A 22nm All-Digital Time-Domain Neural Network Accelerator for Precision In-Sensor Processing

Ahmed M. Mohey, Jelin Leslin, Gaurav Singh, Marko Kosunen, Jussi Ryynänen, and Martin Andraud

***Abstract*—Deep Neural Network (DNN) accelerators are increasingly integrated into sensing applications, such as wearables and sensor networks, to provide advanced in-sensor processing capabilities. Given wearables' strict size and power requirements, minimizing the area and energy consumption of DNN accelerators is a critical concern. In that regard, computing DNN models in the time domain is a promising architecture, taking advantage of both technology scaling friendliness and efficiency. Yet, time-domain accelerators are typically not fully digital, limiting the full benefits of time-domain computation. In this work, we propose an all-digital time-domain accelerator with a small size and low energy consumption to target precision in-sensor processing like human activity recognition HAR. The proposed accelerator features a simple and efficient architecture without dependencies on analog non-idealities such as leakage and charge errors. An 8-neuron layer (core computation layer) is implemented in 22nm FD-SOI technology. The layer occupies 70 µm × 70 µm while supporting multi-bit inputs (8-bit) and weights (8-bit) with signed accumulation up to 18 bits. The power dissipation of the computation layer is 576 µW at 0.72V supply and 500MHz clock frequency achieving an average area efficiency of 24.74 GOPS/mm^2 (up to 544.22 GOPS/mm^2), an average energy efficiency of 0.21 TOPS/W (up to 4.63 TOPS/W), and a normalized energy efficiency of 13.46 1b-TOPS/W (up to 296.30 1b-TOPS/W)**

***Index Terms*—Edge Computing, Human Activity Recognition HAR, Inertial Measurement Unit IMU, In-Sensor Processing, Multiply-and-Accumulate MAC, Neural Network Accelerator, Smart Sensor Interface, Time-Domain Signal Processing**

I. Introduction

The development of embedded sensing applications is crucial to continue advancements in areas such as the Internet of Things (IoT), autonomous driving and more generally new usage for connected objects around us. In particular, sensing and data processing based on inertial measurement units (IMU) is a widely used principle in various applications, for instance in wearables for the general consumer market, in biomedical and health monitoring applications, or in automotive. Many of these applications are evolving towards sensory systems that combine accurate sensors with advanced signal processing or machine-learning (ML) capabilities to develop more intelligent sensing systems in e.g. motion or activity recognition. An example is human activity recognition (HAR), based on smart wearable sensors [1], [2], which is getting popular for the daily monitoring of health data (sleep, recuperation) and physical activities [3]. Fig. 1 shows an example of ML applied to HAR,

The authors are with the Department of Electronics and Nanoengineering, Aalto University, Espoo, Finland. Martin Andraud is also with UCLouvain, ICTEAM, Louvain-la-Neuve, Belgium (Corresponding author: Ahmed M. Mohey (e-mail: ahmed.mohey@aalto.fi).

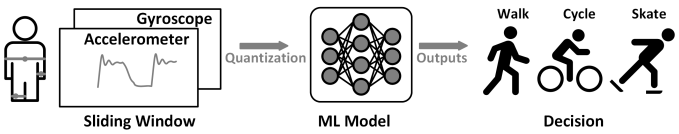


Fig. 1: ML applied to HAR [4]

to classify a range of activities performed by a user equipped with a wearable [4]. Here, the system starts by extracting features from the IMU sensors (a gyroscope and accelerometer) and dividing them into sliding windows. These features can consist of e.g. statistical characteristics of the signal, such as mean, variance, or peak detection. Then, the obtained features are quantized and fed to a classifier that detects the current activity.

Optimizing the energy efficiency and the form factor of embedded sensing systems can be realized with in-sensor processing, where the sensor device is directly intercoupled with advanced data processing capabilities offered by ML, particularly deep learning and deep neural network (DNN) models. However, the hard constraints required for the design of typical low-power tiny wearables necessitate utilizing highly efficient hardware to execute these DNNs. In this context, dedicated hardware accelerators are necessary to execute computation-hungry DNN models. These accelerators typically focus on executing efficient Multiply and Accumulate (MAC) operations, which are at the core of every DNN computation (i.e., every neuron performs a weighted sum of its inputs). The MAC operation realized by one neuron is expressed as:

$$MAC = \sum_{i=1}^{N} X_i W_i, \tag{1}$$

with N is the number of connected neurons, X_i refers to each neuron's inputs and W_i represents the synaptic weights. In classical processors, the MAC computation is heavily dominated by memory transfers, as these processors have separate calculation and memory units. Dedicated accelerators solve this bottleneck by bringing the MAC operation closer to memory, using Near-Memory Computing approaches, or directly inside the memory, using Compute-In Memory approaches [5]. As most DNN inference routines can be accommodated with low-resolution integer computation of 5-8 integer bits [6], [7], MAC operations can be realized either in the digital or analog domain, which implies design trade-offs. Analog-domain MAC has shown better efficiency than digital implementations for low bit-width (up to 6-8 bits integer) [8] but falls steeply for higher bit-widths, due to noise constraints requiring 4 times more power

per extra bit of precision [6]. Hence, digital-domain MAC is typically preferred for resolutions higher than 6-8 integer bits using their lower susceptibility to noise and better scaling properties. In this context, similar to other circuit blocks such as ADCs or sensor interfaces, time-domain implementations of MAC operations could offer a trade-off between analog and digital computing methods. Indeed, time-domain MACs can be implemented using mostly digital circuits, benefiting from scaling; they consume less power than digital MACs due to lower switching activities (thanks to the reduction of digital buses) [9], [10]. Hence, multiple time-based computing methods have been proposed [11]–[16]. Time-based MAC accelerators to attain energy efficiencies of 10-100 TOPS/W [11], [12], [16], which are among the best reported for AI accelerators [17]. However the time-based nature limits the throughput of the system to 0.1-5 GOPS, hence time-based MACs are generally suited for applications requiring efficient computing but lower throughput requirements, for instance HAR. Despite these good results, challenges remain in the development of time-based MAC accelerators. For instance, most existing implementations are not fully digital and rely on analog circuits for signal generation. Hence these blocks suffer from analog non-idealities, in particular leakage and charge errors, which can limit the accuracy of the system. In addition, they exhibit a tendency towards increasing complexity when multi-bit inputs/weights are enabled requiring a relatively complex design of the key components.

To solve these issues, this paper presents an all-digital neural network accelerator that utilizes time-domain approaches to realize highly efficient multi-bit MAC operations with no dependency on analog non-idealities. It consists of an array of combined 8 MAC units $8/8/18$ (neurons) to act as the core processing layer in a Multi-layer Perceptron (MLP) NN for HAR. The UCI HAR dataset [18] is used to verify the capability of the proposed architecture to process precision sensing data with signed accumulation up to 18 bits while achieving more than 90% classification accuracy. The main characteristics of this design are:

- A native precision-scalable and multi-bit support (up to 8 bits), allowing to tailor design as per the application's needs, thereby saving area and power.
- A sequential processing of input features, enabling the use of an arbitrary large number of features for the system without redesigning the architecture. For this reason, the output can accumulate up to 18 bits.
- A simple design and compact footprint to achieve a state-of-the-art area per MAC for time-based accelerators of $612.5\,\mu m^2$. It relies on a simple design based on a single clock source in an event-triggered (on-need) activation. Additionally, dynamic frequency scaling can be deployed for extra power saving at low computation demands. These features ease the integration of many cores for parallel in-sensor processing applications.

The proposed circuit is simulated using a 22nm FD-SOI technology. The layer only occupies $70\,\mu m \times 70\,\mu m$, and it dissipates $576\,\mu W$ at 0.72V supply and 500MHz clock frequency. It achieves an average operation rate of 0.12 GOPS (up to 2.67 GOPS), measuring an average energy efficiency of 0.21 TOPS/W (up to 4.63 TOPS/W), and an area efficiency of 24.74 GOPS/mm^2 (up to 544.22 GOPS/mm^2). It achieves also an average normalized energy efficiency [19] of 13.46 1b-TOPS/W (up to 296.30 1b-TOPS/W).

