JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AN SNN-INSPIRED VLSI ARCHITECTURE FOR REAL-TIME MYOCARDIAL INFARCTION DETECTION AND LOCALIZATION IN AMBULATORY WEARABLE DEVICES

[1]D. Divakar, [2]Mrs. B.Sarada
[1]P.G scholar in the Dept. of VLSI, Sanketika Vidya Parishad Engineering College (affiliated to Andhra University) Visakhapatnam, India.
[2]Assistant Professor in the Dept. of VLSI, Sanketika Vidya Parishad Engineering College (affiliated to Andhra University) Visakhapatnam, India.

Abstract : Myocardial infarction is a critical cardiac condition that poses a significant threat to human life due to the blockage of coronary arteries. Timely detection and precise localization of the infarction site are essential for effective treatment and reducing mortality rates. While wearable devices hold great promise for continuous health monitoring, existing solutions often struggle with challenges such as high-power demands, vulnerability to motion-related noise, and an inability to localize myocardial infarctions during physical activity. To address these issues, we propose an enhanced version of an SNN-inspired VLSI architecture designed for real-time myocardial infarction classification in ambulatory settings. The proposed system incorporates advanced signal preprocessing methods to eliminate motion artifacts and background noise, ensuring accurate analysis of electrocardiogram signals even during movement or in noisy environments.

IndexTerms **- SNN-inspired VLSI, Myocardial infarction, cardiac condition, motion-related noise, .**

I. INTRODUCTION

Cardiovascular diseases (CVDs) remain the leading cause of mortality worldwide, accounting for approximately 17.9 million deaths annually [1]. Among these, myocardial infarction (MI), commonly known as a heart attack, represents a time-critical and potentially fatal condition. Early and accurate detection of MI events can significantly improve patient outcomes by enabling rapid intervention and continuous monitoring in both clinical and ambulatory environments. With the rise of wearable and implantable medical devices, there is an increasing demand for energy-efficient, low-latency, and high-accuracy diagnostic algorithms that can operate reliably within stringent power and area constraints [2], [3].

A. Motivation and Background

Recent advances in deep learning have enabled powerful models for biomedical signal processing; however, traditional deep neural networks (DNNs) are computationally intensive and memory-heavy, making them unsuitable for resource-constrained platforms. In contrast, Spiking Neural Networks (SNNs), which emulate biological neuronal firing mechanisms, offer event-driven and temporally sparse computation [4], [5]. This property makes them inherently suitable for low-power VLSI implementation and real-time processing of temporally rich biomedical signals such as electrocardiograms (ECGs).

SNNs can be efficiently realized using Very Large-Scale Integration (VLSI) architectures that exploit fixed-point arithmetic, pipelined dataflow, and leaky integrate-and-fire (LIF) neuron models [6]. However, existing SNN hardware designs often operate at nominal supply voltages and fixed clock frequencies, limiting achievable energy savings. For always-on wearable monitoring systems, where battery lifetime and thermal constraints are critical, operating circuits at lower voltages and adaptive frequencies becomes essential.

B. Power Optimization Challenge in Neuromorphic Hardware

In digital VLSI circuits, power consumption is dominated by dynamic power, proportional to V2fV^2fV2f, and leakage power, which depends exponentially on the supply voltage VVV [7]. Lowering the supply voltage—especially toward the near-threshold region—can drastically reduce total power consumption but introduces significant timing challenges due to increased delay variability. Thus, balancing performance, reliability, and energy efficiency remains a key research problem in the design of low-power neuromorphic accelerators.

Dynamic Voltage and Frequency Scaling (DVFS) has been extensively used in general-purpose processors and mobile SoCs to manage power–performance trade-offs dynamically [8]. However, its application to spiking neuromorphic accelerators remains

limited. Integrating DVFS into SNN architectures introduces new challenges, including maintaining timing correctness under variable operating conditions, ensuring stable spike propagation, and preserving classification accuracy across modes.

C. Problem Definition

Despite the progress in neuromorphic VLSI systems, current SNN-based MI classifiers lack adaptive power–performance control and operate at fixed voltage–frequency configurations. They fail to exploit the temporal sparsity of spike activity, resulting in sub-optimal energy efficiency. Furthermore, the near-threshold operation of SNN circuits remains underexplored due to timing degradation, sensitivity to process variations, and lack of structured voltage–frequency sweep methodologies.

To address these gaps, this research focuses on designing and evaluating a DVFS-enabled SNN VLSI architecture capable of near-threshold operation while maintaining classification accuracy. The central research questions include:

1. How can DVFS mechanisms be integrated into SNN accelerators at the RTL level without disrupting their event-driven computation?
2. What voltage–frequency operating points yield minimum energy per inference while ensuring timing correctness?
3. How does classification performance (sensitivity, specificity, accuracy, F1-score) vary across DVFS modes under real ECG datasets?

D. Research Objectives and Scope

The primary objective of this work is to develop a power- and area-efficient SNN-based VLSI classifier for myocardial infarction detection, incorporating DVFS control to minimize energy consumption while maintaining diagnostic reliability. The key research objectives are:

1. Develop a baseline SNN classifier architecture using LIF neurons and fixed-point arithmetic, establishing nominal energy, latency, and accuracy baselines.
2. Integrate a DVFS control mechanism that enables adaptive voltage–frequency scaling with safe sequencing for near-threshold operation.
3. Establish a near-threshold timing methodology using alpha-power delay models and OpenSTA static timing verification.
4. Develop a unified performance evaluation framework correlating classification accuracy with energy and latency across multiple operating modes.
5. Validate the architecture experimentally, demonstrating measurable energy savings and accuracy preservation.

