

Hardware Implementation of a High-Precision Fixed-Point MAC for Biomedical Application

Himanshu Verma¹, Anand N², Yogesh Kakasaheb Shejwal³, Jayaraja U. Kidav⁴

¹National Institute of Electronics & Information Technology, Aurangabad, Aurangabad, Maharashtra,
himanshuverma1227@gmail.com

²National Institute of Electronics & Information Technology, Aurangabad, Aurangabad, Maharashtra,
anand.n@nielit.gov.in

³National Institute of Electronics & Information Technology, Aurangabad, Aurangabad, Maharashtra,
yogeshshejwal83@gmail.com

⁴National Institute of Electronics & Information Technology, Aurangabad,
Aurangabad, Maharashtra jayaraj@nielit.gov.in

Abstract. Biomedical signal processing for implantable and wearable technology imposes a crucial design conundrum: operating within strict power and area constraints while obtaining the numerical precision required for clinical validity. With an emphasis on applications like ECG, EMG, and EEG processing, this paper offers a thorough analysis of high-precision Fixed-Point Multiply-Accumulate (MAC) architectures as the key solution to this problem. First, we determine the precision-quantization trade-off by examining the signal-to-noise ratio (SNR) and dynamic range requirements of important biomedical algorithms, such as wavelet transforms, adaptive filters, and light-weight neural networks. Although floating-point units offer inherent dynamic range, optimised fixed-point alternatives are required due to their excessive resource costs. Three axes are used in our review to systematically classify and evaluate MAC architectures: accumulation strategies (guard bits, extended-precision adders), computational core (modified Booth encoding, Wallace/Dadda trees), and system-level optimisation (bit-serial processing, precision-scalable units). We assess their implementation effectiveness and suitability for both ASIC (for ultimate power/area optimisation) and FPGA (for rapid prototyping) targets. With the help of case studies that quantify the effects of MAC architecture decisions on diagnostic metrics like detection sensitivity and feature extraction accuracy in real-time biomedical systems, the paper ends with a synthesis of co-design approaches that connect algorithm development and hardware implementation. The goal of this review is to give researchers and engineers a unified framework for designing effective, high-fidelity signal processing cores for the upcoming generation of autonomous medical devices.

Keywords: Fixed-Point Arithmetic, Multiply-Accumulate (MAC) Unit, Biomedical Signal Processing, Low-Power VLSI Design, Hardware Acceleration, FPGA/ASIC Implementation, Precision-Quantization Trade-off, Wearable Medical Devices, Hardware/Algorithm Co-Design, Digital Filter Implementation.

(Article history: Selected papers from NIELIT's International Conference on Communication, Electronics and Digital Technologies (NICE-DT'26), 29-30Jan, 2026, Gorakhpur, India)

1 Introduction

The fast growth of wearable and implantable medical devices has changed healthcare significantly. These devices allow for continuous, real-time monitoring of physiological signals like the electrocardiogram (ECG), electroencephalogram (EEG), electromyogram (EMG), and photoplethysmogram (PPG). This ability is essential for early diagnosis, managing chronic diseases, and implementing closed-loop therapy. However, handling these subtle signals, which are often affected by noise, at the edge poses a significant engineering challenge due to strict limits on power use, size, and latency. Therefore, computational efficiency is just as important as diagnostic accuracy in real-time biomedical signal processing, which must run complex algorithms with high reliability while using limited battery resources. [1]

The Multiply Accumulate (MAC) operation is at the heart of almost all digital signal processing (DSP) chains. This basic calculation is the basis for FIR and IIR filters that reduce noise, wavelet transforms that analyze time and frequency, and the dense layers of lightweight ML models that automatically extract and classify features. Consequently, the performance, precision, and power profile of the MAC unit directly dictate the capabilities and efficiency of the entire biomedical system.[2]

It is impossible to compromise on the need for high precision when processing biomedical signals. For example, microvolt resolution is necessary to identify minute abnormalities such as ischemic changes in ECG signals. To differentiate low-amplitude neural oscillations from artifacts, EEG analysis requires a high dynamic range. Quantization noise can taint derived parameters like heart rate variability because PPG waveforms are susceptible to it. Quantization errors brought about by inadequate numerical precision can lower the signal-to-noise ratio (SNR), lead to instability in recursive filters, and ultimately result in false positives or negatives, all of which compromise clinical validity.[3]

Since floating-point arithmetic provides a simple solution to dynamic range and precision issues, its hardware complexity leads to excessive power dissipation and silicon area, making it unsuitable for always-on, small devices. The main driving force behind optimized fixed-point implementations is this harsh reality. Fixed-point MAC units can provide the required numerical

fidelity with significant gains in area efficiency and power consumption by carefully adjusting the bit-widths and architectures to the particular statistical characteristics of the target biosignals. This design strategy lowers the energy cost of wireless data transmission and opens the door for long-term, fully autonomous medical monitoring systems by enabling the placement of intelligent processing at the sensor node.[4]

Thus, this review paper focuses on the crucial nexus between hardware efficiency and biomedical algorithmic requirements. We present a thorough examination of high-precision fixed-point MAC architectures, methodically investigating the innovations, trade-offs, and co-design approaches that enable high-fidelity, real-time biomedical signal processing at the edge.

2 Background and Related Work

2.1 Biomedical Signal Characteristics

A basic grasp of the target signals is the first step in designing a high-precision fixed-point Multiply–Accumulate (MAC) unit. The low amplitude, susceptibility to various noise sources, and particular frequency content of biomedical signals define the architectural requirements for any processing hardware. The wide dynamic range of biomedical signals requires careful hardware consideration. For example, diagnostic subtleties such as ST-segment deviations require resolution in the tens of microvolts, whereas the dominant QRS complex in an ECG may have an amplitude of 1-3 mV. Similarly, the range of surface EMG signals is 50 μ V to 5 mV. In order to accurately capture this range without saturation or loss of important diagnostic information, an input quantization resolution usually between 12 and 24 bits is required. Additionally, these signals occupy different frequency bands (e.g., ECG: 0.05–150 Hz; EEG: 0.5–100 Hz), which determine the sampling rates (typically 250 Hz to 1 kHz or higher for neural signals) and, as a result, the processing unit's throughput requirements. In order to handle the full-scale input range and execute computations at the real-time sampling rate, the MAC must support numerical precision.[5]

Biomedical signals are often affected by various sources of noise that the processing chain needs to manage. These include baseline wander, which is low-frequency noise, muscle artifacts from EMG, and broadband noise. There are also motion artifacts, powerline interference at 50/60 Hz and its harmonics, and noise from electrode contact. Additionally, thermal noise comes from the analog front-end. This noisy environment directly influences MAC design. The selected numerical precision should maintain signal integrity above the noise floor. If the bit-width is insufficient, quantization noise increases, potentially masking low-amplitude signal features such as P-waves in ECG signals. Additionally, quantization noise may alias into the frequency band of interest during filtering, thereby diminishing the effectiveness of digital filters implemented with the multiply-accumulate (MAC) operation.[6]

Bit width choice is another important co-design parameter between the algorithm and the hardware. While a resolution of 12 to 16 bits may be adequate for time-domain amplitude estimation in applications such as pulse-based heart-rate determination, other, more challenging, applications like frequency analysis of EEG signals, where the ability to resolve nearby frequency components is critical, or high-order recursive (IIR) filters for active noise cancellation, may demand a higher resolution of 18 to 24 bits. Such high-resolution precision, in addition to being necessary to ensure the stability of the IIR filters, must prevent limit cycles and ensure that the accumulation of the products in the long filters or the transforms themselves does not cause overflows or significant accumulative errors due to truncation of the products. The MAC, therefore, must be designed to provide such precision, possibly higher than the data bits, to ensure accurate processing of the data. [7]

2.2 MAC Operation Fundamentals

Linear digital signal processing is based on the atomic, iterative computing known as the Multiply–Accumulate (MAC) operation. At its core, the MAC performs the fundamental arithmetic:[8]

$$y[n] = \sum_{i=0}^{N-1} a_i \cdot x[n-i] \quad (1)$$

or, in its recursive form critical for feedback systems:

This operation performs an inner product calculation between the vector containing the coefficients (weights) and the vector containing the samples of the data inputs. The non-recursive formula for the convolution sum of an FIR filter is the direct sequential application of the MAC operation. The implementation of this operation on the hardware will determine the performance and precision of the processing system. A simple implementation is to multiply each output value $y[n]$ with the dot product of the set of fixed coefficients $\{h[0], h[1], \dots, h[N-1]\}$ and the input samples within a sliding window. This is the simplest implementation where a separate MAC is responsible for the coefficients and the input samples. Because of its stability and linear phase, the FIR filter is widely used in the biomedical field for baseline wander correction and power line noise cancellation. In the IIR filter, where feedback is added to the signal flow, two MAC computations are needed for each

output value, one in the feedforward path and the other in the feedback path using the previously calculated outputs. The accuracy of the accumulator is crucial because of this recursive nature. Limit cycles or instability may result from round-off or truncation mistakes in the feedback loop, particularly for narrow-band filters employed in ECG rhythm isolation or ECG notch filtering.[9]