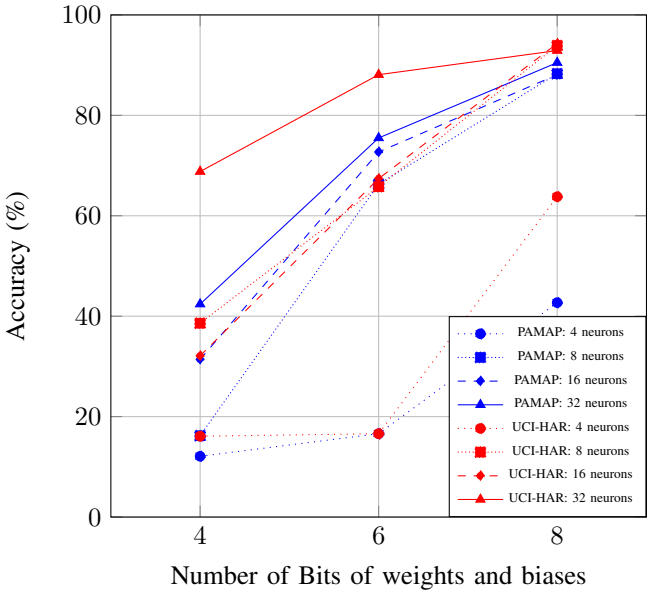


Fig. 2: Model Accuracy vs. Number of Bits in Weights and Bias for different Neuron configurations

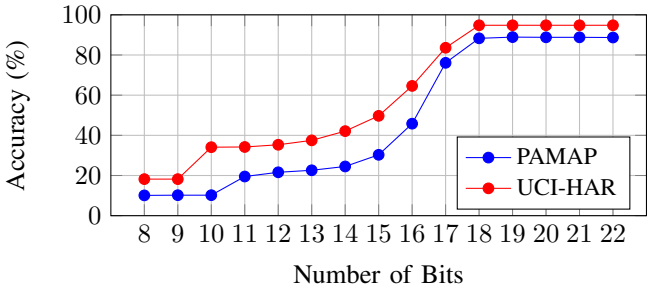


Fig. 3: Model Accuracy vs. Number of Bits in HW Accumulator

This paper is organized as follows. Section II discusses the target application (HAR), including model training, quantization, and optimization. Section III reviews the recent development of time-domain neural accelerators. Section IV presents the design objectives and the proposed architecture. The circuit implementation and simulation results are discussed in section V. A conclusion is given in section VI.

II. Target Application: Online Activity Recognition

The proposed accelerator will be evaluated on online activity recognition applications. These applications require moderate inference speed (in the order of hundreds of milliseconds) but high energy efficiency to be run in the background of portable devices such as wearables or smartphones. Two different activity recognition benchmarks will be compared, the UCI Human Activity Recognition (HAR) [18] containing 6 activities, and PAMAP containing 18 output activities [20]. In

the following, we highlight the analysis for the HAR dataset, yet a similar analysis has been performed for PAMAP and illustrated in Figs. 2 and 3.

A. Human Activity Recognition (HAR)

The HAR dataset was collected from experiments carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (walking, walking upstairs, walking downstairs, sitting, standing, and laying) wearing a smartphone (Samsung Galaxy S II) on the waist. The smartphone's embedded accelerometer and gyroscope were used to capture 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. From these acquired signals, a large number of features can be extracted with pre-processing blocks, typically involving noise filtering, applying various time and frequency domain transformations, and extracting statistics and other information like mean, variance, max, and entropy from these signals. These pre-processing steps are designed to capture the characteristics of human activities in a form that is suitable for machine learning (ML) models. Although many ML models performed well on this dataset, enhancing the hardware efficiency typically requires heavily quantized data and computation with integer numbers, with the constraint of doing the inference in integer hardware. In that regard, ML models such as decision trees, SVM, forests, and regression can have their performance drastically decreased under heavily quantized weights. On the other hand, Neural networks have shown strong resilience and efficiency after quantization [21]. Given that the HAR dataset is composed of time-series data, which are essentially 1-dimensional, classical Multi-Layer Perceptrons (MLPs) can be aptly suited. MLPs can model relationships in sequential data effectively without necessitating the spatial context exploited by 2D Convolutional Neural Networks (Conv2D) or more advanced architectures like recurrent neural networks. In addition, the relative simplicity and lower computational demands of MLPs make them an efficient choice for HAR.

B. Model training and quantization

The HAR dataset is composed of 561 features extracted to describe the activity window. As detailed earlier, the chosen model is an MLP computing all 561 input features, where each neuron necessitates accumulating 561 multiplications. This accumulation of numerous products leads to substantial growth in the dynamic range of intermediate values, up to 16 bits [7]. Various models address this accumulation problem within a limited-resources context (e.g. a wearable device) [22], [23]. These works highlight that quantizing weights and biases to lower bitwidths reduces the model's computational burden, yet the inference is performed on a 32-bit microcontroller, thereby setting the maximum dynamic range of intermediate values to 32 bits. This computation bottleneck is solved in the proposed architecture by performing a successive accumulation of all the features in a single high-resolution accumulation block, accommodating a large dynamic range. The accumulator is the only computation block with higher resolution, as detailed in section IV.

The MLP model is composed of an input layer of 561 features, one hidden layer with a variable number of neurons (4,8,16,32), and one output layer of 6 neurons for HAR (one for each activity). The model is initially trained in Keras and compressed using tflite framework to perform quantization-aware training [24], this converts the pre-trained 32-bit floating point (FP) weights to 8-bit integer (INT) values. In that regard, it has been proven that INT8 provides a superior efficiency than FP8 for DNN inference [7]. Quantization-aware training reduces the memory footprint of the model by up to 90%, depending on the final NN architecture. From there, a uniform quantization is done to further obtain the weights and biases fitting the proposed architecture (4-8 bits).

C. Model selection

Fig. 2 shows the obtained classification accuracy values for both HAR and PAMAP benchmarks for various configurations of the hidden MLP layer (varying number of bits in the weights and biases for 4 different neuron configurations). First, we target to attain 90% accuracy on both benchmarks, which requires at least an 8-bit quantization of weights and biases. Then, we evaluate the minimum number of neurons in the hidden layer, common to both benchmarks, to find the best compromise between accuracy and computation efficiency around this baseline. For HAR, the best accuracy is for 16 hidden neurons with 94.3% (32 neurons have a 92.9% accuracy). Using 8 neurons, the accuracy drop is only 0.4%, which we consider acceptable considering the substantial gain in terms of computations in the hidden layer (50%). Reducing to 4 neurons drops the accuracy to 63% which we considered too low for the application. For PAMAP, the maximum accuracy in our design space was 90.5% with 32 hidden neurons. This accuracy reduces to 88.3% with 8 neurons, which we consider again acceptable considering 75% less computational load for this layer. Reducing to 4 neurons significantly degrades the accuracy to 42%. Fig. 3 shows how the two benchmarks perform with varying number of bits in the hidden layer's accumulator. The PAMAP dataset has 243 variables but it still requires a high accumulator precision, similar to the HAR which had 561 inputs, to prevent a significant accuracy loss. Hence, the chosen configuration for both datasets is to use 8-bit weights and biases, 8 neurons in the hidden layer, and 18-bit resolution for the accumulator. It should be noted that the proposed hardware is fitted to this application but could be made more generic by integrating spare hardware neurons.

III. Background on Time-Domain Accelerators

Time-domain accelerators represent and compute data in the time domain. A time-based MAC operation can be encoded in the timing properties of a pulse, such as delay, phase, or frequency. As such, time-domain computing allows the representation of multi-bit data in a single wire, which can reduce the dynamic power dissipation. Fig. 4 depicts a baseline time-based architecture, based on a digital-to-time converter (DTC). The DTC converts multi-bit digital data (inputs X and weights W) into a single pulse, acting as an interface between the digital and time domain. To perform a MAC operation, the

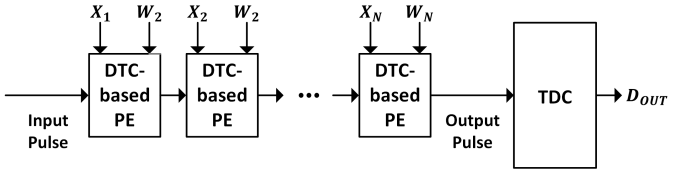


Fig. 4: DTC-Based MAC

DTC is followed by an accumulation phase, governed by the output pulse. The signal is converted back to digital using a Time-to-Digital Converter (TDC).