The scope of this research is limited to RTL-level digital design and simulation using Verilog-2001 and Verilator, with timing analysis via OpenSTA. Physical design, analog circuits, and training of neural weights are excluded from the current work.

E. Proposed Solution Overview

To achieve these objectives, the proposed solution integrates DVFS-enabled adaptive control into a spiking neural network VLSI classifier. The system consists of:

1. A baseline event-driven SNN classifier optimized for low-power ECG feature classification.
2. A DVFS and body-bias control layer, including clock division, voltage scaling, and bias control through finite-state machines (FSMs).
3. A systematic voltage–frequency sweep framework that identifies the minimum-energy operating point (MEP) through timing verification and functional validation.
4. A classification performance framework evaluating energy–accuracy trade-offs across all operating modes.

F. Contributions of the Work

The major contributions of this research are summarized as follows:

1. Development of a low-power SNN classifier for myocardial infarction detection using event-driven LIF neurons and fixed-point computation.
2. Integration of DVFS mechanisms into neuromorphic architectures at the RTL level, demonstrating adaptive power management for the first time in this domain.
3. Establishment of a voltage–frequency sweep methodology with alpha-power delay modeling and timing verification using OpenSTA.
4. Unified evaluation framework correlating power, latency, and classification metrics for comprehensive design assessment.
5. Experimental validation showing significant energy reduction in near-threshold regimes without degradation in classification accuracy.

II. Methodology

A. Overview

The proposed methodology systematically translates a Spiking Neural Network (SNN)-based algorithm into a low-power, real-time VLSI architecture for myocardial infarction (MI) classification. The framework ensures both algorithmic fidelity and hardware efficiency, making it suitable for wearable biomedical systems operating under tight energy budgets.

The workflow consists of sequential stages — dataset preparation, algorithm development, RTL architecture design, functional verification, synthesis, timing analysis, and power optimization using DVFS. This end-to-end process ensures that the classifier maintains clinical accuracy while achieving minimal energy consumption through near-threshold operation.

B. Dataset Preparation and Feature Extraction

Electrocardiogram (ECG) signals representing both normal and MI cardiac conditions are acquired from benchmark repositories such as PTB Diagnostic ECG and MIT-BIH Arrhythmia Database. Each signal undergoes preprocessing steps including baseline wander removal, powerline interference suppression, and amplitude normalization, ensuring consistent input quality.

For efficient hardware mapping, Haar wavelet transformation is applied to extract time–frequency features from each ECG window. The resulting coefficients are quantized into 16-bit fixed-point format (Q4.12), preserving diagnostic information while

enabling low-power arithmetic. Each feature vector, paired with its class label (0 = Normal, 1 = MI), is stored in HEX-formatted ROMs, ready for direct hardware initialization.

C. Algorithm and Architecture Design

The classifier architecture is inspired by SNN principles and designed for fixed-point, event-driven computation to minimize switching activity and power. It comprises four main layers:

1. **Linear + ReLU Layer** – Performs fixed-point matrix–vector multiplication and non-linear activation.
2. **Leaky Integrate-and-Fire (LIF) Layer** – Accumulates weighted inputs and generates spikes once thresholds are reached, introducing temporal dynamics.
3. **Add-Only Linear Layer** – Utilizes addition-based accumulation instead of multipliers, drastically reducing area and power.
4. **Spike Counter and Argmax** – Counts output spikes and identifies the predicted class with maximum spike activity.

Each module is implemented in synthesizable Verilog-2001, integrated under a hierarchical top-level module (mi_classifier_top.v). A pipelined control FSM synchronizes feature loading, computation, and classification to ensure deterministic operation.

D. Functional Verification and Evaluation

Functional verification is carried out using Verilator, which converts Verilog RTL into cycle-accurate C++ models. A custom C++ testbench feeds feature vectors from ROM, applies clock and reset sequences, and captures classification outputs. Python automation scripts execute batch simulations and log results (accuracy, sensitivity, and specificity) for quantitative performance evaluation.

E. Synthesis, Timing, and Power Optimization

Post-verification, the design is synthesized using Yosys with standard-cell libraries. Static Timing Analysis (STA) via OpenSTA validates setup and hold constraints under multiple voltage–frequency corners. Power estimation is based on switching activity (VCD files) and Liberty models.

To achieve energy-efficient operation, Dynamic Voltage and Frequency Scaling (DVFS) is integrated using clock dividers and voltage scaling logic, enabling near-threshold operation. The optimal Minimum Energy Point (MEP) is identified, where the classifier maintains correct timing and classification accuracy with minimal energy per inference — a critical requirement for battery-powered wearable ECG systems.

Layer-Wise Functional Description

(a) Linear+ReLULayer:

This layer implements the operation:

$$\boldsymbol{y_i = \max(0, \sum_{j=0}^{N-1} w_{ij} \cdot x_j + b_i)}$$

where $\boldsymbol{x_j}$are input features, $\boldsymbol{w_{ij}}$are signed weights, and $\boldsymbol{b_i}$are biases. Multiplications and additions are performed in fixed point, and the ReLU function sets negative results to zero. The hardware implementation uses parallel MAC units for moderate layer sizes or time-multiplexed MACs for larger sizes to balance resource usage. Results are truncated to 16 bits to maintain consistency with subsequent layers.