An accurate accumulator, therefore, cannot be compromised on. Just like in FIR filtering, the operations of cross-correlation and auto-correlation in the case of estimating the fundamental frequency, for example, are nothing but a series of MAC processing. It involves shifting the signal past another and summing the productions, thus a job apt for a systolic MAC array processor or a pipelined MAC processor. Adaptive noise cancellers for removing EMG components in ECG use algorithms such as Least Mean Square (LMS). The behavior of such adaptive networks is exceptionally dependent on the numerical precision level of the MAC unit, as the value of the step size parameter μ might be small. Other operations aside from filters, statistical operations make up a majority of MAC unit operations for inference in embedded NNs. In this regard, operations such as point-product operations within a perceptron layer embody a complete equivalent of operations performed in a FIR filter. Thus, a well-optimized MAC unit is an essential engine that drives AI applications. [10]

2.3 Review of Existing MAC Architectures

Implementing a high-precision, area-efficient, and low-power MAC unit is a significant challenge in VLSI design, especially for the new wave of biomedical devices. This challenge is a multi-objective optimization problem where three key parameters computational precision, energy efficiency, and silicon area must be optimized at the same time, all while adhering to strict system-level limits. These limits include dead-lines for real-time processing, limited battery life (often aiming for multi-year use in implants), and strict size requirements for wearability or implantability. The design space for fixed-point MAC units is very diverse, ranging from traditional direct-mapped structures to highly specialized, algorithm-aware accelerators. Each design approach offers different trade-offs, balancing generality, performance, and efficiency in unique ways. [11]

2.4 Booth/Radix Multiplier-Based MACs

In an attempt to overcome the shortcomings in array multipliers, the Modified Booth encoding scheme has emerged as the de-facto standard technique in precision-optimized fixed-point MAC designs. The basic insight in the Modified Booth Algorithm involves recoding the multiplier to minimize the number of non-zero bits and thus the total number of sums to be computed in the most computer-intensive phase of the multiplication operation. The Standard Radix-4 Modified Booth Algorithm looks at the bits in the multiplier in a window of three bits at a time to generate adigit values from the pattern $\{-2, -1, 0, +1, +2\}$. This helps in significantly cutting down the number of partial products to almost half that need to be added compared to the naive method, thereby directly leading to a reduced number of adders in the final reduction tree (usually a Wallace/Dadda tree that is designed for an optimal critical path). In a standard N-bit multiplication, the count of partial products is reduced from N to $\lceil N/2 \rceil$. The hardware implementation needs encoders for every digit and a module for generating the partial product, involving conditional shifts and two's complement negation. In the context of biomedical signal processing, where the operands are naturally signed (such as in the case of biosignals that have both positive and negative peaks), this signing capability directly saves the area in sign extension logic.[12]

The Booth multiplier embedded in a MAC module has many advantages, which are very important for biomedical systems. The immediate consequence of having a smaller number of partial products is that it decreases switching activity at a large combinatorial logic tree level. That translates to dynamic power consumption, which is a primary concern for battery-powered equipment. This technique produces an array of partial products that can lead to layout and timing closure ease. The design can scale up properly to larger bit-widths (24-bit or 32-bit) as required by biomedical computing without having an exponential increase in the critical path. The selection of the radix has important trade-offs.

It is possible to migrate to Radix-8 Booth coding because it requires only about $\lceil N/3 \rceil$ partial products, and this has many benefits in terms of area and possible power savings. The cost of migrating to Radix-8 Booth coding is quite steep because it requires an extra addition for the $\times 3$ multiple, which is either an extra preprocessing time or an extra hardware component. The use of more complex encoders and the requirement to process more values of the digits $\{-4, -3, -2, -1, 0, +1, +2, +3, +4\}$ add to control logic complexity. The Radix-4 transform may be the most efficient between complexity and performance when processing biomedical signals with moderate sample rates, generally below 1 kHz. When incorporating the Booth multiplication algorithm into a fully functional MAC unit, new architecturally based challenges also arise. The accumulator is then required to accommodate the extended-range dynamics that can be present in the multiplication results, particularly when such multiplication is performed at extreme levels (related to the -2 and $+2$ operations).[13]

2.5 Pipelined MACs and Architectural Implementation and Staging

Pipelining is a typical example of a classic throughput-extensive optimization method for MAC modules and directly deals with the critical path penalty in traditional combinatorial implementations. Addition of pipeline stages between calculation stages splits up the long propagation time of the multiplier and accumulator into several segments, which in turn facilitates a considerable increment in the clock rate and, ultimately, the system throughput. Pipelined MAC modules are very useful for implementing long-tap filters in biomedical signal processing systems, in which the sampling rates could vary from several hundreds of HZ in the case of EEG analysis to tens of kHz in neural spike sorting.[14]

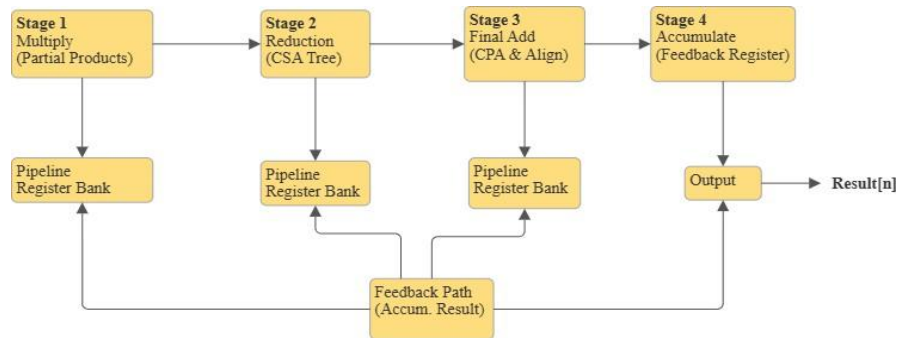


Fig.1 Four Stage Pipeline MAC Architecture

Even though pipelining has a tremendous impact on improving the throughput rate, several fundamental trade-offs are also made that need careful attention in biomedicine applications. The pipeline latency, which is the delay between the input and the available output, linearly increases with an increase in the pipeline stages in figure 1. This additional slack needs to be within the stipulated response time budgets for closed-loop biomedicine applications. However, the rate of output, or throughput, approaches one result per cycle after the pipelined unit is full, thereby facilitating biomedicine applications involving high-data-rate signals in a purely real-time manner. Every additional pipeline register increases the chip area and contributes marginally to both static and dynamic power dissipations. Even the static power dissipation has a two-fold increase; the power dissipation due to the registers themselves and the increased switching activity in higher clock rates made possible by pipelining. This marks a point where diminishing returns exist, i.e., past which the power cost exceeds the throughput advantage.

Pipelining also increases the complexity of control logic, particularly in adaptive bio-medicine algorithms involving coefficient updates and/or conditional execution. In the context of biomedical applications, the following rules are generally followed. Since the data rates of many biosignals (ECG, EEG, EMG) are only of a moderate amplitude, highly pipelined designs are typically unnecessary. A pipelined chip with two stages is typically adequate for supporting the timing requirement at low supply voltage. The key advantage of pipelining is the capability to reduce the critical path through the voltage scaling technique. The MAC circuitry can be scaled down using a lower supply voltage (VDD), delivering the same operational frequency while providing quadratic reductions in power compared to conventional systems ($P_{dynamic} \propto f \cdot C \cdot V^2$). It is a very important method for realizing ultra-low-power consumption. The pipeline can be flexibly configured according to the algorithm. Take the LMS adaptive filter for example. There may specifically be a pipeline for calculating the coefficients update. If it is a multi-mode processor, the pipeline can be designed to be configurable, where some stages are bypassed when the computation involves lower precision to reduce power consumption. [15]

2.6 Distributed Arithmetic (DA)

Distributed Arithmetic (DA) signifies a paradigm shift in the implementation of the MAC operation and relies much less on traditional multiplier-based implementations than other arithmetic units. This is very important in the field of biomedicine since the coefficient sets in such applications are fixed in many biomedicine applications. The standard DA implementation handles one bit of all inputs per clock cycle and needs B clock cycles per output bit. The accumulator performs a shift-add operation: $ACC \leftarrow (ACC \gg 1) + LUT_output$. For signed numbers, the most significant bit (MSB) cycle requires subtraction instead of addition. To address the exponential growth of LUT size (2^K) with tap length K, several decomposition techniques are employed. The K coefficients are partitioned into P smaller groups of L coefficients each ($K = P \times L$). Each group has its own smaller LUT (2^L entries). The outputs from all P LUTs are summed in an adder tree. This trades off LUT size for additional adders. By recoding the input bits from {0,1} to {+1,-1}, the memory size can be halved to 2^{K-1} entries at the cost of slightly more complex address generation and accumulation control. Exploits coefficient symmetry in linear-phase FIR filters to reduce the effective K by nearly half. Uses DA for a subset of coefficients and conventional multipliers for others, optimizing the area-speed trade-off. DA is ideal for fixed-coefficient systems such as FIR filters for baseline wander removal (ECG), Fixed wavelet

filter banks for EEG sub-band decomposition, Predefined correlators for QRS complex detection, Static feature extraction kernels.[16]