A. Baseline time-based computing

In basic time-based computing, multi-bit data can be presented by pulse-width-modulation (PWM), such that the pulse width represents the magnitude of the data [15], [16], [25]. Hence, DTC-based Processing Elements (PEs) are used to output PWM pulses with a duration proportional to the inputs or the inputs-weights product (X, W). Thus, a single PE can perform time-domain multiplication, while cascading multiple PEs in a chain can carry out addition (accumulation) as the input pulse propagates through the chain. This process is expressed as:

$$\begin{aligned} T_{out} &= T_{in} + (X_1W_1 + X_2W_2 + \cdots + X_NW_N)\Delta\tau \\ &= T_{in} + (\sum_{i=1}^{N} X_iW_i)\Delta\tau \end{aligned} \tag{2}$$

where T_{in} is the duration of the input pulse, and $\Delta\tau$ is the DTC resolution. The PE can be realized in different manners.

a) Single-bit PE multiplication: Fig. 5(a) illustrates a first DTC-based PE implementation [15]. It performs a single-bit multiplication using a voltage generator and a starved inverter with cascaded NMOS devices. The voltage generator produces 4 distinct voltage levels V_0, V_1, V_2, and V_3, each increasing the duration of the input pulse (T_{in}) by $1\Delta\tau$, $2\Delta\tau$, $3\Delta\tau$, or $4\Delta\tau$, respectively. One voltage level is selected based on the digital input (X_i) using a decoder. The delay path is only activated when the binary weight $W_i = '1'$, otherwise the DTC is disabled by skipping the delay path. More delay levels can be generated by producing more distinct voltage levels or by cascading more PWM stages with binary weighted loads.

b) Multi-bit PE multiplication: To avoid using a voltage generator and allowing multi-bit multiplication, Fig. 5(b) shows a PE using ladder inverters [16], [26]. It relies on the sequential charging of v_{out0} to v_{out3} when v_{in} decreases. v_{out0} directly transits from low-to-high via the PMOS device P_0. While v_{out1} transits via PMOS devices P_0 and P_1, thus it can only transit after the transition of v_{out0}. Similarly, v_{out2} only transits after v_{out1}, and v_{out3} transits after v_{out2}. Following the ladder inverters, a multiplexer is used to map the input-weight product to one of the outputs, and thereby set T_{out} with respect to the rising edge of v_{in}.

c) Challenges of the basic architecture: The architecture in Fig. 4 dedicates separate DTC-based PE for each input/weight pair (spatially unrolled). Hence, it relies on the ability of the DTC to output accurate absolute delays proportional to the input or the input-weight product. Utilizing this architecture for precision applications that require generating many delay levels (e.g. 255 levels for an 8-bit product) can be tedious because it necessitates using complex delay calibration and tuning and/or using larger devices to overcome mismatches between the PEs, reducing the overall area and power efficiencies. In a trial to overcome this limitation, more advanced architectures have been presented [11], [12].

B. Advanced time-based computing

a) Single DTC with gated ring oscillators: the area efficiency can be improved by employing a single DTC, with the accumulation performed in the phase domain using a gated ring oscillator (GRO) [11], as Fig. 5(c) shows, when the oscillator is enabled by the DTC output (DTC_{OUT}), its phase ϕ advances by $\phi[n] = \phi[n-1] + \frac{2\pi}{10}X_iW_i$ for 5-stage oscillator, otherwise it holds its phase information. The pulse width of DTC_{OUT} is proportional to the digital input X_i, and the oscillation frequency is linearly controlled by the weight W_i. To realize continuous accumulation, a counter detects when the GRO phase returns to 0. The readout logic samples the GRO phase and the counter output to generate the MAC operation result. Fig. 5(d) shows how to realize a signed accumulation by utilizing bi-directional GRO. The output of the DTC is provided to the forward-direction GRO when the sign is positive and to the reverse-direction GRO when the sign is negative. This allows the phase to increment/ resp. decrement when the sign is positive/ resp. negative. In this case, an up/down counter detects when the oscillator returns to its initial state, such that it increments its value in the forward direction and decrements its value in the reverse direction. This architecture is very efficient (see Table II), but non-idealities must be carefully considered, specifically when scaling up. For instance, leakage and charge-injection errors [27] can degrade the oscillator phase information. Also, linearly controlling the oscillator frequency with high bit-width can be challenging.

b) Single DTC with gated delay lines: to provide a fully digital implementation, GROs can be replaced with a bi-directional gated delay line (GDL) [12]. Fig. 5(e) shows that the GDL is enabled by the DTC output, which allows a signal to propagate in the GDL (forward or reverse direction); otherwise, the DL state is preserved by the memory (latch) mode. The state of the delay line increases/decreases when the signal propagates in the forward/backward direction by the amount of the pulse width [28]. Fig. 5(f) shows a timing diagram example. When the sign is positive, the GDL state increases ('1' propagates in the forward direction). For $X_1 = 4$, the state (thermometer code) increases by the amount of the pulse width $DTC_{OUT} = 4\tau_d$: "0000000000" (initial state) to "1111000000". When the sign is negative, the GDL decreases ('0' propagates in the reverse direction). When the signal reaches the edge of the delay line, a full-scale signal is asserted to allow the signal's complement to propagate in a loop configuration through an inverter. An up/down counter is used to detect the full-scale condition. This configuration enables long-duration time accumulation. In this architecture, no delay control is implemented. This