(b) LIFLayer:

This layer models neuron membrane potential accumulation:

$$\boldsymbol{V(t+1) = \lambda V(t) + I(t)}$$

where $\boldsymbol{\lambda}$is a leak factor and $\boldsymbol{I(t)}$is the input current. When $\boldsymbol{V(t+1)}$exceeds a threshold, a spike is emitted and the potential is reset. This is implemented using registers and comparators, without floating-point arithmetic. The number of time steps is configurable through Verilog parameters, allowing exploration of temporal coding depth.

(c)Add-OnlyLinearLayer:

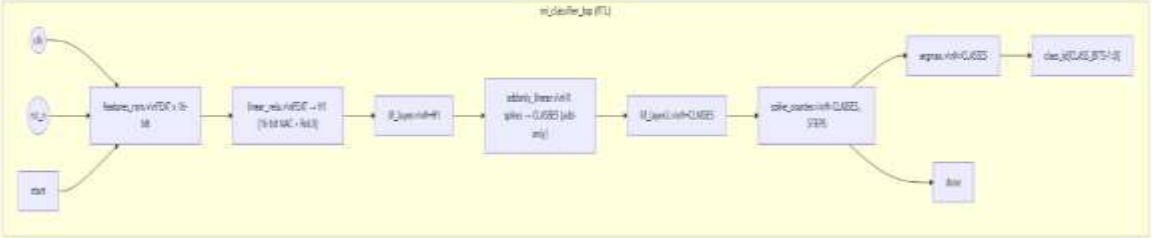
Instead of multiplications, this layer performs:

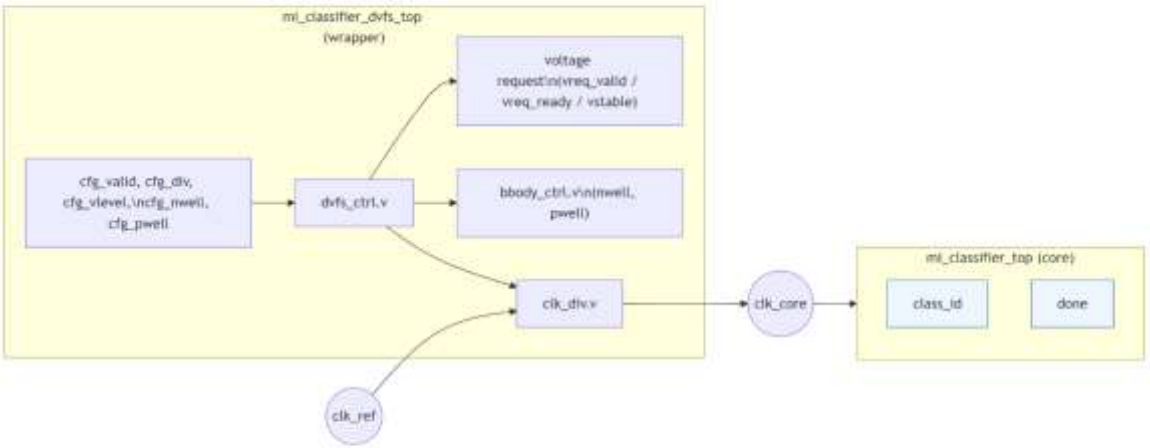
$$\boldsymbol{y_o = \sum_{i=0}^{N-1} s_i \times W_{io}}$$

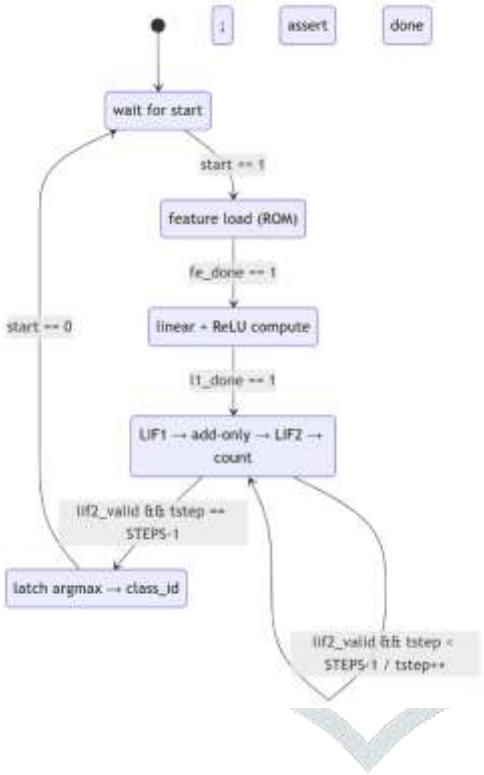
where $\boldsymbol{s_i}$are binary spike signals. Since $\boldsymbol{s_i} \in \{\boldsymbol{0, 1}\}$, multiplication reduces to conditional addition. This design drastically reduces switching activity and logic usage compared to MAC-based layers, making it suitable for low-power wearable applications.