3 Fixed-Point Precision Requirements In Biomedical Processing

3.1 Fixed-Point vs Floating-Point Trade-offs

The choice between fixed-point and floating-point arithmetic represents a fundamental architectural decision that dictates the power profile, area efficiency, and programming model of biomedical signal processing systems. This trade-off is a real game-changer when it comes to wearable and implantable devices that have limited resources - basically anything that can't just suck down a lot of battery power or be some kind of clunky device. The efficiency of these devices is literally a matter of life and death and form factor is key too. Floating point arithmetic is a bit of a lifesaver here - using IEEE 754 single precision for example gives you a 24 bit mantissa and an 8 bit exponent which means you end up with roughly 7 decimal digits of precision over an absolutely massive dynamic range. This is a huge plus because it basically eliminates the risk of overflow when you're dealing with recursive computations and makes it a whole lot easier to develop the algorithms. On the flip side fixed point arithmetic has a static scale factor that's just predetermined - basically the binary point is set in stone. And while fixed point can in theory keep up with the precision of floating point (if you for example use a 24 bit word length), it's got an incredibly limited dynamic range because it's all about integers - so you end up needing to do a lot of careful scaling just so you don't overflow while still managing to get enough precision for the really low signal components. [17]

A 32-bit floating-point multiplier involves about 3-5 times as many logic gates as a 32-bit fixed-point multiplier. This leads directly to an increase of 40-60% in power and an area of 2-3 times that of the fixed-point multiplier. Additionally, and specifically for a floating-point multiplier, there is the added complexity of the multiplier's accumulator stage, which involves aligning the mantissa (barrel shifters) and renormalizing the result of each addition. Such differences make this complexity overhead frequently un-feasible for biomedical applications that require a microwatt power budget. Recent results have demonstrated that optimized fixed-point MAC modules have achieved energy efficiencies of 5-10 pJ/operation at a resolution of 24-bit fixed-point processing, whereas that of single-precision floating-point processing consumes 50-100 pJ/operation at a pitch of 28nm CMOS technology.[18]

Floating-point operations generally involve greater pipeline delays (Multiply-Add: 5-8 cycles) against fixed-point operations (2-4 cycles) because of the multi-step normalization procedure involved. Though pipelining can make the throughput equivalent, it affects real-time interaction during closed-loop control scenarios, such as responsive neurostimulation. Fixed-point processing allows easier interaction with real-time operating systems and real-time power management because it is deterministic and requires simpler control logic. The trade-off here lies emphatically with fixed-point computation in the majority of biomedical edge devices, as there is a greater potential for static scaling of algorithmic coefficients during devices with relatively predictable signal ranges. Floating-point computation makes sense when developing Biomedical Workstations or Research Platforms. [19]

3.2 Word-Length Analysis for Biomedical Signals

The optimal word length is a crucial co-design parameter. Various biosignals require different word lengths depending upon their physiological properties. ECG signals require a medium word length with a medium signal range of 0.1 to 5 mV. The most essential information of ECG is provided by the subtle signal properties. The complex requires 8-10 bits for reliable detection, but accurate measurement of ST-segment deviations for ischemia detection demands higher precision. Analysis of high-resolution ECG for late potentials requires up to 1 μ V resolution. Consequently, 12-16 bit ADCs are standard, with internal processing typically requiring 16-24 bit fixed-point arithmetic in the MAC to preserve subtle features through filtering and feature extraction operations.[20].

Surface EMG amplitudes range from 50 μ V to 5 mV, but the information content is often in the frequency domain or requires high dynamic range for force estimation. Intramuscular EMG and neural spike recordings demand higher resolution to distinguish individual motor unit action potentials. For 14-18 bit ADCs are common, with internal processing requiring 18-32 bit precision in the accumulator to maintain signal integrity during wavelet transforms or spectral analysis. These signals present the most demanding requirements. PPG signals for pulse oximetry and blood pressure estimation contain subtle amplitude modulations (<1% of DC component) that require high signal-to-noise ratios. EEG signals range from 1 μ V to 100 μ V, with clinical interest often in microvolt-scale event-related potentials. Both require high-resolution MAC units with 24-32 bit internal precision to extract meaningful information from noise. In particular is given in the Table 1, the AC component of PPG signals requires careful preservation through multiple stages of processing, and therefore, digital filtering operations require extended bit-widths. [21]

Table 1. Word-Length Analysis for Biomedical Signals

Signal Type	ADC Resolution	Recommended MAC Input Width	Required Accumulator Width	Key Application
ECG(Standard)[19]	12–14 bits	16 bits	24–32 bits	Arrhythmia detection
ECG(HRV/High-Res)[19]	16 bits	18–24 bits	32–40 bits	Heart rate variability
Surface EMG[20]	14–16 bits	18 bits	28–36 bits	Muscle activity monitoring
Neural Spikes[20]	16–18 bits	20–24 bits	32–40 bits	Brain-machine interfaces
PPG[21]	16–18 bits	20–24 bits	32–40 bits	Pulse oximetry, BP estimation
EEG[19]	16–24 bits	24 bits	40–48 bits	Seizure detection, BCI

3.3 Effects of Rounding, Saturation, and Overflow

The finite-precision nature of fixed-point arithmetic leads to a number of non-ideal effects that can seriously degrade system performance unless managed properly. Such effects are especially critical in biomedical applications because the integrity of the signal has a direct bearing on the validity of diagnosis. In the MAC data path, every truncation or rounding operation introduces quantization error that is typically modelled as additive white noise with uniform distribution. For rounding operations, the error variance is $\sigma^2 = 2^{-2b}/12$, Here, b is the number of fractional bits. When it comes to recursive filters, such an effect may result in an accumulation of these noises and even lead to limit cycles that take the form of sustained oscillations and continue even when the input is zero. Another effect of quantization is the signal to quantization noise ratio (SQNR). This has significant effects on the detectability of low signals in biomedical signals. For example, if the SQNR is not adequate, it may hide the ERP signals necessary for BCI.[22]

Overflow happens when the value obtained after an operation exceeds the representable value range in the fixed-point form. A straightforward two's complement method produces errors of magnanimous values, leading to the copulation of the whole signal. Biomedically, saturation arithmetic is used, and the computations saturate to the greatest representable value. This eradicates the possibility of catastrophic errors, although it introduces nonlinear distortion, which may affect the succeeding stages involving linear processing. More advanced techniques involve block floating-point processing, wherein all calculations in a certain block involve the same exponent, and the exponent varies and is updated to deliver enhanced dynamic range, thereby requiring no full floating-point processing at all. In biomedical signal processing, the computations differ differently from general computing, requiring deterministically precise numerical processing to ensure the efficacy and validity that the FDA deems necessary for the approval and usage of medical equipment and devices. Floating-point calculations, although convenient and useful, lead to discrepancies in processing between different platforms owing to differences in rounding and evaluation order, making fixed-point processing, if properly implemented, bit-exactly reproducible between different platforms, which is essential and necessary for the usage and approval of FDA-approved medical instruments and equipment. [23]

4 High-Precision Fixed-Point MAC Architecture

4.1 Architectural Overview

The architectural design of a high-precision fixed-point MAC unit is a natural integration of specialty computational blocks, each optimized for the unique biomedical signal processing requirements. The architecture needs to balance between numerical accuracy and hardware efficiency, allowing for deterministic behavior while minimizing power consumption. A canonical high-precision in the figure 2 MAC data path presents a staged computational flow: Operand Conditioning, where the input operands are subjected to sign extension, scaling, and alignment to a common fixed-point format; a Multiplication Core that computes the full precision product; an Accumulation Stage that sums the product to a running sum; and Output Conditioning, which includes rounding, saturation, and format conversion. For biomedical applications, this data path is usually optimized for 24-32-bit internal precision, carefully considering the preservation of low-amplitude signal components through several stages of processing. The architecture often provides configurable modes of precision to adapt to different signal types: 16-bit for EMG envelope detection, 24-bit for EEG spectral analysis, etc. By their very nature, biomedical signals contain both positive and negative deflections. Hence, good

sign handling is crucial. Most of the implementations use two's complement representation along the data path because this avoids separate circuits for sign-magnitude conversions at several places along with considerable simplification of arithmetic operations. Within the multiplier, the signed algorithms such as Booth encoding or Baugh-Wooley multiplication take two's complement inputs directly without any preprocessing or post-processing. For accumulation, the sign extension is carried out dynamically and the width of the accumulator is decided based on worst-case signed range expansion. [24]

