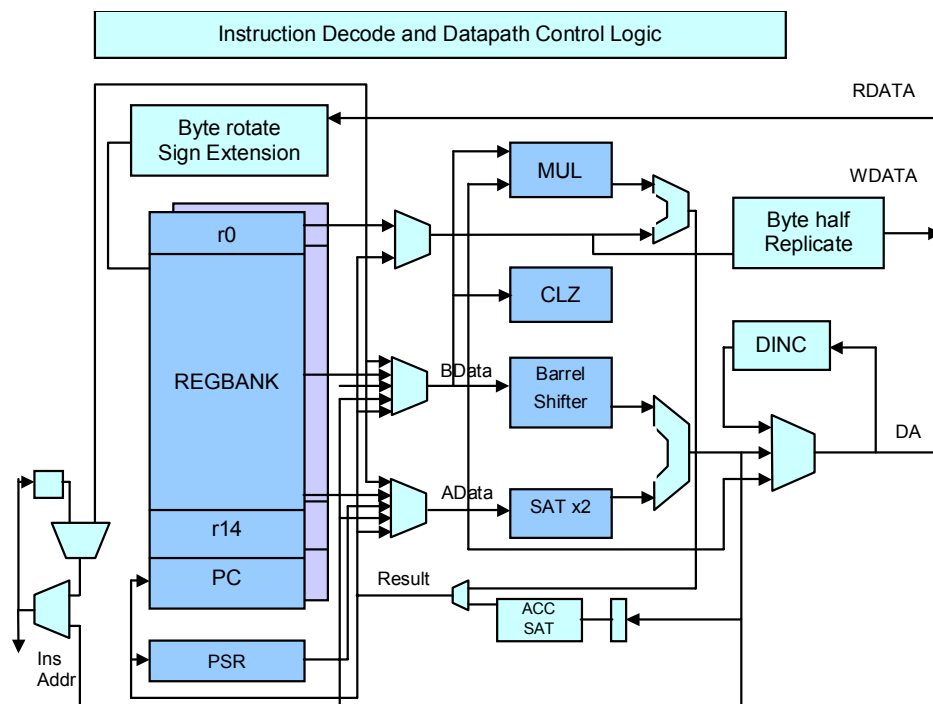


Figure 1. ARM9E datapath highlighting DSP-enhanced hardware modifications.

## Case Study: Speech Codecs

The GSM-AMR (Adaptive Multi Rate) speech codec standard has been selected by the Third Generation Partnership Project (3GPP) for evolved GSM, UMTS and WCDMA networks. Its near-wireline quality speech transmission and efficient spectrum usage means that GSM-AMR may be selected for other next generation wireless and packet-switched networks.

The AMR system adapts speech and channel coding rates according to the quality of the radio channel. The radio resource algorithm, enhanced to support AMR operation, will allocate a half-rate or full-rate channel according to channel quality and the traffic load in the cell. The AMR codec concept is adaptable not only in terms of its ability to respond to radio and traffic conditions but also to be customized to the specific needs of network operators.