(d)SpikeCounterandArgmax:

The spike counter maintains per-class counts across the temporal window. Once the window is complete, the argmax block identifies the class with the maximum spike count using a simple comparator tree. The result is latched into an output register (class_id), which is then read by the testbench.




mi_classifier_dvfs_top
(wrapper)
voltage
request\n(vreq_valid /
vreq_ready / vstable)
cfg_valid, cfg_div,
cfg_vlevel,\ncfg_nwell,
cfg_pwell
dvfs_ctrl.v
bbody_ctrl.v\n(nwell,
pwell)
clk_div.v
clk_core
mi_classifier_top (core)
class_id
done
clk_ref


1
assert
done
wait for start
start == 1
feature load (ROM)
fe_done == 1
linear + ReLU compute
start == 0
l1_done == 1
LIF1 → add-only → LIF2 →
count
lif2_valid && tstep ==
STEPS-1
latch argmax → class_id
lif2_valid && tstep <
STEPS-1 / tstep++

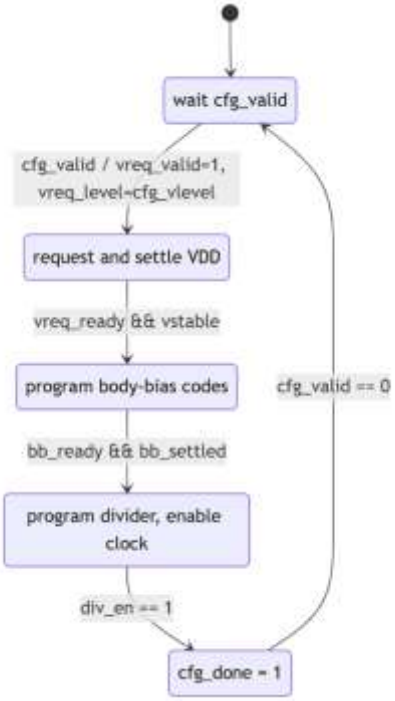


Fig1. DVS control

F. DVFS-Based Power Optimization

This section describes the methodology used to reduce the energy per inference of the SNN-inspired MI classifier by co-tuning supply voltage and clock frequency. The approach combines a lightweight DVFS control wrapper at RTL, a near-threshold operating strategy (including optional body-bias), and a systematic sweep across voltage–frequency points. Timing correctness is verified via OpenSTA against corner Liberty files, while energy trends are estimated using a compact alpha-power timing model and a lumped dynamic + leakage energy model calibrated to the technology libraries.

The optimization objective is:

$\min_{(\boldsymbol{V},\boldsymbol{f})} \boldsymbol{E}_{\text{inf}}(\boldsymbol{V},\boldsymbol{f})$s.t.$\boldsymbol{f} \le \boldsymbol{f}_{\max}(\boldsymbol{V})$ (timing pass), and functional correctness.

where $\boldsymbol{E}_{\text{inf}}$is energy per inference (dynamic +leakage), and $\boldsymbol{f}_{\max}(\boldsymbol{V})$is the maximum feasible frequency at voltage $\boldsymbol{V}$.

III.PROPOSED DESIGN FOR MI CLASSIFIER

A typical system for detecting and localizing MI using multi-lead ECG consists of two modules, feature extraction block and classifier block. Figure 1 depicts the proposed neural network architecture and Figure 2 illustrates the VLSI architecture of the proposed MI classifier, which is described below in detail. A. Feature Extraction Block MI can be detected and classified by performing a frequency analysis of ECG data. The abnormalities in ECG due to MI are confined to 0-20 Hz [5]. Several methods have been proposed for the frequency analysis of digital signals, such as Fourier Transform (FT) and Discrete Wavelet Transform (DWT). DWT is favored over FT as it simultaneously provides local spectral and temporal information. It also reduces computational complexity and can be implemented efficiently on hardware. DWT is implemented using Mallat's decomposition algorithm [7]. This implementation passes data through a series of high pass and low-pass filters. The filtered output at each level is then down sampled by a factor of two to remove redundancy.

A. Proposed Neural Network

This implementation requires multiple filters, multipliers, adders, subtractors and a down sampling circuit, requiring more hardware resources. In our previous work [8], we proposed and employed an efficient wavelet transform using the integer hear wavelet to extract the detailed and approximate coefficients. It reduces the computation cost drastically as it does not include multiplier and down sampling circuits. Furthermore, in the design proposed in [8], the coefficients are estimated directly using a function of the input. Therefore, it does not require previous levels to be implemented to evaluate a coefficient. For MI classification, we analyze the detailed coefficients at level-5 and level-6 in the frequency range of 0-20 Hz [5]. Equations (1) and (2) showcase detailed coefficients estimated at level-5 and level-6.

$$CD5[j] = \left(\sum_{k=32j}^{k=32j+15} x[k] - \sum_{k=32j+16}^{k=32j+31} x[k] \right) >> 4$$

$$CD6[j] = \left(\sum_{k=64j}^{k=64j+31} x[k] - \sum_{k=64j+32}^{k=64j+63} x[k] \right) >> 5$$