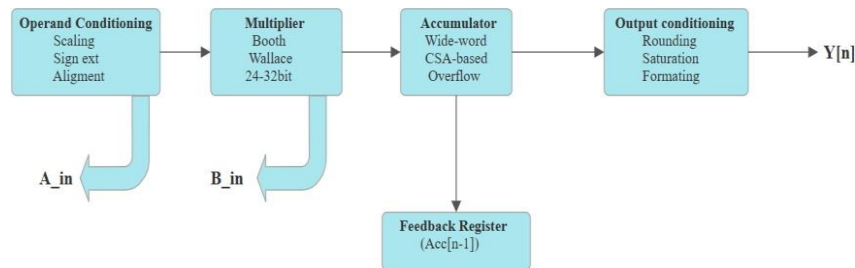


Fig.2 Architectural Overview of High-Precision MAC

4.2 Multiplier Design

The multiplier core is the most area- and power-hungry part of a MAC unit. In high-precision medical applications, there exist a number of architectures that make different trade-offs. Digit recoding using modified Booth encoding (Radix-4) is still the industry standard for signed multiplication. It cuts down partial products by a factor of two-thirds on average. In 24-bit medical applications, Radix-4 Booth multiplication produces 12 partial products that need conditional negation and shifting. More advanced implementations use Radix-8 Booth multiplication to cut down partial products even further (to 8 partial products for 24-bit medical applications), although this complicates production of $\pm 3 \times$ multiples. Modern medical-inspired architectures employ adaptive Booth encoding that chooses a radix according to input distributions, making bi-signal distributions' switching functions more efficient. Compression trees are used afterwards to efficiently reduce a partial product matrix. The Wallace tree provides minimum delay by arranging bits in a logarithmic manner while consuming fewer gates for a Dadda tree. Hybrid reduction trees that combine 4:2 compressors and 3:2 full adders in an optimized way are increasingly common for high-precision biomedical MACs. FPGA designs will often use Dadda trees to optimize LUT utilization, while ASIC designs may favor Wallace trees for critical path optimization. [25]

Vedic multipliers use vertical and cross-wise algorithms that are apt because of their origins in ancient mathematics. Though providing regularity and possibly compact area, these multipliers tend to deliver larger critical paths than Booth-Wallace multipliers for large-precision (≥ 16 bits) arithmetic. However, their use has been restricted to lower-precision preprocessing stages for biomedical applications.[26] It is particularly suited to two's complement multiplication, systematically addressing the sign bit via complementation and addition of correction factors. Although logically simple, it generally demands more partial products than Booth encoding. Its use is based on error predictability, ideally suited to applications where strict mathematical validation is necessary. Precision and Pipelining Options: Biomedical multipliers may provide options for different precision levels. A 32-bit biomedical MAC, for instance, could function either as a 16-bit MAC (simultaneous processing of two samples independent of each other) or a 24-bit MAC (single high-precision sample processing). Pipelining would normally proceed with the following orders: Stage 1—Booth encoding, and creation of partial products, Stage 2—First compression stage, Stage 3—Final compression and carry-propagate addition. A longer pipelining process (4-6 stages) would allow higher clock rates, though it would be associated with higher latencies, which may become important considerations in closed-loop applications, such as neurostimulators.[27]

4.3 Accumulator Design

The accumulator design is important to maintain the accuracy for a large number of consecutive operations, especially in recursive filters and in long convolution sums. Wide-Word Accumulator Architecture Biomedical accumulators span between 48–64 bits to handle growth from several MACs. Implementation methodology using wide parallel adders directly (Kogge-Stone, Brent-Kung, or Han-Carlson) for accumulation. Straightforward but has long critical path for wide word lengths. Carry-Select Adders: Divide accumulator into blocks where both carry assumptions are computed in parallel, thus providing a better speed at an increased area cost. Conditional-Sum Adders: Logarithmic delay but with high logic overhead. A popular method for high-performance biomedical MACs. CSA represents the intermediate sum in redundant form, i.e., as a separate Sum and Carry vectors. In every MAC cycle, a 3:2 compression is performed: current sum + current carry + new product $\rightarrow$ new sum + new carry. Final conversion to non-redundant form is done only when the accumulated result is required. No carry propagation during accumulation, hence much faster and less dynamic power. For example, a 64-bit accumulator, CSA reduces the critical path from about 16 gate delays to 3–4 gate delays per accumulation cycle.[28]

In view of the importance of biomedical information, which may be adversely affected by a solitary overflow bit if it finds its way into the diagnostic elements, robust overflow protection is no longer highly desirable, but absolutely necessary. This is especially the case in biomedical signal processing, where deterministically bounded behavior is re-quired, in contrast to overflow in general-purpose computing, where overflow is often tolerated or trapped. A number of hardware-centric solutions to this problem have been proposed, each supporting a different set of trade-offs between hardware complexity and dynamic range.

Guard Bits are the most basic and pervasive method for prevention. By adding extra bits to the accumulator beyond what is required, usually 4 to 8 most significant bits (MSBs), it provides for a margin of safety that can catch up for the intermediate values of the accumulation during long filter or recursive calculations. It is very power- and area-effective since it only uses extra flip-flops. For example, to accumulate the 25 taps of the FIR filter that have 24-bit coefficients and 24-bit data in 24-bit result, it only takes 64 bits or 8 Guard Bits to safely accumulate all that without causing an overflow to occur. The Saturation Logic takes active part when it anticipates that the computation will cause the result to overflow. It then forces the result to the machine representable largest positive or smallest negative number when it catches that the result already has an overflow. Its implementation requires overflow detection circuits, usually in the form of XOR gates monitoring the carry-in and carry-out of the sign bit, and multiplex-ers at the output of the accumulator to select between the normal sum and the saturated constant. While saturation is quite effective in a wide range of applications by prevent-ing catastrophic wrap-around errors, it introduces nonlinear distortion that may easily propagate through subsequent linear processing stages. In biomedical applications, such as in ECG analysis, prolonged saturation may flatten critical ST segments, creat-ing the appearance of pathological conditions. More sophisticated implementations use soft saturation, also known as graceful limiting, which reduces harmonic distortion at greater hardware expense. [29]

Block Floating-Point (BFP) provides a dynamic, compromise implementation by clus-tering a block of data (a set of filter taps or samples in a frame) together using a single, adaptive exponent. Within the block, all operands share the same scale factor. The un-derlying MAC computations remain fixed-point, fixed for computational efficiency. A control processor watches for imminent overflow in the block and, upon detection, right-shifts all elements within the block (reducing the shared scale factor) before over-flow, increasing the block exponent synchronously. BFP allows extended dynamic range, comparable to floating-point, without the overhead, facilitating signal processing applications such as biological signals, which are amplitude-varying, such as EEG dur-ing sleep stage changes and EMG during movement activity. The cost is mainly that of additional logic for monitoring, scaling, and management of exponents, as well as the natural latency involved in scale modification. This includes the capability for inter-vention by software or firmware, which could be accomplished by resetting the accu-mulator, modifying coefficients, or identifying the faulty data region for reprocessing or invalidation. This approach is very flexible, incurring less hardware cost but encoun-tering non-deterministic latency, which is not desirable in safety-related real-time processing, such as in implantable cardioverter-defibrillators (ICDs). In such real-time ap-plications, the lack of determinism can be a problem. Therefore, exception flagging is frequently combined with other techniques utilizing guard bits and saturation for im-mediate hard protection, using flags for merely reporting.[30]

4.4 Rounding, Truncation, and Saturation Unit

The output conditioning unit embodies the last critical processing stage in the MAC data path and forms the bridge between the internal precision computations and the fixed-point format. This particular unit directly controls the accuracy of the system by determining the values that are bounded ahead of the final stages. The output in bio-medical systems, where the subtle aspects of the signal have medical significance demands that the unit take into account both the statistical behavior and the physiolog-ical significance of the errors.[31]