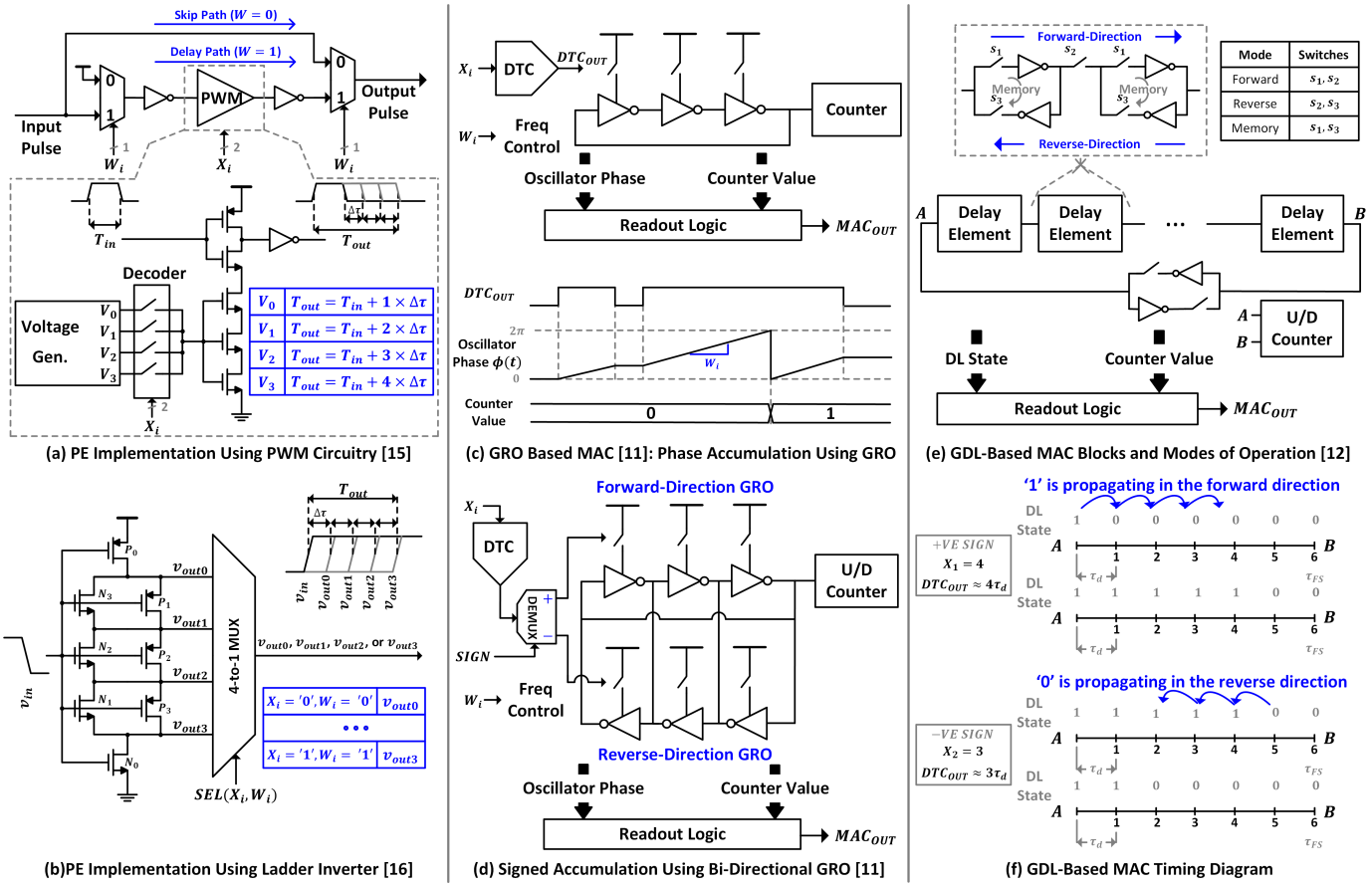


Fig. 5: State-of-Art Time-Based Architectures

means that multi-bit weights are realized either by utilizing a single delay line with configurable length sequentially or by utilizing multiple delay lines for each weight bit to allow parallel operation, which could impact the area efficiency.

C. Summary of the current challenges

Analog-based implementations are generally achieving competitive results for low precision of inputs, weights and outputs (I/W/O), for instance, 1-8/1/8 precision with calibration in [15] or 1/3/4 with calibration in [16]. However, to the best of our understanding, these analog-based implementations follow a similar trend as described in [6]: when the precision increases, the efficiency falls steeply with an increasing implementation cost due to analog non-idealities (matching, leakage, noise, ...). Thus, these implementations are typically targeting low-precision applications. Using analog blocks such as voltage generators or GROs increases the systems' susceptibility to leakage, noise, and variations, specifically considering modern CMOS technologies and high-precision applications. This may induce computation errors and/or highly increase the system complexity. Replacing GROs with delay lines allows for fully digital implementation, but maintains a relatively high complexity to build multiple proportional delay lines. Hence, this work targets a more elegant and efficient all-digital implementation for in-sensor processing applications.

IV. Proposed Architecture

Fig. 6 depicts the proposed computation layer for HAR application (which could be easily adapted to other applications integrating more neurons). In this section, we present the original time-based MAC array based on our previous work [29] and its extension towards a full system:

- The precision of the TAC is upgraded to support up to 18-bit signed accumulation with 8-bit signed inputs and 8-bit weights (as decided in Section II).
- Various post-accumulation functionalities integrated into the neurons including RELU activation, bias addition, and quantization.
- The implementation of a SRAM memory, to store all necessary DNN weights and biases.

A. Time-domain MAC circuit

The proposed time-domain MAC architecture [29] is shown in Fig. 7. The topology has been chosen to enable a simple and fully digital implementation, which can easily be scaled according to the needs of in-sensor processing applications. It is composed of a digital-to-time converter (DTC) which converts 8-bit digital input X_i into a PWM signal DTC_{OUT}, and a time accumulator (TAC) which uses 8-bit weights W_i and a sign bit $SIGN_i = 1('1'), -1('0')$ to perform signed accumulation governed by the DTC. An 18-bit accumulation

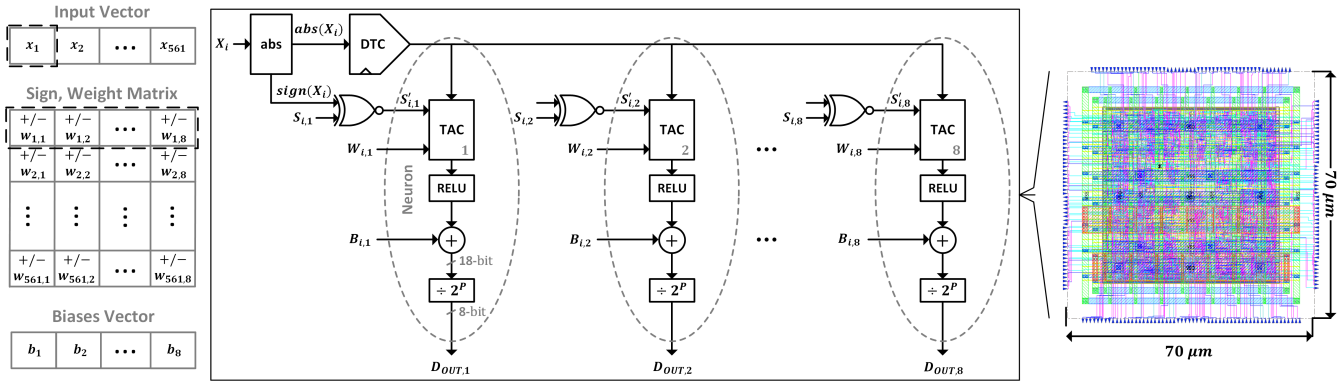


Fig. 6: Proposed 8-Neuron Computation Layer

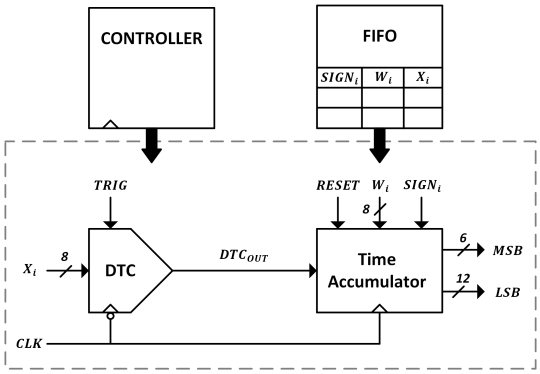
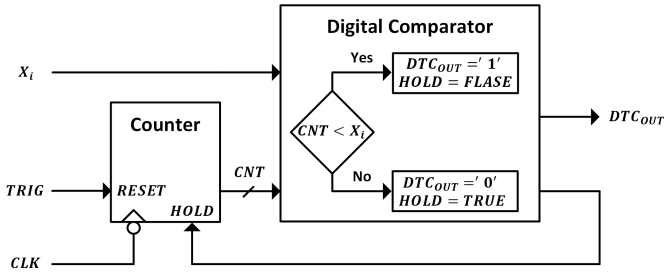
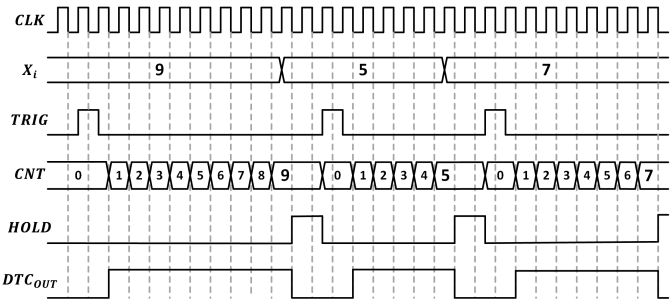


Fig. 7: Proposed Time-Based Architecture



(a) DTC Block Diagram



(b) DTC Timing Diagram

Fig. 8: Proposed Digital-to-Time Converter DTC

result D_{OUT} (6 most-significant-bits MSB, and 12 least-significant-bits LSB) is produced by the TAC state machine. The data required by the DTC and the TAC (X_i, W_i, and $SIGN_i$) are provided by a FIFO (memory macro), while the required control signals ($TRIG$, and $RESET$) are provided by a controller (sequencer).

1) Digital to Time Converter (DTC): Fig. 8 shows the implementation of the proposed DTC and its timing diagram. As shown in Fig. 8a, the DTC is composed of a falling-edge-triggered counter and a digital comparator. A DTC operation is triggered on need by the $TRIG$ signal, which resets and enables a counter.

