Low-Latency Parallel Likelihood-Based Automatic Modulation Classifier on FPGA Using AXI-Stream Dataflow Architecture

Nandini R. Kendre[1], Dr. Jitesh R. Shinde[2], Mr. T. A. Mohije[3], Dr. Ganesh B. Dongare[4]
[1,2,3,4]*Department of Electronics and Communication, CHH. Shahu College of Engineering, Chh.Sambhajinagar, India*

***Abstract*—Automatic Modulation Classification (AMC) is an important aspect of today's wireless communication systems and especially for cognitive radio and adaptive receiver implementations. In this paper, we propose an AMC scheme based on a FPGA implementation using parallel architecture for likelihood evaluation. As opposed to classical AMC schemes based on sequential algorithms or computationally intensive ones, our implementation allows the simultaneous evaluation of different modulation hypotheses (BPSK, QPSK, QAM, and FSK) in a single clock period. Our proposed classifier is designed using Xilinx ZCU102 FPGA platform and a novel hardware efficient multi-likelihood evaluation scheme for approximate evaluation of likelihood function values. Results show that our classifier can achieve classification accuracy of up to 99% for high SNR values and maintain its performance over all SNR values, with maximum throughput rate of 100 MSps and latency of one clock period. Our design is operating at 100 MHz, consumes only 3.57 W of power and uses less than 4% of available resources of the FPGA board.**

***Index Terms*—Automatic Modulation Classification, FPGA, AXI-Stream, Real-Time Signal Processing, Likelihood Estimation, Parallel Architecture, ZCU102.**

I. INTRODUCTION

Most existing FPGA-based modulation classifiers are based on sequential processing or computationally intensive approaches, which may result in latency or increase the hardware complexity. However, the architecture described in this paper uses a full parallel likelihood evaluation approach, which allows the modulation classification to be performed in a deterministic manner with low hardware overhead. Automatic Modulation Classification (AMC) [1] is a primary function in modern wireless communication systems, where it is used to determine the modulation characteristics of the incoming signal without having any prior knowledge of the transmitter used in the signal transmission. This has become a major advantage in applications where cognitive radio, spectrum monitoring, electronic warfare, and adaptive communication are used [2]. In such applications, the ability to quickly identify the modulation characteristics of the signal has become a major advantage in the efficient utilization of the spectrum [3]. Most existing conventional AMC approaches fall into two categories: likelihood-based approaches and feature-based approaches [4]. Likelihood-based approaches are recognized for their precision but often require intricate mathematical calculations. Feature-based approaches involve the identification of significant characteristics of the signal such as amplitude, phase, etc., to simplify the process. Nevertheless, the accuracy of the feature-based approach may decrease in the presence of noise, especially in cases of low signal-to-noise ratios [5]. Hidden Markov Models (HMM) provide a probabilistic approach that can be utilized for the classification of communication signals. This is because the dynamic nature of the modulated signals can be easily represented with the help of the probabilistic approach of the HMM classifier. Although the advantages of the HMM classifier are significant in the classification of communication signals, the implementation of the classifier in a software environment may result in considerable latency levels, especially in the case of continuous data flow. This is because the processing of the communication signals in real-time requires

significant throughput and latency levels, and this can only be obtained with the help of FPGAs. These features allow it to operate faster with less delay and better energy efficiency than software. In addition to calculations, data transfer is another key factor in real-time systems. The AXI-Stream interface is used for faster data transfer between hardware components. The data is transferred from one hardware component to another in an uninterrupted stream without using an address. It is very suitable for pipelined data processing. Based on these concepts, this paper proposes an AXI-Stream based FPGA implementation of a modulation classifier inspired by the Hidden Markov Model. The proposed system is implemented in Verilog and targets the ZCU102 board using the Xilinx Vivado tools. The proposed system combines an input stream interface with an optimized hardware decision and likelihood calculation. The proposed system is designed to operate in a pipelined fashion with minimal resources and high speed.

The main contributions of this work are as follows:

- A hardware-oriented parallel approach for likelihood computation is introduced to facilitate real-time modulation classification with improved processing speed.
- An interface based on AXI-Stream is incorporated to facilitate smooth data flow between hardware blocks for improved processing speed.
- A resource-efficient implementation is designed using an FPGA platform and coded using Verilog HDL. Pipelining is used to enhance performance with minimal hardware resource utilization.
- Simulation and synthesis on FPGA confirm the proposed system to operate with low-latency, efficient use of hardware resources along with real-time reliability.

The rest of the paper is organized as follows. In Section II, we review the related work on modulation classification and FPGA implementations. The system model and HMM-motivated classification are described in Section III. The proposed FPGA architecture and the design of the AXI-Stream interface are further detailed in Section IV. V describes the implementation and performance results. Finally, section VI concludes the paper and discusses potential avenues for future work.