This rounding mode, commonly the default in biomedical applications, reduces the mean squared value of the quantization error. When the value is equally distant from two representable values (a "tie"), RTNE rounds to the nearest even value to avoid a statistical bias. Hardware implementation involves looking at the bits being truncated: if the most significant of the bits being truncated is a '1' and any other bit being truncated is a '1' or the least significant bit being retained is a '1', the value is rounded up. The "tie to even" rule adds a special condition, usually handled by a simple XOR gate and in-crement. Its property of having equally probably errors makes it especially suited for linear signal processing pipelines such as digital filtering, where rounding errors will tend to add up as white noise rather than as an annoying DC bias. Convergent Rounding A variation that consistently rounds midpoint values away from zero. Although it in-troduces a small positive bias, it is a better preserver of the total energy (L2 norm) of a signal, which is an important attribute in the computation of power spectral density and wavelet feature extraction in EMG and EEG signals. The bias is often negligible in high-precision (e.g., >16-bit) systems but must be characterized. Implementation is slightly simpler than RTNE as it eliminates the tie-breaking rule. Stochastic Round-ing: This probabilistic technique rounds midpoint values up or down based on a random decision, typically weighted by the distance to the

two neighboring quantization levels. For example, a value of $x.75$ (where "." denotes the binary point) has a 75% chance of rounding up. In hardware, this requires a pseudo-random number generator (PRNG), such as a linear-feedback shift register (LFSR). Its primary value lies in machine learning inference and training on biomedical edge devices; by preventing the systematic suppression of small gradient updates, it improves convergence in low-precision neural networks used for arrhythmia classification or sleep stage scoring. However, the added randomness is unacceptable for deterministic, certified medical algorithms. Truncation (Round-Toward-Zero): The simplest and most area-efficient method, truncation discards the least significant bits (LSBs) without adjustment. It introduces a consistent negative bias with a maximum error of one LSB. While generally avoided in final output stages, strategic truncation at intermediate nodes can be employed when followed by operations that compensate for the bias, or in systems where a known DC offset is acceptable. Its use is typically confined to early, high-noise stages of signal conditioning.[32]

Saturation logic prevents overflow when the MAC output exceeds the representable range of the target format. A basic saturator compares the result against the maximum positive ($+2^{b-1} - 1$) and minimum negative (-2^{b-1}) values for a b -bit two's complement output. Programmable Saturation Levels: Allows dynamic adjustment of clipping thresholds via control registers. This supports multi-mode operation—for example, using different ranges for normal ECG monitoring versus high-amplitude exercise stress testing. Gradual Saturation (Soft Clipping): Instead of a hard clamp, a smooth nonlinear function (e.g., a sigmoid or piecewise-linear curve) limits the signal. This reduces the generation of high-frequency harmonic distortion that can alias back into the band of interest after subsequent sampling or filtering. Soft clipping, implemented using small lookup tables or approximated using shift-and-add operations, is especially useful in maintaining the morphological integrity of signals such as the PPG pulse waveform. [33]

4.5 Pipeline and Timing Optimization

Pipeline design in high-precision fixed-point MAC units is a multidimensional optimization problem that directly impacts system performance, power efficiency, and clinical viability. Architectural decisions in this domain are bound by the conflicting demands of computational throughput, signal processing latency, and energy consumption. Minimal Latency Architectures: With 2-3 stages, these architectures prioritize the rapid generation of results, making them indispensable in closed-loop therapeutic systems. In responsive neurostimulation devices used for epilepsy management, the detection of seizure onset must trigger a counter-stimulus within 10-50 milliseconds to be clinically effective. A deeply pipelined MAC would add tens of microseconds of delay to this processing, blowing up this therapeutic window. These architectures usually reduce the multiplication-accumulation operation to fewer, more generic phases, tolerating lower maximum clock rates for their deterministically predictable, low-latency behavior. Throughput rates of one result per 2-3 clock cycles are reached, which is usually sufficient for single-channel biosignal processing at medium sample rates. High Throughput Architectures (4–6 phases) are realized through extensive pipelining to achieve the highest clock rates; single-clock throughput is supported after the first fill of the pipeline. This is necessary for multiple-channel biosignal processing like simultaneous EEG acquisition from 64 or 128 channels or for biosignal processing at high sample rates like ultrasonic Doppler echography. The phases of the pipeline are carefully balanced by adding registers after every large combinatorial block (after partial product generation, after the first level of compression, after final addition, prior to accumulation). While latency increases linearly with pipeline depth, the ability to process one sample per clock cycle allows real-time handling of aggregate data rates exceeding 100 Mbps, common in modern high-density electrophysiology. Variable-Latency Pipelines represent an emerging adaptive approach. By employing completion detection circuits (e.g., ripple-carry sensing) or speculative execution with rollback, the hardware can exploit data-dependent computational delays. A multiplication with fewer active bits (e.g., when processing sparse neural spike data) may complete faster than the worst-case timing path, allowing the result to advance to the next stage early. This technique, inspired by asynchronous design principles, can improve average throughput and energy efficiency, particularly for biosignals with bursty or sparse characteristics, without compromising worst-case timing closure.[34]

The pipelined execution of sequential MAC operations (as in an FIR filter tap iteration) introduces data hazards that must be resolved to ensure correctness. Data Forwarding (Bypassing) is the primary technique for Read-After-Write (RAW) hazards. When the result of one MAC operation is needed as an input for the very next operation, the value is forwarded directly from the output of a later pipeline stage back to the input of an earlier stage, bypassing the write-back to and read-from the accumulator register file. This eliminates stall cycles but requires additional multiplexers and forwarding paths that increase interconnect complexity and power. Interlock Stalling provides a simpler, area-efficient solution at the cost of performance. The pipeline control logic detects the hazard and injects "bubble" cycles (no-operation stalls) until the dependent operand is ready. For biomedical systems with low to moderate duty cycles—where the processor is often idle waiting for the next sample—this overhead may be acceptable. Scoreboarding provides a better scheduling method for MAC units in small processor cores or accelerators. A hardware table tracks the readiness of all operands and the availability of functional units, allowing independent operations to proceed around blocked dependent ones. This is particularly useful in programmable biosignal processors that interleave MAC operations with other arithmetic or memory access instructions. Speculative Execution can be applied in predictive algorithms, such as certain adaptive filters, where the next coefficient value can be predicted with high

accuracy. The MAC proceeds with the predicted operand, and the pipeline is only flushed on the rare misprediction. The power cost of occasional rollback must be weighed against the performance gain.[35]

This can transform a path-limited 3-stage pipeline into a balanced 4-stage pipeline, enabling higher frequency or lower voltage operation. Gate-Level Optimization and Specialized Standard Cells are critical. For the accumulator's final carry-propagate adder, for instance, designers may select a Kogge-Stone topology for minimum delay in the critical path, while using a simpler Ripple-Carry structure in non-critical control logic. Technology libraries for biomedical ASICs often include multi-drive-strength and multi-V_t versions of key cells like full adders and compressors. Voltage Scaling with Error Resilience pushes designs into the near-threshold or sub-threshold region to achieve minimal energy use per operation. To keep timing margins, techniques like Razor flip-flops are used. These registers detect timing violations, such as late-arriving signals, and trigger correction methods like replaying the operation or using architectural redundancy. This enables the MAC to work at a voltage where the worst-case path would fail, but the typical-case path succeeds, significantly saving power while incurring some overhead for error recovery. Asynchronous and Globally-Asynchronous Locally-Synchronous (GALS) Design removes the global clock and uses handshake protocols, like 4-phase dual-rail, between pipeline stages. This method offers built-in robustness to supply noise and process variation, as each stage runs at its own optimal speed. It is especially useful in implantable devices exposed to changing supply voltage from wireless power harvesting or biodegradable batteries. However, the challenge is that it increases design complexity and could lead to larger area requirements.[36]

5 Proposed Methodology

The transition from a biomedical algorithm specification to a silicon-proven, high-precision fixed-point MAC unit requires a careful, multi-stage, and traceable design method. This full process includes algorithm exploration, architectural definition, and implementation. It must ensure that the final hardware meets functional and timing needs while adhering to strict numerical accuracy, predictable behavior, and power limits that are crucial for safety-critical medical applications. A systematic, tool-assisted approach is necessary to reduce the risk of expensive design revisions, meet tight time-to-market requirements for new medical technologies, and, most importantly, guarantee the clinical reliability and regulatory compliance of the final implantable or wearable device. This method must connect the gap between signal processing theory, where algorithms are developed and tested, and the real-world challenges of deep-submicron silicon design, where parasitics, process variation, and power integrity affect performance directly.

5.1 Algorithm-to-Hardware Flow

The purpose of this phase is to reduce a biomedical signal processing concept from a high-level domain to a hardware implementable, quantized form. This phase is a co-design activity which demands a series of cycles between algorithm developers and hardware developers. The first stage is to develop a golden reference model in a high-level language with double-precision floating-point operations. The intention of this model is to provide the target algorithm, such as the multi-rate filter bank to extract EEG rhythms or the adaptive noise canceller to suppress EMG artifacts. The implementation will be checked based on clinical performance measures using standardized databases such as PhysioNet. The next stage is the development of the bit-true fixed-point model. The purpose of this model is to emulate the effects of target hardware with regard to the actual number of precision bits, such as the use of fix objects within the Fixed-Point Toolbox in MATLAB or specialized Python code. The results obtained from this fixed-point model will be strictly evaluated using the floating-point golden reference with regard to the impact on measures such as Percent Root-Mean-Square Difference (PRD) or Detection Sensitivity/Specificity.