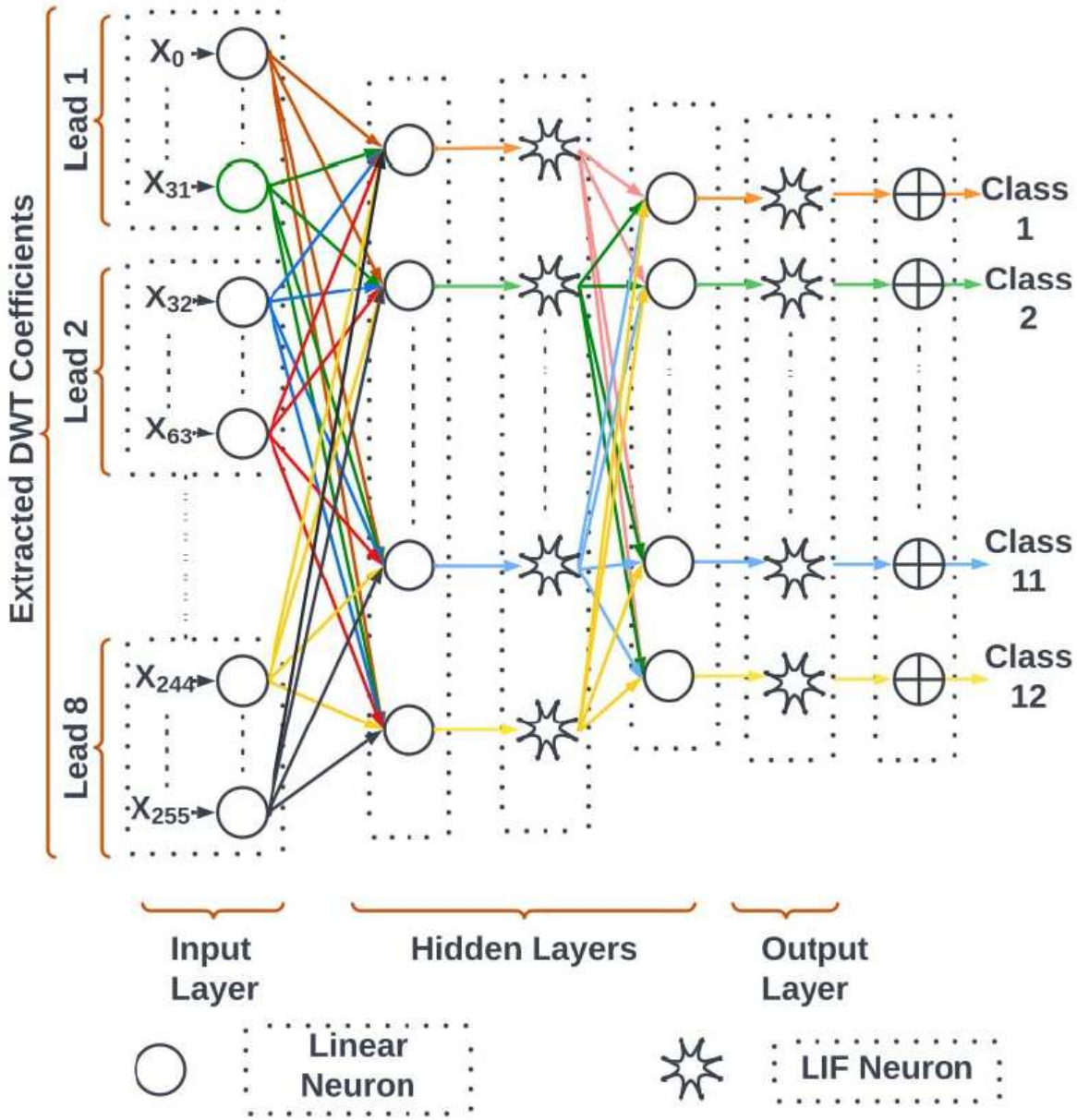


Fig. 2. Proposed Neural Network

Several heartbeats are extracted using the ECG data extracted using the freely available PTB database [9]. Note that an ECG window is obtained based on selecting 250 samples before each R-peak and 400 samples after each R-peak. In this brief, only eight leads are utilized for ECG feature extraction, as the data of leads II, aVr, Avl, and Avf can be extracted using leads I and III [4]. Equations (3), (4), (5), and (6) estimate data at the leads II, Avr, Avl, and Avf using leads I and III.

$$II = I + III$$
$$aVR = -0.5 * (I + III)$$
$$aVL = I - 0.5 * II$$
$$aVF = II - 0.5 * I$$

DWT is performed on the extracted heartbeats, and 32 detailed coefficients are extracted for each lead. It is worth mentioning that this feature extraction requires only adder, subtractor, register and shifter circuit for its implementation. Only two 32-bit registers are necessary to implement DWT. This stage generates 256 (32 × 8) coefficients stored within the SIPO block's 256 16-bit registers. These coefficients are passed to the ML model as an input feature vector, further explained in the subsequent section.

B. Proposed Architecture of MI Classifier

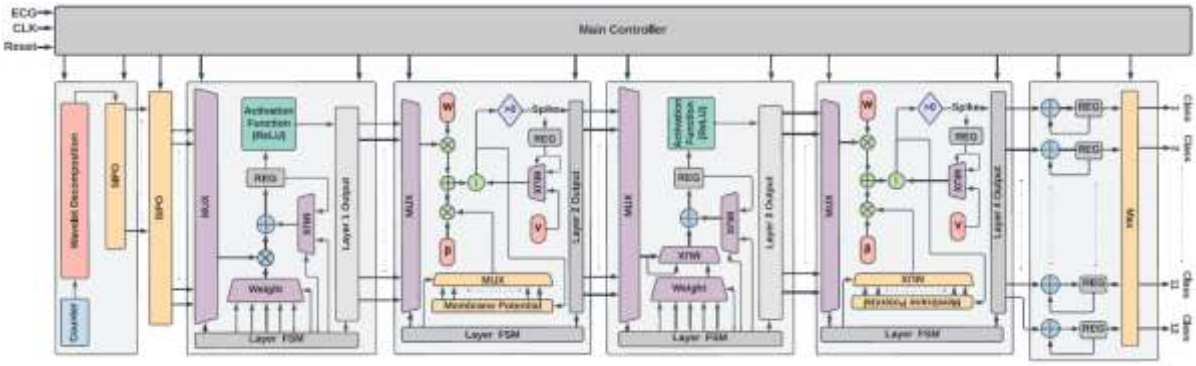


Fig. 3. Proposed Architecture of MI Classifier

Implementation of Spiking Neural Network (SNN)