The digital comparator continuously compares the counter value CNT and the input value X_i. As long $CNT < X_i$, the comparator outputs '1' ($DTC_{OUT} = '1'$). Otherwise, it outputs '0' ($DTC_{OUT} = '0'$) and activates the $HOLD$ signal to freeze the counter state and then wait for the trigger of a new operation.

The timing diagram of the DTC when a sequence of inputs equal 9, 5, and 7 is applied (these numbers are chosen for illustrative purposes), is shown in Fig. 8b. The DTC uses the clock period T_{CLK} as its time reference i.e., the pulse width (duration) of DTC_{OUT} equal to $X_i \times T_{CLK}$.

2) Time Accumulator (TAC): The proposed TAC is implemented by the state machine shown in Fig. 9a. The state machine is triggered by the clock rising edge and it advances when the DTC_{OUT} is high. It can advance bi-directionally based on the sign of the accumulation $SIGN_i$, while the step size of each advancement is determined by the weight W_i.

If the accumulation contains more states than what is encoded in the state machine $LSB[11:0]$, the number of times the state machine returns to its initial state in both directions is tracked by $MSB[5:0]$ bits allowing long-time accumulation as needed (see Fig. 9a). A return while the state machine is advancing in the positive direction (+ve sign) leads to an increment ($MSB+=1$), and a return while the state machine is advancing in the negative direction (-ve sign) leads to a decrement ($MSB-=1$). Both the MSB bits and the current state $LSB[11:0]$ represent the result of the MAC operation. For example, the operations described in (3) are executed as shown in Fig. 9b. Firstly, the state machine advances from its initial state (0) in the positive direction (its value increments) by a step of 6 as defined by W_1. The advancement continues for 9 clock cycles as defined by X_1 (DTC_{OUT} pulse duration). This operation results in an output equal to 54.

Following the first operation, the second operation ($X_2 = 5, W_2 = 15$) is triggered with a negative sign. Therefore, the state machine advances in the negative direction (its value decrements) starting from the last state ($MSB = 0, LSB = 54$). As shown in Fig. 9b, the state machine returns to its initial state while it advances indicating that the output of this operation is negative. This return is tracked by decrementing the MSB ($MSB = -1$). In this case, the output of the operation is -21. The last operation ($X_3 = 7, W_3 = 12$) in Fig. 9b is triggered with a positive sign. The resultant advancement in the positive direction returns the state machine to its initial state leading to a positive output equal to 63. This value is reserved until a new operation is triggered.

Fig. 9c shows the timing diagram of the MAC operation in (3) obtained by the simulation results utilizing both the proposed DTC and TAC. Note that having the DTC operating at the falling clock edge and the TAC operating at the rising clock edge, makes the architecture immune to time domain non-idealities like jitter and skew.

$$X.(SIGN)W = \begin{bmatrix} 9 \\ 5 \\ 7 \end{bmatrix} . \begin{bmatrix} (+)06 \\ (-)15 \\ (+)12 \end{bmatrix} \\ = (54 - 75) + 84 = -21 + 84 = 63 \tag{3}$$

B. Determination of the operation rate

In the proposed architecture, the necessary time T to perform $2N$ operations (multiply and accumulate) depends on the clock frequency $1/T_{CLK}$ and the absolute values of the inputs $abs(X_i)$. T can be determined as:

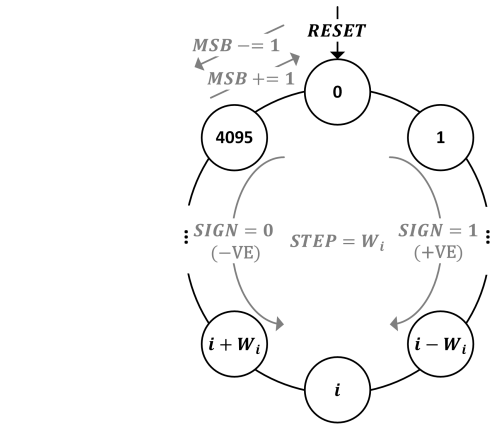
$$T = \sum_{i=1}^{N} abs(X_i)T_{CLK} + 2T_{CLK} \simeq \\ (X_{avg}T_{CLK} + 2T_{CLK}) \times N \tag{4}$$

Here, $X_{avg} \times T_{CLK}$ represents the average time required by the DTC (per MAC operation) while $+2T_{CLK}$ represents the controller overhead.

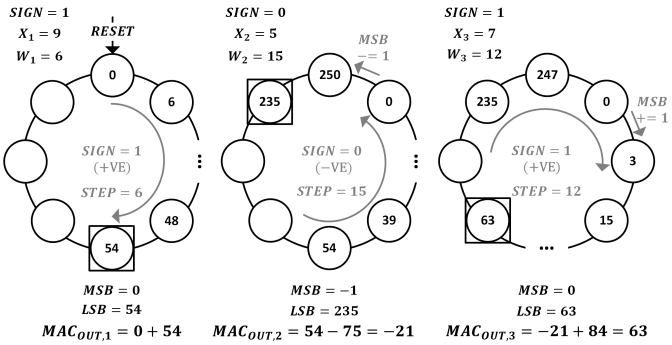
Similarly, the operation rate R is a function of the input X_i can be determined as follows:

$$R = \frac{2N}{T} \simeq \frac{2}{(X_{avg}T_{CLK} + 2T_{CLK})} \tag{5}$$

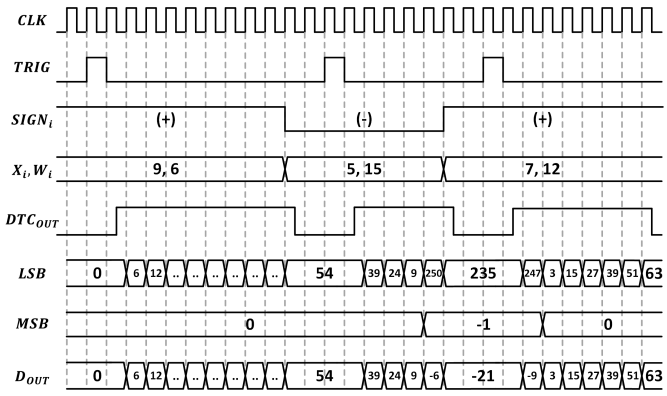
Fig. 10 shows the operation rate R in GOPS as a function of the average input X_{avg} at different clock frequencies. A larger operation rate is obtained for lower input values and higher clock frequency. Also, employing more computation cores for the same number of operations increases the operation rate further.



(a) TAC State Machine



(b) Accumulation Example



(c) MAC Operation Timing Diagram

Fig. 9: Proposed Time Accumulator TAC

C. Post-accumulation functionalities

As shown in Fig. 6, the input vector $[x_1, x_2, ...x_{561}]$ is applied sequentially (one-by-one) to the 8-neuron computation layer (the hidden layer in a 2-layer neural network). For each signed input X_i, a vector of unsigned weights $[w_{i,1}, w_{i,2}, ...w_{i,8}]$ and a vector of sign bits $[s_{i,1}, s_{i,2}, ...s_{i,8}]$ is applied to neuron 1 to neuron 8, respectively. Here, the sign of the accumulation is determined by an XNOR gate (see Fig. 6). If the sign bit and the extracted sign of the input are the same (e.g., both are negative or both are positive), positive accumulation is carried out (the state machine advances in the positive direction). Otherwise, negative accumulation is carried

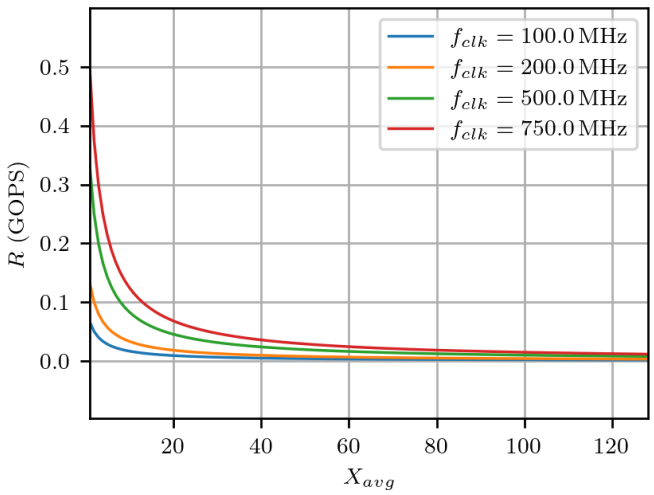


Fig. 10: Operation Rate Vs. Input and Clock Frequency

out (the state machine advances in the negative direction). Similarly, the biases $[b_1, b_2, ...b_8]$ are applied to the neurons in the last operation (zero biases are applied for other operations). As shown in Fig. 6, the 18-bit accumulation result is quantized by level shifting ($\div 2^{P=10}$). Then, the quantized 8-bit outputs of the hidden layer can be applied directly to the output layer which requires 6 neurons for the activities to be classified (walking, walking upstairs, walking downstairs, sitting, standing, and laying).