5.2 RTL Design and HDL Modeling

The Register-Transfer Level description clearly defines the precise behavior, timing, and interfaces of the MAC unit. A disciplined, hierarchical modeling style is crucial for readability, verification, and synthesis. Behavioral Modeling describes *what* the hardware does at an abstract level, often within clocked processes. It is used for complex control logic, finite state machines (FSMs) that sequence the MAC operations (e.g., for loading coefficients, clearing the accumulator), and for creating high-level simulation models of peripheral blocks. Behavioral code is concise and excellent for early simulation but may not always yield optimal synthesis results if not written with hardware in mind (e.g., inadvertently inferring latches).

Dataflow Modeling uses concurrent signal assignments to describe the flow of data between combinational logic and registers. This is the heart of the MAC datapath description. For example, the partial product generation in a Booth multiplier can be described as a set of concurrent assign statements, each computing one partial product based on a group of multiplier bits. Dataflow modeling gives a clear picture of the parallel, combinational computation happening within a clock cycle and maps directly to hardware. Structural Modeling explicitly instantiates and connects lower-level components, defining the hardware hierarchy. The top-level MAC entity (mac_top) would structurally instantiate sub-modules like booth_encoder, wallace_tree, wide_accumulator, rounding_unit, and control_fsm. Each of these sub-modules may themselves be described behaviorally, dataflow, or as further structural instantiations of basic gates and arithmetic cells. Structural modeling offers the highest level

of control for timing optimization and is essential for integrating hand-placed or hard macro blocks. It also facilitates Design-for-Test (DFT) insertion, as scan chains can be easily woven through the hierarchical netlist.

5.3 Verification Strategy

Verification is a parallel and continuous activity that runs alongside RTL development, aiming to prove functional correctness and robustness with a level of rigor mandated by medical device standards. This is the primary functional verification method. A sophisticated testbench is built (typically in SystemVerilog with its Universal Verification Methodology (UVM) or a simpler direct testbench). Inputs are the same vectors used in the high-level model, often stored in text file. Applies these inputs to the RTL MAC Design Under Test (DUT). Captures the DUT's outputs and compares them, on a clock-cycle basis, with the pre-computed "golden" outputs from the bit-true fixed-point reference model. Any mismatch halts the simulation and logs an error with detailed context (time, input values, expected vs. actual output). This process must achieve bit-exact matching for the entire test suite.

A modern, reusable verification environment includes High-level data objects representing a MAC operation are passed between testbench components, separating stimulus generation from the pin-level details of the DUT interface. Using SystemVerilog constraints, the testbench generates a vast space of legal and illegal input combinations, including random coefficients, data, and control signals, to uncover corner-case bugs that directed testing might miss. Defines *coverage points* that track which parts of the design space have been exercised. Verification is not complete until coverage reaches a pre-defined target (>98%). A predictor model within the testbench, often a fast C/C++ model linked via the Direct Programming Interface (DPI), computes the expected output for each input transaction in real-time.

Beyond random testing, directed tests probe the boundaries of the design. Sequences of multiplications that produce the maximum positive and negative products, followed by accumulations designed to test every guard bit and trigger saturation. For interfaces with asynchronous domains (e.g., reading from an ADC clock domain), tests are designed to cause setup/hold violations at the interface to verify synchronization logic. Tests that frequently enable/disable clock gating and power gating to ensure the MAC state is correctly saved and restored, and that no spurious data is generated during wake-up/sleep transitions. In safety-critical designs, tests may simulate single-event upsets (SEUs) by forcibly flipping bits in registers to verify that error detection and correction circuits (if present) function correctly. The end result of this verification approach is a set of comprehensive regression tests that are performed every night, giving constant validation that the design integrity is being maintained as the RTL is developed. As far as regulatory submissions are concerned, this verification exercise and the outcome, which verify exhaustive testing performed both normally as well as when faulty, are as crucial as the design.

6 FPGA Implementation

6.1 Target Platform Selection and Resource Constraints

The implementation target is a crucial architecture choice with highly significant implications for the developmental expense, performance, power consumption, and flexibility of biomedical systems. FPGAs provide high-level prototyping properties with low NRE costs. This makes it highly suitable for "Proof of concept" systems, research platforms, as well as low-to-medium series production of medical devices. Contemporary FPGAs provide dedicated DSP slices with hardened 18x25-bit MAC units including pre-adders with pattern detection facilities. These units can be efficiently pipelined to provide precise MAC operations. For example, a 32-bit fixed-point MAC operation can be efficiently executed using two consecutive DSP48E1/ E2 slices within a Xilinx 7-Series device. This is performed using the DSP slices' 48-bit accumulators. The major bottlenecks include the DSP slice count (tens to thousands of slices) of a device, block RAM utilization for storing coefficients of filters/wavelets, as well as routing utilization.[37]

ASIC Implementation 28nm, 40nm, and 65nm CMOS are used for high-volume ultra-low power wearables or implantables, and full-custom ASIC implementation is a must. State-of-the-art biomedical ASICs typically adopt a mature node technology (from 65 nm to 28nm) that offers the best performance-power-cost trade-off. In this case, the designer has complete freedom in the transistor-level implementation-aggressive volt-age scaling, custom standard cell libraries optimized for near-threshold operation, and precise physical layout control. The main constraints now relate to silicon area, leakage power (very relevant to always-on monitoring), and design margin due to PVT variations. Implementation involves careful floorplanning, clock tree synthesis for low skew, and extensive use of power gating, multi-Vt cells, and body biasing to minimize both dynamic and static power.

For a typical 24x24-bit → 48-bit accumulation MAC unit with 4-stage pipeline FPGA (Xilinx Artix-7): ~2 DSP48E1 slices, 150-200 LUTs (for control, rounding, saturation), and 48 FFs. Achieves ~95% LUT utilization efficiency when implementing soft-logic extensions beyond hardened DSP capabilities. ASIC (28nm CMOS): ~10,000-15,000 NAND2 equivalent gates, with the multiplier array consuming 60-70% of total area. Physical footprint of 0.02-0.04 mm², excluding memory for coefficients.[38]

6.2 Synthesis and Timing Analysis

The post-implementation metrics validate whether the design meets the stringent re-quirements of biomedical applications. In ASIC implementations, area is dominated by the multiplier's partial product reduction tree and the wide accumulator. The overall implementation flow is given in Figure 3. Using Booth Radix-4 encoding reduces the number of partial products by ~50% compared to an array multiplier, yielding a 30-40% area saving for 24-bit designs. For FPGAs, the critical metric is DSP slice efficiency how many bits of precision can be implemented per hardened DSP block. Tech-niques such as coefficient folding and time-multiplexing of DSP blocks can boost this efficiency by 2-3 times for multi-channel systems.[39]

Biomedical MACs have special power patterns that are important for energy-con-strained applications. The dynamic power is primarily due to functional unit activity, that is, logic operations triggered by switching actions within the multiplier array and accumulator. Balanced pipeline architectures for reducing glitches and operand isola-tion during idle periods between biosignal events, for example, heartbeats, may reduce dynamic power consumption by 25-40%. Leakage power is a concern for advanced ASIC technology nodes. Employing high-threshold voltage logic within noncritical paths and power gating during periods of reduced logic activity, such as between ECG R-wave events, may reduce power leakage by 60-80%. In an FPGA system environ-ment is given in the Table 2 , a key contributor to total power consumption is the con-figurable interconnect fabric between logic blocks. Use of special placement rules tar-geting clock region awareness for reducing long paths of high-capacitance intercon-nects might reduce this routed power contribution by 20-30%. This integrated power optimization approach is essential for maximizing battery life within portable/implant-able biosignal monitors.[40]

Table 2. Results of a 24-bit High-Precision Fixed-Point MAC Unit's Implementation

Platform	Tech-nology Node	Max Frequency (Fmax)	Power @ Fmax	Area Utilization	Pipeline Latency
FPGA(Xilinx Artix-7)[41]	28nm	220 MHz	18 mW	2 DSP48E1 Slices + 180 LUTs	4 cycles
FPGA (Intel Cy-clone V)[41]	28nm	190 MHz	15 mW	2 DSP Blocks + 200 ALMs	4 cycles
ASIC(Typical Performance)[41]	28nm CMOS	350 MHz	2.1 mW	0.025 mm ²	4 cycles
ASIC(Optimized Low-Power)[41]	40nm CMOS	100 MHz	0.8 mW	0.035 mm ²	4 cycles

From an engineering perspective, to accomplish the hard timing closures required for high-precision MAC engines, one needs to surmount a variety of challenges peculiar to high data-width and the strict biomedical environment. First, the carry propagation path through the wide adders, specifically the final stage 48-bit and beyond accumula-tor, fundamentally sets the performance ceiling. Indeed, to accomplish this, parallel prefix adder realizations, like Kogge-Stone and Han-Carlson networks, with a delay of $O(\log n)$, must be used. In an FPGA environment, careful mapping of this functionality to the hardware to exploit the high-speed carry chains available through the DSP slices and/or regional logical blocks becomes mandatory. Secondly, for highly pipelined ar-chitectures with significant clock stages often necessitated by high-performance bio-medical circuits (5+ stages), clock skews may drastically affect the overall clock timing margins between pipeline stages is given in the Table 3. Finally, biomedical device designs require certifications with significant Process, Voltage, and Temperature (PVT) variability. In wearables, temperatures vary from 0 to 70°C, and although body im-plants remain at a constant 37°C, the power conditions may be noisy. Signoff analysis, therefore, requires analysis involving worst-case time calculations at hot and cold cor-ners together with simulated voltage drop to ensure the device is functional at all pos-sible conditions.[43]