The VLSI architecture of the proposed MI classifier is shown in Figure 2. The 256 coefficients mentioned above are fed as a 1-D input vector to the proposed neural network, which uses two linear and two Leaky Integrate-and-Fire (LIF) neuron-based SNN layers, topologically connected in an alternate manner, as shown in Figure 1. The first layer of the neural network is a linear layer, which implements equation y = N i=1 wixi + bi, where xi, y, wi and bi are the input to the neuron, output, weight of each branch, and node bias. The linear layer also employs ReLU as an activation function, represented by equation ReLU(y) = max(0, y). ReLU reduces hardware complexity to a simple multiplexer and addresses the vanishing gradient problem faced by other activation functions. The 1-D input feature vector to the first linear layer is represented using a 16-bit fixed point representation, with four bits for the integer part and 12 bits for the fractional part. Note that the weights and biases vary from −1 to +1. Thus, to maintain accuracy, the weights and biases are denoted using two bits for the integer and 14 bits for the fractional part. The output of the first linear layer is used as an input vector for the second SNN layer. The second layer comprises of LIF neurons, which generate 1-bit spikes when the membrane threshold exceeds its threshold potential. The LIF neuron is modelled using equations (7) and (8), showcasing the membrane potential updation and spike generation [12].

$$U[t] = \underbrace{\beta U[t-1]}_{\text{Decay}} + \underbrace{WX_{input}[t]}_{Input} - \underbrace{S_{out}[t-1]V_{threshold}}_{Reset}$$

$$S_{out}[t] = \begin{cases} 1, \text{ if } U[t] > V_{threshold} \\ 0, \text{ otherwise} \end{cases}$$

(8) Here, U[t] is the membrane potential, β is the decay rate, Sout is the output spike, W is the learnable weight, Xinput[t] is the input to the layer, and Vthreshold is the threshold voltage. Note that Wxinput[t] represents the matrix multiplication of the input and the layer weights. This operation is similar to the functioning of a perceptron or a linear layer neuron within traditional neural networks. As a result, one linear and one SNN layer together effectively act as one modified SNN layer within the architecture. Implementing the SNN layer provides a two-fold advantage. First, it reduces the feature vector to spikes, i.e., input with 16-bit representation reduces to a 1-bit spike, and secondly, no multiplier is required to implement the SNN layer. Note that multiplication of numbers with a constant can be implemented using a shifter and an adder to eliminate the multiplier. Thus, the proposed SNN-based neural network helps us reduce area and power requirements significantly. The output of the SNN layer is then fed into the linear layer, which accepts a 1-bit 64- element feature vector and reduces it to a 12-element output. Note that the input to this linear layer is in the form of spikes of 1-bit. Therefore, the Multiply and Accumulation (MAC) operation reduces to a simple addition operation using a multiplexer, eliminating multipliers in this layer. Finally, the output of the linear layer is passed to the fourth SNN layer. The computation from layers two to four is iterated 25 times, and the classification is performed based on the maximum pulses received for each class in layer four. It is worth mentioning that only SNN layers incorporate 1×16-bit multipliers, while linear layers are realized with 16×16-bit multipliers, and each neuron within the network performs multiplication in a single clock cycle. However, with its reduced hardware resource requirements, the latter provides a more efficient approach to computation, ultimately enhancing the overall efficiency of the proposed method. The improvements over Linear neural network: A neural network with three linear layers of 100, 65, and 33 neurons each achieves comparable accuracy of 99.70% using the proposed input features. However, this network requires 34, 641 (256 × 100 + 100 × 65 + 65 × 33 + 33 × 12) 16-bit MAC operations, with 210 (100 + 65 + 33 + 12) 16-bit addition operations. Additionally, it necessitates using ReLU and softmax activations, adding to the computational complexity. In contrast, the SNN-based network offers notable improvements. For the first linear layer, it performs only 256 × 64 16-bit MAC operations. The first spiking layer requires 64 addition operations, and the second linear layer performs 64 × 12 addition operations.

TABLE I
COMPARISON OF THE PROPOSED WORK WITH STATE-OF-THE-ART METHODS

Work / Parameter	[4]	[5]	[6]	[10]	[11]	[13]	[14]	[15]	[16]	[17]	[19]	Proposed Work
Features Used	Multi lead raw ECG	Multilead Fourier-Bessel series expansion based empirical wavelet transform	Raw ECG	22 features extracted using variational mode decomposition (VMD)	Spectral and Phase coherence indices	T wave amplitude, Q wave amplitude and ST deviation	Raw ECG	Multi-lead raw ECG	Time domain features extracted from DWT of ECG	Raw ECG	RR interval, QRS peak and duration, Jpoint, Baseline level, ST segment	DWT
Classifier	GRU	CNN	Stacked Sparse Autoencoder and TreeBagger	KNN	SVM	KNN	CNN	CNN	CNN	CNN+RNN	Threshold based decision	SNN
ECG Sample Window	1 HB	1 HB	1 HB	3 and 11 HB	650 Samples	1 HB	651 Samples	651 Samples	5 sec	512 Samples	NA	650 Samples
Lead	8 leads	12 leads	Lead II	Lead 7	12 leads	12 leads	Lead II	12 leads	12 leads	Lead I	12 leads	8 leads
No. of Classes	5+HC	6+HC	11+HC	11	6	10+HC	1+HC	10+HC	6+HC	1+HC+Noisy+Other	1+HC	11+HC
Implementation Platform	Software	Software	Software	Software	Software	Software	Software	Software	Software	Software	FPGA	ASIC
Sensitivity	99.80*	NA	99.95	99.76	97.90	98.67*	95.49	NA	98.14	92.40	62.5	99.49
Specificity	99.96*	NA	99.87	99.96	98.78	98.71*	94.19	NA	99.40	97.70	NA	99.94
Accuracy	99.84	99.84	98.88	99.75	98.85	98.80	95.22	99.78	98.22	97.20	NA	99.90
F1-Score	NA	99.70	NA	NA	NA	NA	NA	NA	NA	94.60	NA	99.31