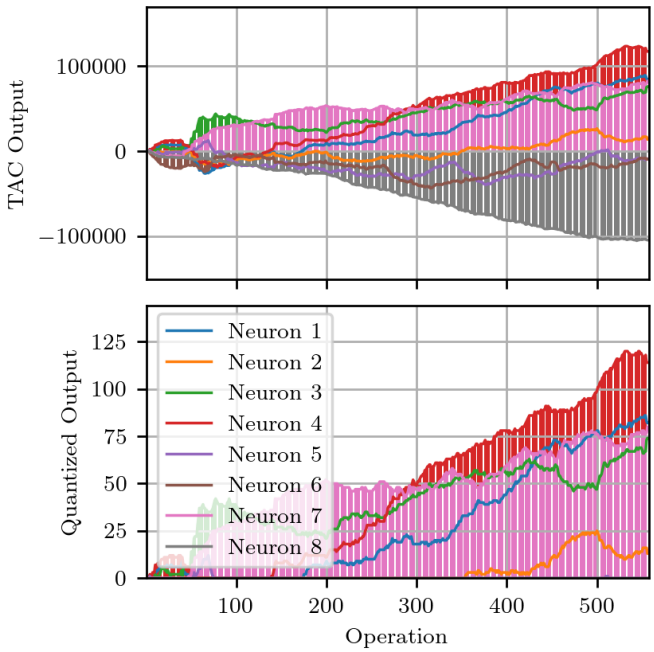
1) FIFO using SRAM array: As described in Fig. 7, the proposed architecture requires a first-in-first-out FIFO or a memory macro to store the application data as needed. An array of 6T SRAM is implemented for this purpose. Using SRAM leverages its unique architectural characteristics for high-speed data processing and low power consumption, e.g. its quick access times, making it an ideal implementation choice for low-power applications that require fast data enqueueing and dequeuing. For our application, the SRAM occupies 180 µm × 97 µm, and it operates at frequencies up to 712MHz.

2) SRAM and input control: The memory and input sequence are controlled by an additional finite state machine (FSM). The FSM starts setting up the initial SRAM address in its idle state. Then, the controller generates the $TRIG$ signal to trigger the DTC and perform a signed accumulation (governed by the DTC output pulse). The controller acknowledges the end of the ongoing accumulation by detecting the falling edge of the DTC pulse. A new operation is then initiated by generating the $TRIG$ signal. This process is shown in Fig. 9c, where the input values are applied sequentially, and the $TRIG$ signal is generated following the falling edge of the DTC output (DTC_{OUT}) initiating a new operation.

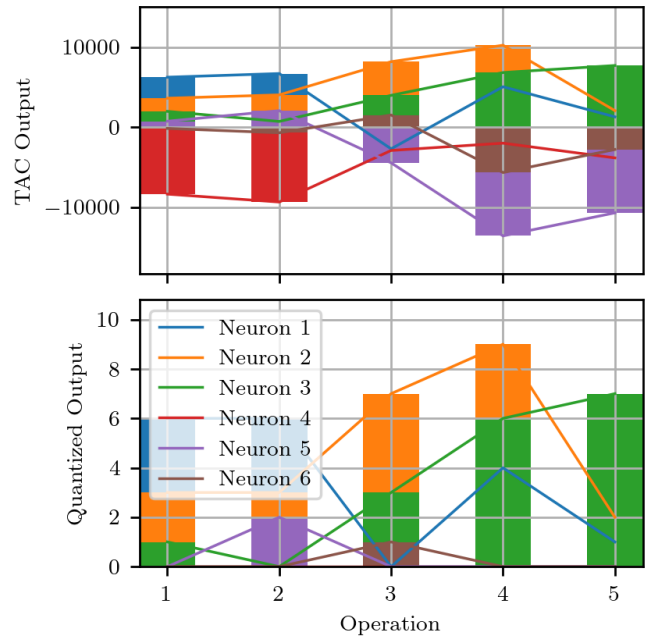
For each operation, the FSM progresses through states to read the 8-bit input data, weights (64-bit, 8-bit for each neuron) and biases, incrementing the SRAM address for each step. The read data is then pushed to the accelerator for processing. Afterwards, the FSM waits for the accelerator to complete the processing, to either loop back for more inputs or to proceed to write processed data back into the SRAM.

These cycles repeat until all data is processed, culminating in a cleanup phase that resets the system, making it ready for the next computation. In this way, the FSM ensures the sequential processing of data from reading, to processing, and writing back in a controlled manner.

V. Simulation Results



(a) Outputs of the Hidden Layer



(b) Outputs of the Output Layer

Fig. 11: Outputs of the Computation Layers

The 8-neuron computation layer only occupies 70 µm × 70 µm, and it dissipates 576 µW at 0.72V supply and 500MHz clock frequency. In this case, the equivalent area of a single neuron is 612.5 µm^2, which is the smallest compared to other time-based architectures. This allows the possibility of

TABLE I: Power Dissipation Versus Clock Frequency

Clock Frequency (MHz)	Power Dissipation (µW)
0.1	22.84
1	22.80
10	28.42
100	118.8
200	220.0
500	576.0
700	832.7

utilizing a large number of neurons even in area-constrained applications, to enable more parallelism and increase the system's throughput.
As the accumulation is time-based, the operation rate of the 8-neuron layer depends on the clock frequency and the value of the sum of the magnitude of the inputs. Higher clock frequency and low values can lead to higher operation rates (see Fig. 10).

A. Performances of the proposed architecture

The computation layer is characterized at 500MHz and around the average value ∼64 of the input (magnitude) range (0 to 128). Fig. 11 shows the RTL behavioural simulation of the hidden and output layers to verify the quantitive outputs of the TAC (signed 18-bit) and the final quantized outputs (signed 8-bit). The time taken for the hidden layer to output the quantized final results in this test scenario is ∼105 µs (52813 clock cycles). While for the output layer, ∼746 ns (373 clock cycles) is required. The number of required clock cycles and operation rate given are extracted from the timing diagrams, and they can be determined from the formulas given in (4)-(5). The highest rate is recorded for binary inputs with a unity average. Note that the operation rate dependency on the clock frequency allows deploying dynamic frequency and voltage scaling at low computation demands for power savings. Table I shows the average power dissipation (tool extracted) at 0.72V supply versus the clock frequency. We have included an extraction at 100 kHz operating frequency to match the settings used in [30] and provide a fair power consumption estimation. Note that below 10MHz, the leakage dominates the power dissipation.
Since the performance is dominated by the hidden layer due to the large number of required operations, a single 8-neuron layer can be utilized for both the hidden and output layers in a temporal sequential manner. Only 5 operations are performed by the output layer with 6 neurons (2 neurons will be idle).