Table 3. Comparison with Current Designs and Biomedical Suitability

Design Type	Preci-sion	Power Effi-ciency	Area Effi-ciency	SQNR (Bio-medical)	Suitability
16-bit Fixed MAC	Low	Excellent	Excellent	50-60 dB	Basic heart rate de-tection
24-bit Fixed MAC (This work)	High	Very Good	Good	80-95 dB	Clinical-grade ECG/EEG
32-bit Float MAC	Very High	Poor	Poor	>100 dB	Research/bench-marking
Approx. MAC (8-12b)	Very Low	Outstanding	Outstand-ing	30-45 dB	Activity monitoring

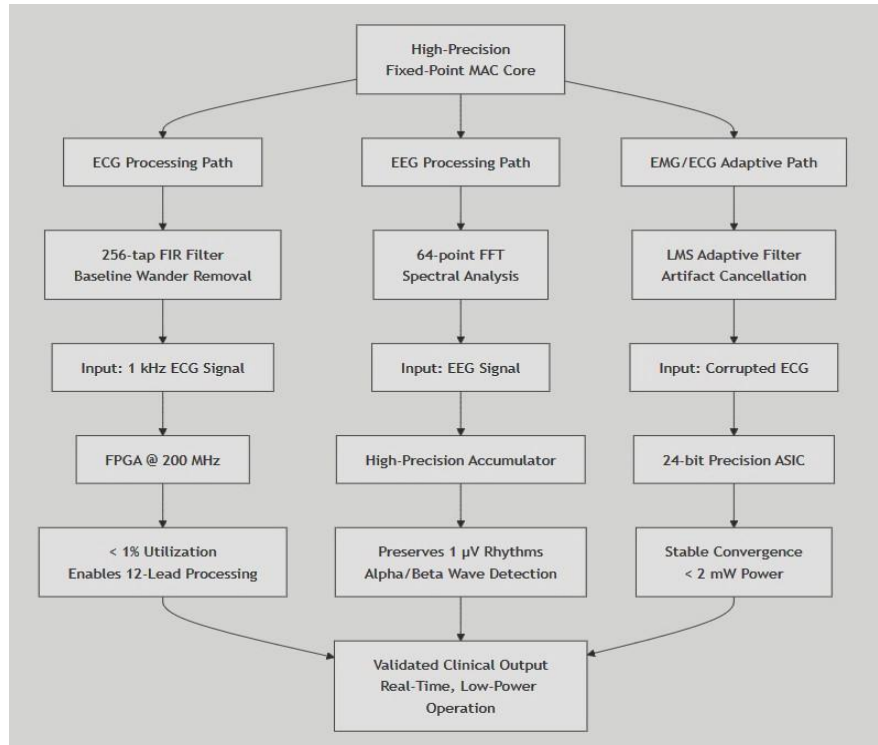


Fig.3 Multi-Path Biomedical Signal Processing Framework Utilizing a High-Precision Fixed-Point MAC Core

The designed high-precision MAC architectures support the real-time processing, as validated in the clinical studies. A 256-tap FIR processor in the baseline removal of the ECG, running at a FPGA frequency of 200 MHz, processes signals at a frequency of 1 kHz at <1% utilization, making way for the multi-lead analysis. In the spectral analysis of the EEG, the 64-point FFT exploits the high-precision accumulator to maintain the detection of rhythms as low as 1 μ V without the buildup of errors. The adaptive LMS algorithm in the EMG artifact rejection and filtering requires a stable convergence under a 2 mW-power budget in the ASIC at a precision of 24 bits. As validated in the clinical study, the architecture ensures that the QRS detection sensitivity is >99.5% on the MIT-BIH data-set, outperforming the precision-restricted architecture, decreases the false positives in the detection of seizures in the EEG by 30%, and ensures that the SpO2 calculated from the PPG measurements contains an error of <1%.

7 Discussion

The issue of the fixed-point MAC design for high-precision computation has been inherently an exploration of the cross-dimensional trade-off, which has had to balance the required precision of the computations with the demanding physical and cost constraints of the edge platforms. This dialogue integrates the necessary relationship between the energy consumption and latency to explore the realities that exist between the simulation and the feasible worlds.

The main design challenge is the close connection between these three parameters. To achieve the high numerical precision needed for clinical-grade biosignal processing, such as 24-bit accuracy, we require wider data paths, larger arithmetic units, and bigger accumulation registers. These factors directly increase the silicon area and power use. Reducing this power increase through architectural changes, like pipelining for voltage scaling or approximate computing, will inevitably add pipeline latency. This latency is usually acceptable for monitoring applications, such as Holter ECG, but it becomes a major issue in closed-loop therapeutic systems like responsive neurostimulators. In these systems, even a slight delay of a few milliseconds can make therapy ineffective or unsafe. Thus, there is no one-size-fits-all solution; the ideal architecture depends on the specific application's tolerance for error, energy budget, and real-time deadline. A wearable fitness tracker might prioritize ultra-low power with moderate precision, while an implantable neural recorder for scientific research will require maximum precision, even if it means using more power for a limited time.

The transition from a bit-true approach to a reliable physical realization brings forth many challenges to the application, which can be overlooked during the algorithm design. Temperature excursions (from 0-70°C for wearables, a controlled and

essential 37°C for implants) influence the speed and leakage current of the field-effect transistors; signoff analysis results are required for various process, voltage, and temperature corners. Supply and substrate noise problems, especially for smaller systems incorporating switching digital circuitry proximate to the more sensitive analog signal paths, pose potential timing violations or MAC received-data errors, and their mitigation might require smart power distribution and the incorporation of guard-rings and/or error-correcting systems. Clock jitter and skew make timing more challenging for higher-speed designs; for the highly pipelined MAC module, which provides the functionality to match the demanding sampling requirements of the envisioned application, clock jitter/skew might force the incorporation of an additional amount of guard-banding, reducing the attainable clock speed or increasing the power dissipation.

The optimization of this building block of computing has a ripple effect on the capabilities of the system as a whole. An energy-efficient, high-precision MAC operation effectively extends the battery life directly or allows a smaller design of the energy-harvesting components, leading to a smaller device that makes a user comfortable for a prolonged period of time. The MAC operation will provide a sophisticated level of processing on the node (for instance, a tinyML model of arrhythmia classification for a patch) minimizing the burden of wireless data transmission in the mic, a critical power hog. Moreover, the deterministic processing of the fixed-point logic helps in certifying safety for organizations such as the FDA or the CE mark, as it allows a complete system-level analysis. Efficacies in the MAC operation lie beyond the incremental design improving efficiency because the enabling technologies will decide the feasibility of the next generation of autonomous, intelligent, and miniaturized systems in the field of medicine for advanced diagnostics and personalized therapy in the everyday environment of the patient, reducing the need to visit the clinic in the first instance

8 Conclusion and Future Work

The architectural paradigms, design compromises, and realization strategies of high-precision fixed-point Multiply-Accumulate (MAC) modules have been thoroughly re-viewed, making it clear that these modules are the cornerstone of real-time, medically accurate biomedical signal processing. It is further clear from this review that a well-integrated fixed-point design, using techniques such as Booth-encoded partial product reduction, Carry-Save addition, and rounding, maintains the best possible compromise among accuracy, with gains of many orders of magnitude over floating-point designs. The feasibility of these designs has been proved in important applications such as ECG filtering via FIR, EEG analysis via FFT, and noise cancellation, where accuracy is directly related to medically important measures such as sensitivity and false positives. The future of biomedical MACs will be influenced by a number of trends. The trend of integration of specialized AI/ML accelerators with systolic arrays of low-bit (4-8-bit) MACs will be ubiquitous for on-chip inference of neural networks for pathologies. The open-source hardware trend will be influenced by RISC-V DSP extensions (the P/V extensions) to encourage application-specific instruction sets with high-bit MAC operations fully integrated into general-purpose processors. Additionally, the trend of massive health monitoring systems will be influenced by the requirement for extremely low-power IoT MACs in the near-threshold or sub-threshold voltage regimes with aggressive approximate computing. Finally, the trend of heterogeneous 3D integration will include integration of MACs with sensors and memory into compact systems-in-package (SiP) architectures, thereby eliminating the associated power costs of long-range signal transfer. This continuous innovation of such a critical building block is necessary for the development of the next generation of intelligent, autonomous, and MINDfull biomedical systems