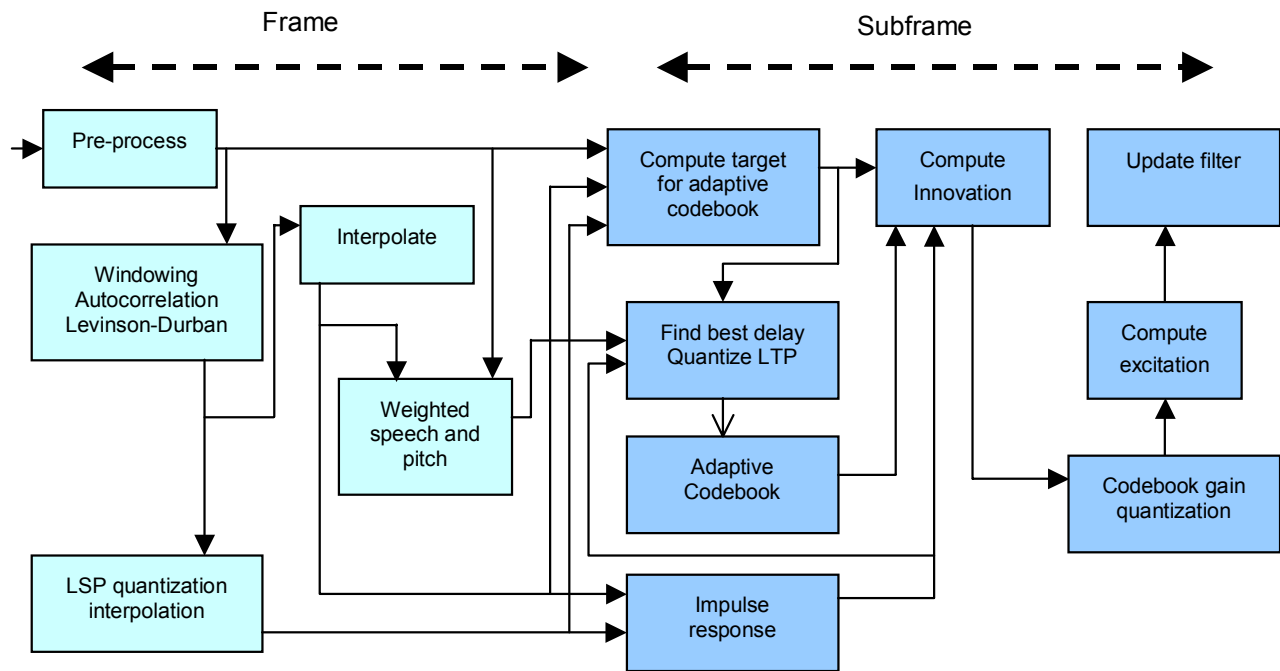


Figure 2. Simplified Diagram of the GSM adaptive multi-rate encoder.

The AMR codec uses eight source codecs with bit-rates ranging from 12.2 to 4.75 kbit/s. The coder operates on speech frames of 20ms corresponding to 160 samples at the sampling frequency of 8000 sample/s. It performs the mapping from input blocks of 160 speech samples in 13-bit uniform PCM format to encoded blocks, and from encoded blocks to output blocks of 160 reconstructed speech samples. The coding scheme for the multi-rate coding modes is the Algebraic Code Excited Linear Prediction Coder (ACELP). At each of the 160 speech samples, the speech signal is analyzed to extract the parameters of the CELP model (LP filter coefficients, adaptive and fixed codebooks' indices and gains). These parameters are encoded and transmitted. At the decoder, these parameters are decoded and speech is synthesized by filtering the reconstructed excitation signal through the LP synthesis filter.

The adaptive multi-rate speech codec is described in a bit-exact arithmetic form using fixed-point ANSI-C code to allow for easy type approval as well as general testing purposes of the adaptive multi-rate speech codec.

#### ARM AMR Implementation

The key to implementing the AMR algorithm, as with many signal processing applications, is efficient digital filtering and correlation. The ARM9E architecture allows a 'block' implementation approach to correlation operations, which is very efficient because register transfers are minimized. The ARM9E instruction set allows multiplication of 16-bit values by packing them into the 'top' and 'bottom' half words of a 32-bit register. The top and bottom 16-bit coefficients can be multiplied by the top and bottom data half words, and accumulated in a 32-bit register to implement a single dot product. Instead of immediately shifting the input data to generate the next product, the implementation uses four general-purpose registers as

accumulators to calculate products for three other combinations of input data and coefficient, without moving either the data or coefficient.

The effect of reusing the data in the blocked implementation is that four correlations are implemented in one pass, saving a substantial number of load/store operations over the entire correlation. The block correlation operation is illustrated in figure 3.