Given this configuration, the computation layer achieves an average operation rate of 0.12 GOPS (up to 2.67 GOPS), an average energy efficiency of 0.21 TOPS/W (up to 4.63 TOPS/W), and area efficiency of 24.74 GOPS/mm^2 (up to 544.22 GOPS/mm^2) Its normalized energy efficiency [19] is 13.46 1b-TOPS/W (up to 296.30 1b-TOPS/W). While the normalized and scaled area efficiency is 1583.36 1b-GOPS/mm^2 (up to 18963.2 1b-GOPS/mm^2).
If the accumulation number of bits is considered in this metric, the normalized efficiency will increase 18 times. To the best of our knowledge, the proposed circuitry achieves one of the highest recorded accumulation ranges.
Moreover, thanks to the simplicity of the design, it can be easily tailored to offer precision scalability as per the application needs thereby saving area and power. Additionally, other configurations can be easily implemented to target various applications by integrating more neurons, or by utilizing more computation layers.

B. Comparison with existing time-based architectures

A comparison with the state-of-art is conducted in Table II. Based on the comparison, the key features of the proposed architecture can be summarized as follows:

1) The architecture has no dependency on analog and time domain non-idealities, and it has an all-digital implementation allowing more reconfigurability and easier integration into other digital circuits.
2) The proposed architecture achieves good precision by supporting multi-bit inputs (8-bit), weights (8-bit), and outputs (18-bit). Also, the precision can be tailored based on the application requirements.
3) The 8-neuron computation layer achieves an average operation rate of 0.12 GOPS, and an average energy efficiency of 0.21 TOPS/W (13.46 1b-TOPS/W).
4) The equivalent area of a neuron is 612.5 µm^2 which considered among the smallest implementations. The 8-neuron computation layer achieves an area efficiency of 24.74 GOPS/mm^2 (1583.36 1b-GOPS/mm^2). Note that, the normalized area efficiency in Table II is scaled with respect to the 22nm node, see [31] for better insights.
5) The architecture utilizes a single clock source, with an operation rate proportional to the clock frequency. Using high clock frequency leads to relatively high power dissipation. Therefore, the proposed architecture fits well with applications that do not require high operation rates like smart IMU sensors.

C. Comparison with other activity recognition implementations

To provide more comparison points with implementations using activity recognition, we evaluated our approach against state-of-the-art microcontroller units (MCU) and custom ASIC implementations targeting HAR. Regarding accuracy, state-of-the-art MCU implementations reported for HAR are 92.3% for UCI HAR and 93.1% for PAMAP [22]. ASIC designs improved this accuracy using additional sensors and processing, achieving 95% on HAR [30]. Our implementation achieves 93.9% for UCI HAR and 88.3% for PAMAP, with a standard data processing, which is competitive with state-of-the-art accuracies. Regarding power, the DNN classifier used in the digital ASIC implementation in [30] achieved 10 µW power consumption at a 100kHz frequency and 1V power supply. In comparison, Table I reports the estimated power dissipation for various clock frequencies, showing that under 1MHz frequency, leakage dominates the overall power dissipation, at 22.8 µW. This difference is due to the technology node (65nm versus 22nm). Yet, the proposed circuit can achieve significantly higher frequency operation (700MHz), opening the possibility of heavy duty-cycling to reduce the general power consumption.

TABLE II: Comparison of State-of-Art time-based MAC Architectures

	JSSC'19 [11]	JSSC'20 [12]	ISSCC'19 [13]	JSSC'22 [14]	OJCAS'23 [16]	This work
Process	28 nm	40 nm	65 nm	65 nm	65 nm	22 nm
All-Digital Imp.	NO	YES	NO	NO	No	**Yes**
Domain	Phase	Time	Time	Hybrid	Time	**Time**
Clock frequency (MHz)	-	25	1×10^{-3} - 1.5	2.12-90	-	500
Power Dissipation (μW)	60.73	30.17	0.3 - 3.4	126.72	697	576
Supply Voltage (V)	0.7	0.537	0.4 - 1	0.7 - 1.1	1.2	0.72
Precision (I/W/O)	8/8/8	4/1/8	3-8/3-8/-	4,7/4,7/16	1/3/4	**8/8/18**
Area per MAC (μm^2)	960	124×10^3	200×10^4	2000	∼5025	**612.5**
Operation Rate (GOPS)	0.78	0.365	2.73×10^{-3}	5.98	4.03 [a]	0.12
Energy Effic. (TOPS/W)	12.4	12.08	9.1	47.19	116	0.21
Norm. Effic. (1b-TOPS/W)	793.6	48.32	81.9	755.04	208.8	13.46
Area Effic. (GOPS/mm^2)	1300	2.94	1.36×10^{-3}	21.81	20	**24.74**
Norm. Area Effic. (1b-GOPS/mm^2) [b]	125243.11	32.42	0.54	2191.42	376.79	**1583.36**

[a] Normalized to 1 neuron throughput

[b] Normalized 1-b Area Efficiency = Area Efficiency (TOPS/mm^2) × No. of input bits × No. of weight bits $\times(1+80\%)^{\log_2 \frac{node^2}{(22nm)^2}}$ assuming 80% area improvement per technology node

VI. Conclusion

This paper presents a time-domain neural network architecture that targets in-sensor processing applications. The proposed 8-neuron computation layer computes sequential inputs and supports signed accumulation up to 18 bits. A single clock is required allowing dynamic frequency and voltage scaling for configurable performance demands and power savings. Moreover, the architecture's small sizes and low complexity allow utilizing a large number of neurons even in small chips to enable more parallelism and increase the throughput. The architecture has an all-digital implementation that benefits from technology scaling and it has no dependency on analog non-idealities allowing easy integration into other on-chip digital circuits. These features make the proposed accelerator suitable for precision sensing applications, adding advanced capabilities with low cost to the next-generation smart sensors.

References

[1] S. García-De-Villa, D. Casillas-Pérez, A. Jiménez-Martín, and J. J. García-Domínguez, "Inertial sensors for human motion analysis: A comprehensive review," *IEEE Transactions on Instrumentation and Measurement*, pp. 1–39, 2023.

[2] Z. Zhongkai, S. Kobayashi, K. Kondo, and T. H. M. Koshino, "A comparative study: Toward an effective convolutional neural network architecture for sensor-based human activity recognition," *IEEE Access*, vol. 10, p. 20547–20558, 2022.

[3] M. Rana and V. Mittal, "Wearable sensors for real-time kinematics analysis in sports: A review," *IEEE Sensors Journal*, vol. 21, no. 2, p. 1187–1207, 2021.

[4] F. Daghero, A. Burrello, C. Xie, M. Castellano, L. Gandolfi, A. Calimera, E. MACII, M. Poncino, and D. J. Pagliari, "Human activity recognition on microcontrollers with quantized and adaptive deep neural networks," *ACM Transactions on Embedded Computing Systems*, vol. 21, no. 4, pp. 1–28, 2022.

[5] J.-s. Seo, J. Saikia, J. Meng, W. He, H.-s. Suh, Anupreetham, Y. Liao, A. Hasssan, and I. Yeo, "Digital versus analog artificial intelligence accelerators: Advances, trends, and emerging designs," *IEEE Solid-State Circuits Magazine*, vol. 14, no. 3, pp. 65–79, 2022.

[6] B. Murmann, "Mixed-signal computing for deep neural network inference," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 1, pp. 3–13, 2021.

[7] M. van Baalen, A. Kuzmin, S. S. Nair, Y. Ren, E. Mahurin, C. Patel, S. Subramanian, S. Lee, M. Nagel, J. Soriaga *et al.*, "Fp8 versus int8 for efficient deep learning inference," *arXiv preprint arXiv:2303.17951*, 2023.

[8] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology," *Neural Computation*, vol. 10, no. 7, pp. 1601–1638, 1998.

[9] H. A. Maharmeh, N. J. Sarhan, C.-C. Hung, and M. I. M. Alhawari, "A comparative analysis of time-domain and digital-domain hardware accelerators for neural networks," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021.

[10] P. S. Locatelli, D. M. Colombo, and K. El-Sankary, "Time-domain multiply-accumulate unit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 6, pp. 762–775, 2023.

[11] Y. Toyama, K. Yoshioka, K. Ban, S. Maya, and A. S. K. Onizuka, "An 8 bit 12.4 tops/w phase-domain mac circuit for energy-constrained deep learning accelerators," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 10, p. 2730–2742, 2019.