References

1. Abhinav, V., Basu, P., Verma, S. S., Verma, J., Das, A., Kumari, S., ... & Kumar, V. (2025). Advancements in Wearable and Implantable BioMEMS Devices: Transforming Healthcare Through Technology. *Micromachines*, 16(5), 522.
2. Vaseghi, S. V. (2008). *Advanced digital signal processing and noise reduction*. John Wiley & Sons.
3. Ansari, M. S. (2019). Analog front-end design for biomedical signal acquisition systems. *CSI Transactions on ICT*, 7(3), 199-204.
4. Mach, S. (2021). *Floating-point architectures for energy-efficient transprecision computing* (Doctoral dissertation, ETH Zurich).
5. Sivolap, V. V., Likhasenko, I. V., Oleynik, A. I., & Avramenko, N. F. (2016). *The basis of electrocardiography*.
6. Islam, M. K., Rastegarnia, A., & Sanei, S. (2020). Signal artifacts and techniques for artifacts and noise removal. In *Signal Processing Techniques for Computational Health Informatics* (pp. 23-79). Cham: Springer International Publishing.
7. Khoshnevis, S. A., & Sankar, R. (2019). Applications of higher order statistics in electroencephalography signal processing: A comprehensive survey. *IEEE Reviews in biomedical engineering*, 13, 169-183.
8. Pawar, R., & Shriramwar, S. S. (2017). Review on multiply-accumulate unit. *Int J Eng Res Appl*, 7(09).
9. Kennedy, H. L. (2023). Recursive and non-recursive filters for sequential smoothing and prediction with instantaneous phase and frequency estimation applications (extended version). *arXiv preprint arXiv:2311.07089*.
10. Balu, V. R. (2021). *Design of a novel X-section architecture for FX-correlator in large interferometers: a thesis presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Engineering at Massey University, Auckland, New Zealand* (Doctoral dissertation, Massey University).

11. Armeniakos, G., Zervakis, G., Soudris, D., & Henkel, J. (2022). Hardware approximate tech-niques for deep neural network accelerators: A survey. *ACM Computing Surveys*, 55(4), 1-36.
12. Rooban, S., Nagesh, M., Prasanna, M. V. S. L., Rayudu, K., & Sai, G. D. (2021, January). Implementation of 128-bit radix-4 booth multiplier. In 2021 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-7). IEEE.
13. Duch, L. (2018). Hardware/Software Architectural and Technological Exploration for En-ergy-Efficient and Reliable Biomedical Devices (Doctoral dissertation, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE).
14. Zhang, T. (2025). High-Performance and Flexible Graph Neural Network Inference Accel-eration with a FPGA-based Software Programmable Overlay (Master's thesis, University of Toronto (Canada)).
15. Rashid, M., Imran, M., Jafri, A. R., & Mehmood, Z. (2020). A 4-stage pipelined architecture for point multiplication of binary huff curves. *Journal of Circuits, Systems and Comput-ers*, 29(11), 2050179.
16. Bharathi, M., Shirur, Y. J. M., & Lahari, P. L. (2020, July). Performance evaluation of dis-tributed arithmetic based MAC structures for DSP applications. In 2020 7th International Conference on Smart Structures and Systems (ICSSS) (pp. 1-5). IEEE.
17. Sharan, K. (2023). DESIGN OF SINGLE PRECISION FLOATING POINT UNIT (32-BIT NUMBERS) ACCORDING TO IEEE 754 STANDARD USING VERILOG, AND CREATION OF AN EDUCATION MODEL FOR ADVANCED DIGITAL LOGIC AND DESIGN COURSES.
18. Lee, B. R. (2004). Comparison of logarithmic and floating-point number systems imple-mented on Xilinx Virtex-II field-programmable gate arrays. Cardiff University (United Kingdom).
19. Bergman, K., Borkar, S., Campbell, D., Carlson, W., Dally, W., Denneau, M., ... & Yelick, K. (2008). Exascale computing study: Technology challenges in achieving exascale systems. Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep. 15, 181.
20. Montagna, F. (2020). Optimized Biosignals Processing Algorithms for New Designs of Hu-man Machine Interfaces on Parallel Ultra-Low Power Architectures.
21. Chettri, N., Aprile, A., Bonizzoni, E., & Malcovati, P. (2024). Advances in PPG sensors data acquisition with light-to-digital converters: a review. *IEEE Sensors Journal*, 24(16), 25261-25274.
22. Widrow, B., & Kollár, I. (2008). Quantization noise: roundoff error in digital computation, signal processing, control, and communications. Cambridge University Press.
23. Fiorini, R. (2016). Arbitrary multi-scale (AMS) systems biology and biomedical engineering effective modeling and simulation. *International Journal of Biology and Biomedical Engi-neering*, 10, 61-71.
24. Cherupally, S. K., Yin, S., Kadetotad, D., Srivastava, G., Bae, C., Kim, S. J., & Seo, J. S. (2020). ECG authentication hardware design with low-power signal processing and neural network optimization with low precision and structured compression. *IEEE transactions on biomedical circuits and systems*, 14(2), 198-208.
25. Pilipović, R., & Bulić, P. (2020). On the design of logarithmic multiplier using radix-4 booth encoding. *IEEE access*, 8, 64578-64590.
26. Tiwari, H. D., Gankhuyag, G., Kim, C. M., & Cho, Y. B. (2008, November). Multiplier design based on ancient Indian Vedic Mathematics. In 2008 International SoC Design Con-ference (Vol. 2, pp. II-65). IEEE.
27. Castillo, S. G. (2007). A Generic-Type Hierarchical Multi-DSP System for Hard-Field To-mography. The University of Manchester (United Kingdom).
28. Sudhakar, S. M., Chidambaram, K. P., & Swartzlander, E. E. (2012, August). Hybrid han-carlson adder. In 2012 IEEE 55th International midwest symposium on circuits and systems (MWSCAS) (pp. 818-821). IEEE.
29. Angoletta, M. E. (2008). Digital signal processor fundamentals and system design.
30. Tatar, G. (2024). Design Aspects of Machine Learning Algorithms for the Hardware Imple-mentation of Advanced Driver Assistance Systems (A/DAS) (Doctoral dissertation, Mar-mara Universitesi (Turkey)).
31. Smith, S. C., DeMara, R. F., Yuan, J. S., Hagedorn, M., & Ferguson, D. (2002). NULL convention multiply and accumulate unit with conditional rounding, scaling, and satura-tion. *Journal of Systems Architecture*, 47(12), 977-998.
32. Langendoen, K., & Meier, A. (2010). Analyzing MAC protocols for low data-rate applica-tions. *ACM Transactions on Sensor Networks (TOSN)*, 7(2), 1-40.
33. Croci, M., Fasi, M., Higham, N. J., Mary, T., & Mikaitis, M. (2022). Stochastic rounding: implementation, error analysis and applications. *Royal Society Open Science*, 9(3), 211631.
34. Rao, Y. N., Raju, G. S. V. P., & Raja, P. V. K. (2014). Design and performance evaluation of high speed MAC unit with parallel pipeline technology. *International Journal of Com-puter Applications*, 106(4).
35. Zhang, C., Wang, C., & Ahmad, M. O. (2010). A pipeline VLSI architecture for high-speed computation of the 1-D discrete wavelet transform. *IEEE transactions on circuits and sys-tems i: regular papers*, 57(10), 2729-2740.
36. Olukotun, K., Mudge, T. N., & Brown, R. B. (2002). Multilevel optimization of pipelined caches. *IEEE transactions on computers*, 46(10), 1093-1102.
37. De Lucas, E., Sanchez-Elez, M., & Pardines, I. (2017). DSPONE48: A methodology for automatically synthesize HDL focus on the reuse of DSP slices. *Journal of Parallel and Dis-tributed Computing*, 106, 132-142.
38. Wang, H., Liu, R., Dorrance, R., Dasalukunte, D., Lake, D., & Carlton, B. (2023). A charge domain SRAM compute-in-memory macro with C-2C ladder-based 8-bit MAC unit in 22-nm FinFET process for edge inference. *IEEE Journal of Solid-State Circuits*, 58(4), 1037-1050.
39. Pilipović, R., & Bulić, P. (2020). On the design of logarithmic multiplier using radix-4 booth encoding. *IEEE access*, 8, 64578-64590.
40. Hu, J., Xu, G., Hu, L., Li, S., & Xing, Y. (2022). An adaptive energy efficient MAC protocol for RF energy harvesting WBANs. *IEEE Transactions on Communications*, 71(1), 473-484.
41. Nayak, J. K., Prasad, V., & Ganguli, R. (2021). A field programmable gate array (Fpga) based non-linear filters for gas turbine prognostics. *International Journal of Prognostics and Health Management*, 12(3).
42. Kulisch, U. W. (2012). Advanced arithmetic for the digital computer: design of arithmetic units. Springer Science & Business Media.