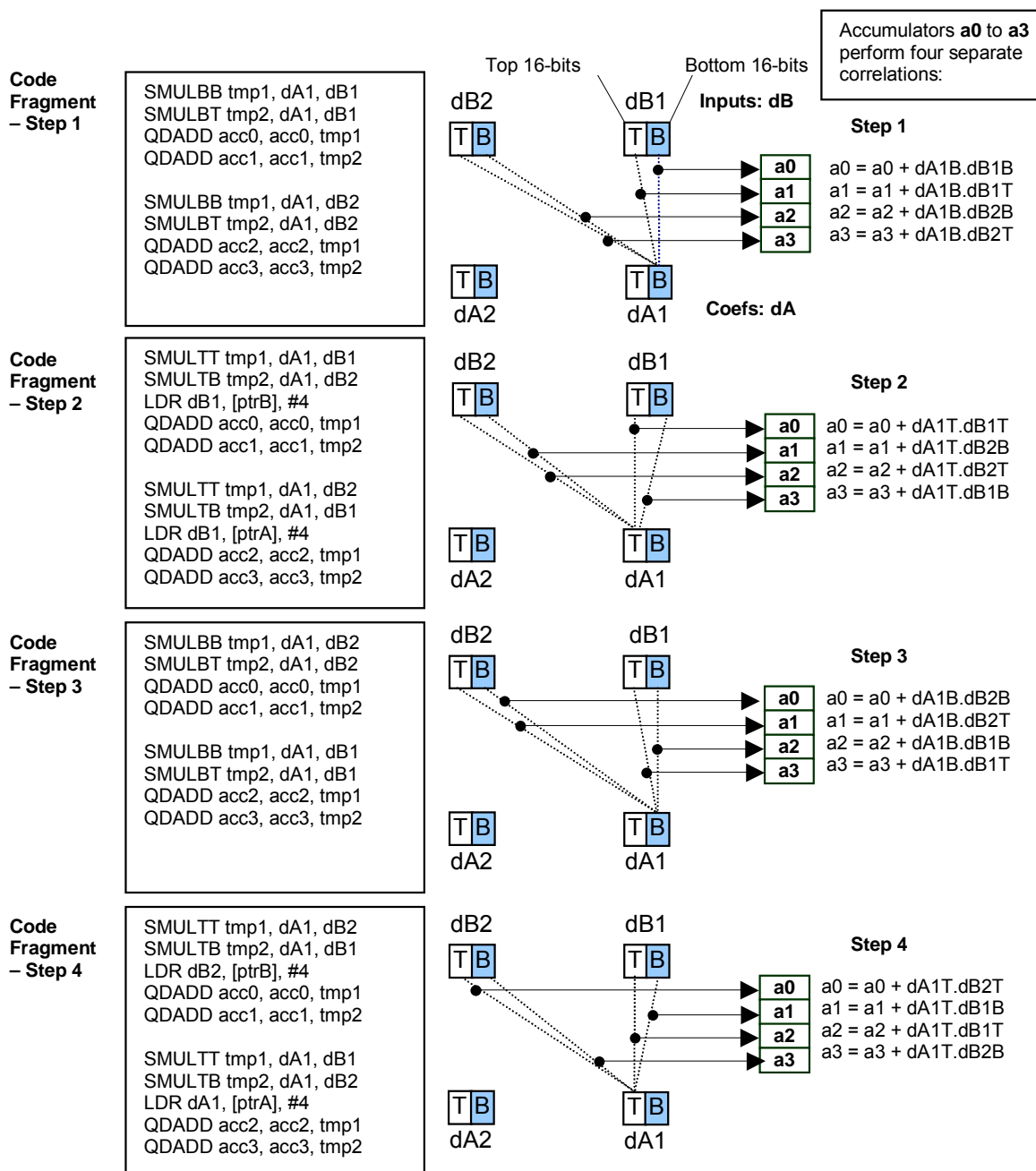


Figure 3. Inner loop correlation Multiply-Accumulates on ARM9E.

Block correlation also benefits from a single-cycle saturating addition available on the ARM9E. The equivalent operation takes four cycles on the ARM9.

The result of this correlation efficiency on the computationally-intensive AMR encode function is that the ARM9E takes just 66% of the cycles required by the ARM9 architecture.

Processor	Encode worst case	Inner loop MAC	Decode worst case
ARM9	55MHz	11 cycles	9MHz
ARM9E	33MHz	5 cycles	7MHz

Table 2. ARM AMR Performance.

### G.723.1 Voice Codecs

The ITU standard G.723.1 is a speech compression algorithm that is an international standard for videoconferencing applications. It encodes 8KHz sampled speech signals for transmission over either 6.4 or 5.3 Kbps channels. G.723 provides approximately 4KHz of toll-quality speech bandwidth. It is the mandatory speech coder for Visual Telephony over GSTN and mobile radio. G.723.1 is also specified as an optional speech coder for Visual Telephony over ISDN, B-ISDN, guaranteed QoS LAN, non-guaranteed QoS LAN, Voice over Frame Relay and Voice over IP (VoIP).