[12] A. Sayal, S. S. T. Nibhanupudi, and S. F. J. P. Kulkarni, "A 12.08-tops/w all-digital time-domain cnn engine using bi-directional memory delay lines for energy efficient edge computing," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 1, p. 60–75, 2020.

[13] N. Cao and M. C. A. Raychowdhury, "14.1 a 65nm 1.1-to-9.1tops/w hybrid-digital-mixed-signal computing platform for accelerating model-based and model-free swarm robotics," in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2019.

[14] S. Gweon, S. Kang, and K. K. H.-J. Yoo, "Flashmac: A time-frequency hybrid mac architecture with variable latency-aware scheduling for tinyml systems," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 10, p. 2944–2956, 2022.

[15] J. Yang, Y. Kong, Z. Zhang, Z. Liu, J. Zhou, Y. Wang, Y. Liu, C. Guo, T. Hu, C. Li, L. Liu, J. Zhang, S. Wei, J. Yang, and S. Yin, "Timaq: A time-domain computing-in-memory-based processor using predictable decomposed convolution for arbitrary quantized dnns," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 10, pp. 3021–3038, 2021.

[16] H. A. Maharmeh, N. J. Sarhan, and M. I. M. Alhawari, "A 116 tops/w spatially unrolled time-domain accelerator utilizing laddered-inverter dtc for energy-efficient edge computing in 65 nm," *IEEE Open Journal of Circuits and Systems*, vol. 4, pp. 308–323, 2023.

[17] P. Houshmand, J. Sun, and M. Verhelst, "Benchmarking and modeling of analog and digital sram in-memory computing architectures," 2023. [Online]. Available: https://arxiv.org/abs/2305.18335

[18] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz *et al.*, "A public domain dataset for human activity recognition using smartphones." in *Esann*, vol. 3, 2013, p. 3.

[19] N. R. Shanbhag and S. K. Roy, "Comprehending in-memory computing trends via proper benchmarking," in *IEEE Custom Integrated Circuits Conference (CICC)*, 2022.

[20] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," *2012 16th International Symposium on Wearable Computers*, pp. 108–109, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:10337279

[21] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[22] F. Daghero, A. Burrello, C. Xie, M. Castellano, L. Gandolfi, A. Calimera, E. Macii, M. Poncino, and D. J. Pagliari, "Human activity recognition on microcontrollers with quantized and adaptive deep neural networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 21, no. 4, pp. 1–28, 2022.

[23] K. Penner, F. Wittenfeld, B. Steinhagen, M. Hesse, and U. Rückert, "Tinyml optimization for activity classification on the resource-constrained body sensor bi-vital," in *2023 IEEE 19th International Conference on Body Sensor Networks (BSN)*. IEEE, 2023, pp. 1–4.

[24] TensorFlow, "Quantization aware training," https://www.tensorflow.org/model_optimization/guide/quantization/training, 2024, accessed: 2024-03-10.

[25] D. Miyashita, S. Kousai, and T. S. J. Deguchi, "A neuromorphic chip optimized for deep learning and cmos technology with time-domain analog and digital mixed-signal processing," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 10, pp. 2679–2689, 2017.

[26] M. Alhawari and N. A. M. H. Perrott, "A 0.5v ¡4uw cmos photoplethysmographic heart-rate sensor ic based on a non-uniform quantizer," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, 2013, pp. 1–3.

[27] M. Z. Straayer and M. H. Perrott, "A multi-path gated ring oscillator tdc with first-order noise shaping," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, p. 1089–1098, 2009.

[28] K. Kim and W. Y. S. Cho, "A 9 bit, 1.12 ps resolution 2.5 b/stage pipelined time-to-digital converter in 65 nm cmos using time-register," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, p. 1007–1016, 2014.

[29] A. M. Mohey, M. Kosunen, J. Ryynänen, and M. Andraud, "Toward all-digital time-domain neural network accelerators for in-sensor processing applications," in *IEEE Nordic Circuits and Systems Conference (NorCAS)*, 2023, pp. 1–6.

[30] G. Bhat, Y. Tuncel, S. An, H. G. Lee, and U. Y. Ogras, "An ultra-low energy human activity recognition accelerator for wearable health applications," *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, oct 2019.

[31] H. Wang, R. Liu, R. Dorrance, D. Dasalukunte, D. Lake, and B. Carlton, "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*, vol. 58, no. 4, pp. 1037–1050, 2023.

Ahmed M. Mohey (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Ain Shams University, Cairo, Egypt, in 2012 and 2018, respectively. Since 2013, he held several industrial and research positions to develop software and electronic products. He is currently pursuing the doctoral degree in Electronics and Nanoengineering at Aalto University, Espoo, Finland. He is interested in analog and mixed-signal integrated circuit design with an emphasis on time-domain signal processing, sensing interfaces, power management, in-sensor/memory computing, and embedded systems.

Jelin Leslin is a doctoral candidate at Aalto University specializing in hardware-aware AI models. He earned his bachelor's degree in Electronics and Communication Engineering from Anna University in 2017 and a master's degree in Electrical Engineering in 2020 through the EIT Digital dual university program, studying at the Technical University of Eindhoven and KTH Royal Institute of Technology. His research focuses on energy-efficient implementations of AI models through model compression, hardware-aware training, and custom hardware design. During his master's, he worked with Volvo Trucks on hardware acceleration for motion control in autonomous vehicles, which was also the focus of his thesis.

Gaurav Singh (Student Member, IEEE) earned his M.Sc. degree in VLSI from Amity University, Noida, India, in 2015. He is currently advancing his doctoral studies at the Department of Electronics and Nanoengineering, Aalto University, Finland. His research focuses on the development of low-power system-on-chip (SoC) solutions for sensor data processing. His work particularly explores microprocessors, low-power SRAM memories, and serial interfaces to enhance communication and debugging. His research work also includes integrating RISC-V processors with Compute In-Memory cores as hardware accelerators, aimed at improving power efficiency in AI applications.

Marko Kosunen (Member, IEEE) received his M.Sc, L.Sc and D.Sc (with honors) degrees from Helsinki University of Technology, Espoo, Finland, in 1998, 2001 and 2006, respectively. He is currently an Associate Professor at Aalto University, Department of Electronics and Nanoengineering. Academic years 2017-2019 he visited Berkeley Wireless Reserarch Center, UC Berkeley, on Marie Sklodowska-Curie grant from European Union. In addition to his academic duties, currently he is one of the three co-chairs of Microelectronics Finland, a academia-industry collaboration organization for microelectronics research and education. He has authored and co-authored more than hundred journal and conference papers and holds several patents. His current research interests include programmatic circuit design methodologies, digital intensive time-based data converters and transceiver circuits, and RISC-V microprocessor implementations with DSP accelerators.

Jussi Ryynänen (Senior Member, IEEE) was born in Ilmajoki, Finland, in 1973. He received the M.Sc. and D.Sc. degrees in electrical engineering from the Helsinki University of Technology, Espoo, Finland, in 1998 and 2004, respectively. He is a full professor and the Dean of school of electrical engineering, Aalto University, Espoo. He has authored or co-authored more than 200 refereed journal and conference papers in analog and RF circuit design. He holds seven patents on RF circuits. His research interests are integrated transceiver circuits for wireless applications. Prof. Ryynänen is currently SG Member for the European Solid-State Circuits Conference (ESSCIRC) and the IEEE Nordic Circuits and Systems Conference (NORCAS). He has served as TPC member of IEEE International Solid-State Circuits Conference (ISSCC), and as a Guest Editor for the IEEE Journal of Solid-State Circuits.

Martin Andraud (Member, IEEE) is an assistant professor in microelectronics at UCLouvain, Belgium, and a visiting professor at Aalto University, Finland. He received his PhD from Grenoble University, France, in 2016. He was a postdoctoral researcher successively with TU Eindhoven in 2016 and KU Leuven from 2017 to 2019. Between 2019 and 2024, he was an assistant professor at Aalto University, Finland. His research interests include the interface between edge AI, hardware/software co-design, testing, and reliability of custom ASIC for various AI accelerators, for instance, mixed-signal Compute-In-Memory architectures and alternative to deep learning models (probabilistic reasoning or hybrid AI).