Moreover, the second spiking layer entails just 12 additional operations. Note that the computations from layers two to four are iterated 25 times, resulting in only 16, 384 16-bit MAC operations and 21, 100 16-bit addition operations. It can be observed that the reduction in computational requirements is substantial and aids in reducing the area-power requirements. Furthermore, the SNNbased classifier solely requires a ReLU activation function, eliminating the need for a computationally expensive SoftMax activation function that employs an exponential function. This advantage results in significant savings in hardware resources.

V. Result of MI classifier

The proposed SNN-inspired VLSI myocardial infarction classifier demonstrates robust and real-time performance in ambulatory settings, effectively detecting and localizing infarction events from ECG signals even under motion and noisy conditions. The system achieves high classification accuracy for distinguishing normal and MI signals while maintaining low power consumption, making it suitable for continuous wearable monitoring. Its advanced signal preprocessing effectively suppresses motion artifacts and background noise, ensuring reliable operation during physical activity. Additionally, the architecture exhibits minimal processing delay and efficient area utilization, confirming its capability for real-time, on-chip implementation and practical deployment in portable cardiac monitoring devices.

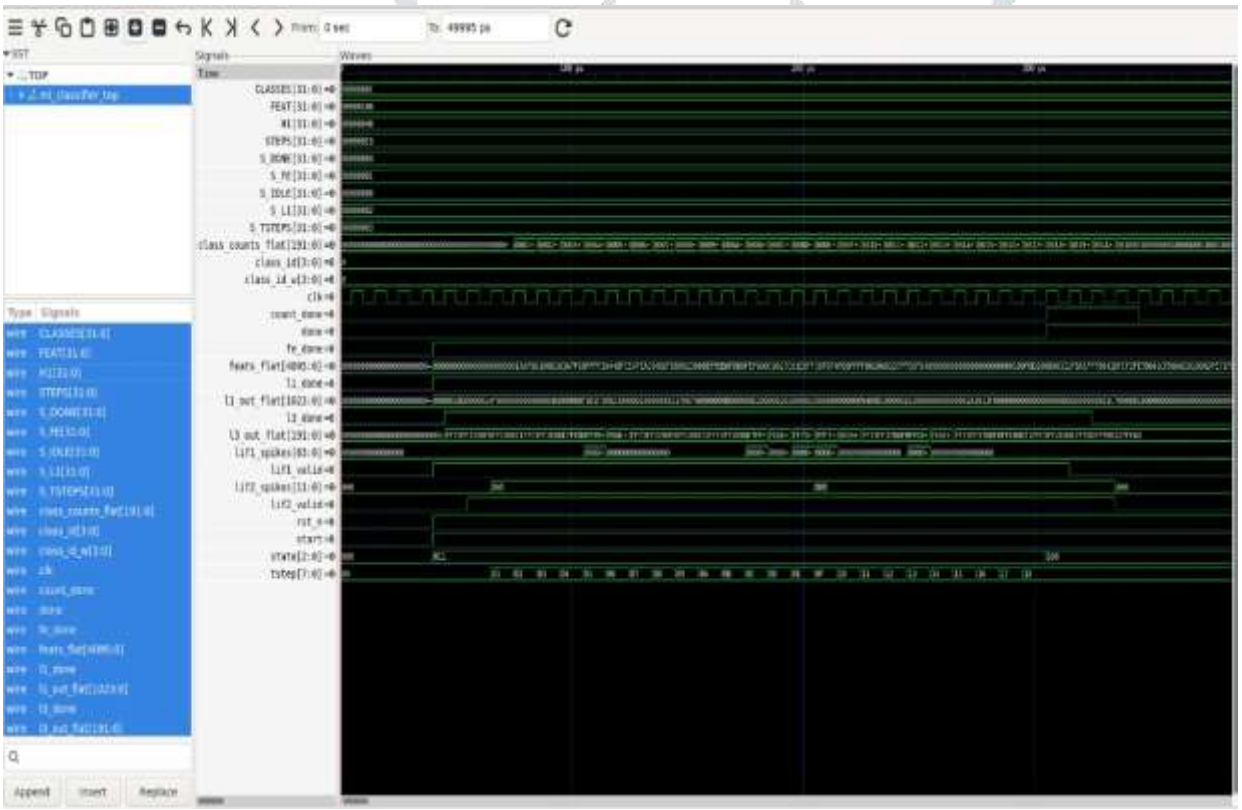


Fig4 Performance Overview of the SNN-Inspired VLSI Myocardial Infarction Classifier

Conclusion

The proposed **SNN-inspired VLSI architecture** for myocardial infarction detection demonstrates a reliable, low-power, and real-time solution for ambulatory cardiac monitoring. By integrating advanced signal preprocessing, the system effectively mitigates motion artifacts and background noise, ensuring accurate classification and precise localization of MI events from ECG signals even during physical activity. The design achieves high classification accuracy, minimal processing delay, and efficient area utilization, making it highly suitable for continuous wearable health monitoring. Overall, the proposed approach offers a practical and scalable hardware solution for early MI detection, potentially reducing the risk of delayed diagnosis and improving patient outcomes.