Voice codecs can benefit substantially from the DSP enhanced instructions. The ARM9E is able to implement the G.723.1 algorithm using around 25% of the processor's total available performance – equivalent to approximately 45MHz, with the ARM9 requiring round 76MHz to implement the algorithm. The use of 16-bit precision throughout and the extensive application of saturating arithmetic make the ARM9E ideally suited to these codec applications.

### Comparative Analysis

For many signal-processing algorithms, the peak-processing requirement is often determined by the ability to perform the dot product function efficiently. Figure 4 summarises the consistently better performance of the ARM9E compared with ARM9 for this function. The improvement in performance with the DSP-enhancements is on average 2x.

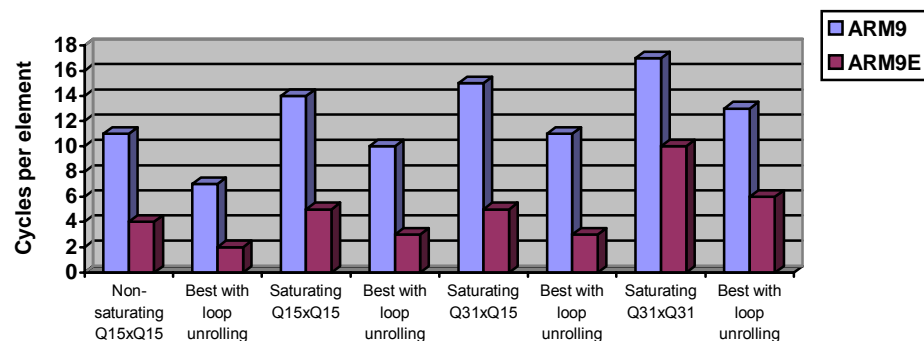


Figure 4. Dot product performance for ARM9E and ARM9

Although the ARM DSP-enhanced architecture does not match the raw processing performance of specialist DSP cores, the ARM architectures can of course provide an integrated platform for implementation of the complex control that is a fundamental requirement of all systems.

One of the most important benefits of the ARM DSP-enhanced solution, and a significant advantage over pure DSP core implementations, is that all the required processing can be performed on the ARM as a standalone processor. This helps reduce power consumption, minimizes chip area and considerably simplifies the hardware and software development process.

The ARM can perform the key algorithmic processing whilst also fulfilling the requirements of the system control functions, such as management of IO, card memory, display and keyboard. In contrast, a DSP-based implementation would require a separate microcontroller to run the rest of the system. An implementation based around two processors will require additional chip area. In addition, development of protocols for control and data exchange between the DSP and the microcontroller will increase the overall system complexity. Integrating all of the functionality onto a single processor is therefore a critical factor in easing the development process and accelerating time-to-market.

Because the ARM DSP-enhanced processors are centred on a single memory system, the availability of a unified memory map considerably simplifies the overall software design task. For systems running an RTOS, it is usually a straightforward task to call the DSP function through an API. In contrast, most RTOS do not provide API support for DSPs, and so the DSP-based solution would require development of bespoke scheduling routines – something that is complex and prone to timing difficulties when the task has to be scheduled out to a second processor.

## Summary

The ARM DSP architecture enhancements and instruction set extensions give designers access to embedded cores capable of implementing high-performance signal processing algorithms without compromising on control performance. The ARM ‘E’ cores increase the flexibility and extend the application space of programmable solutions. These architectures, when implemented as the core of a system-on-chip design, can provide an optimal solution to many of the emerging algorithms calling for high-performance peak processing combined with complex control capability.

ARM’s approach to enhancing the DSP capability of its popular RISC cores is to balance the increase in hardware resources so that sufficient performance is achieved for many critical signal-processing operations without unacceptable increases in area and power consumption.

As well as a range of market-leading embedded processor IP, ARM provides solutions for core-based SoC development, including an advanced embedded software development suite (ADS) and development boards, peripheral PrimeCell IP blocks, and the AMBA SoC standard on-chip bus.