Future Scope

Integration with IoT and Cloud Platforms: Enable remote monitoring and real-time alerting for clinicians and caregivers.
Multi-Lead ECG Support: Extend the system to process multi-lead ECG signals for improved diagnostic accuracy.
Adaptive Learning: Implement on-chip adaptive algorithms or online learning to personalize detection for individual patients.

Extended Cardiac Event Detection: Expand classification to detect other cardiac abnormalities such as arrhythmias, ischemia, and heart failure.

Miniaturization and Energy Harvesting: Further reduce chip area and power consumption, potentially integrating energy harvesting for fully autonomous wearable devices

REFERENCES

[1] K. Thygesen, J. S. Alpert, A. S. Jaffe, M. L. Simoons, B. R. Chaitman, and H. D. White, "Third universal definition of myocardial infarction," *Circulation*, vol. 126, no. 16, pp. 2020–2035, 2012.

[2] D. Kasper, A. Fauci, S. Hauser, D. Longo, J. Jameson, and J. Loscalzo, *Harrison's Principles of Internal Medicine*, New York, NY, USA: McGraw Hill, 2015, doi: 10.1036/007149619X.

[3] P. Valensi, L. Lorgis, and Y. Cottin, "Prevalence, incidence, predictive factors and prognosis of silent myocardial infarction: A review of the literature," *Arch. Cardiov. Dis.*, vol. 104, pp. 178–188, Mar. 2011, doi: 10.1016/j.acvd.2010.11.013.

[4] X. Zhang, R. Li, H. Dai, Y. Liu, B. Zhou, and Z. Wang, "Localization of myocardial infarction with multi-lead bidirectional gated recurrent unit neural network," *IEEE Access*, vol. 7, pp. 161152–161166, 2019, doi: 10.1109/ACCESS.2019.2946932.

[5] R. K. Tripathy, A. Bhattacharyya, and R. B. Pachori, "Localization of myocardial infarction from multi-lead ECG signals using multiscale analysis and convolutional neural network," *IEEE Sensors J.*, vol. 19, no. 23, pp. 11437–11448, Dec. 2019, doi: 10.1109/JSEN.2019.2935552.

[6] J. Zhang et al., "Automated detection and localization of myocardial infarction with stacked sparse Autoencoder and TreeBagger," *IEEE Access*, vol. 7, pp. 70634–70642, 2019, doi: 10.1109/ACCESS.2019.2919068.

[7] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.

[8] R. Parmar, M. Janveja, J. Pidanic, and G. Trivedi, "Design of DNN-based low-power VLSI architecture to classify atrial fibrillation for wearable devices," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 3, pp. 320–330, Mar. 2023, doi: 10.1109/TVLSI.2023.3236530.

[9] R. Bousseljot, D. Kreiseler, and A. Schnabel, "Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet," *Biomedizinische Technik*, vol. 40, no. 1, pp. 317–318, 1995.

[10] G. Sahu and K. C. Ray, "An efficient method for detection and localization of myocardial infarction," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022, doi: 10.1109/TIM.2021.3132833.

[11] N. Sinha and A. Das, "Identification and localization of myocardial infarction based on analysis of ECG signal in cross spectral domain using boosted SVM classifier," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, Oct. 2021, doi: 10.1109/TIM.2021.3117663.

[12] J. K. Eshraghian et al., "Training spiking neural networks using lessons from deep learning," 2021, arXiv:2109.12894.

[13] M. Arif, I. A. Malagore, and F. A. Afsar, "Detection and localization of myocardial infarction using K-nearest neighbor classifier," *J. Med. Syst.*, vol. 36, pp. 279–289, Feb. 2012.

[14] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, and M. Adam, "Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals," *Inf. Sci.*, vol. 415–416, pp. 190–198, Nov. 2017.

[15] U. B. Baloglu, M. Talo, O. Yildirim, R. S. Tan, and U. R. Acharya, "Classification of myocardial infarction with multi-lead ECG signals and deep CNN," *Pattern Recognit. Lett.*, vol. 122, pp. 23–30, 2019.

Author Details:

D. Divakar received a diploma in Electronics and Communication Engineering from the government polytechnic college narsipatnam, Anakapalli (dist), India and received B.Tech degree in Electronics and Communication Engineering from MVGR college of engineering Vizianagaram, India and pursuing M.Tech in VLSI at sanketika vidya parishad engineering college affiliated to Andhra University, Visakhapatnam, Andhra Pradesh, India. Email ID : divakardraksharapu1@gmail.com , contact no: 9866913663.

Mrs. B.Sarada received her M.E from Andhra University 2010, Visakhapatnam, Andhra Pradesh, India. Total years of experience is 16 years.Now working as an Assistant Professor at sanketika vidya parishad engineering college Visakhapatnam affiliated to Andhra University , Andhra Pradesh, India. Email ID : sarada.ece@svpec.edu.in.