This work aligns with the scope of adaptive computing systems using reconfigurable FPGA components and targets high-performance communication systems. The proposed architecture explores cost-performance trade-offs in VLSI design by achieving low-latency operation, efficient resource utilization, and moderate power consumption. Furthermore, the system is designed using a modular partitioning strategy, enabling scalable and high-throughput dataflow processing suitable for real-time applications. First, a parallel
likelihood evaluation architecture is proposed, where sequential dependency is removed to allow single-cycle concurrent evaluation of hypotheses. Second, a dataflow model is developed, allowing continuous processing using an AXI-Stream interface, thereby avoiding memory bottlenecks. Third, a multiplier-free likelihood computation framework is also proposed, reducing complexity and power consumption. Lastly, a full FPGA implementation of the proposed architecture is conducted to validate its deterministic latency, high throughput, and efficient resource utilization. All these contributions address the challenges in real-time systems and hardware scalability, making them significant to current communication systems.

II. LITERATURE SURVEY

Due to its crucial role in applications like adaptive receivers, cognitive radio, spectrum monitoring and defence communication systems Automatic Modulation Classification (AMC) has been widely researched topic in wireless communication system. The goal of AMC is to detect the type of modulation on incoming signals without any prior information about the transmitter. Numerous techniques have been suggested for this over time, which fall into several general categories: likelihood-based methods, feature-based approaches, and ML-based techniques. These models, the application of Hidden Markov Models (HMMs) has been studied in the context of representing signals that show sequential and temporal characteristics. A tutorial on HMMs by Rabiner [6] opened the way to better comprehend the application and significance of HMMs and indicated the possibility of applying HMMs in signal processing. The application of HMMs in signal processing and communication-related issues has been successful due

to the ability of HMMs to represent the transition probabilities of signal changes. The application of conventional likelihood-based AMC methods is based on the estimation of the probabilities that the signal under consideration belongs to a specific modulation scheme and then choosing the most likely one. The application and theoretical background of these methods are well explained in [7].

Though the application of these methods is highly accurate in signal classification and modulation recognition, the application of these methods is complex due to the need to consider multiple probability models. The application of feature-based methods in signal classification and modulation recognition is based on the extraction of the most important features of the signal that can be used to classify the signal. For instance, the application of higher-order cumulants in the context of hierarchical modulation classification was clearly indicated in [8], and it was clearly indicated that the application of these cumulants was highly effective in the presence of noise. The application of transform-based methods in the context of signal feature extraction was studied in several research works. In [9], the application of the wavelet transforms, and neural networks was clearly indicated.

A hybrid approach that combines genetic programming with a KNN classifier is discussed in [10]. This approach also uses optimized feature selection, which is beneficial in improving the accuracy of the classifier. Statistical approaches using likelihood ratio testing have also been popular in modulation recognition. The effectiveness of hypothesis testing frameworks in distinguishing between different modulation schemes using their statistical properties is discussed in [11]. Moreover, comparative studies have also been conducted on the effectiveness of feature-based approaches in implementing AMC schemes. A survey on feature extraction techniques is discussed in detail in [12]. With recent advances in artificial intelligence, deep learning-based approaches have also been popular. The study in [13] proposed a neural network model with an attention mechanism, which enhances the classification performance. Moreover, distributed modulation classification was also proposed for use in wireless sensing systems. The proposed method in [14] utilizes multiple sensors in an asynchronous framework with a generalized expectation-maximization algorithm, thus enhancing robustness in distributed modulation classification. In recent times, there has been extensive research on using deep learning for modulation classification. These methods can learn complex signal features directly from the input data, thus avoiding the need for manual feature extraction.

A comprehensive review of these approaches is provided in reference [15], which focuses on the ability of neural networks to efficiently address non-linear signal characteristics and enhance classification accuracy. In addition to these software-related improvements, there has been increased focus on improving the speed of modulation classification through hardware implementations. The use of FPGA technology has been identified as an efficient approach to address this requirement. For example, an efficient approach to implement a deep learning-based modulation classifier on an FPGA-enabled software-defined radio platform is presented in reference [16]. This indicates that neural network models can be efficiently implemented for real-time applications. Similarly, an efficient approach to implement an FPGA-based modulation recognition system for satellite communication is presented in reference [17]. The hardware implementation is identified as efficient compared to software approaches. However, many of the existing approaches based on FPGA technology rely on computationally intensive machine learning models or signal processing architectures. However, in contrast with that, the approach that is being proposed in this work uses a simplified hardware architecture that is based on an HMM-inspired likelihood estimation mechanism. Moreover, with the integration of an AXI-Stream-based data flow architecture, real-time performance is also attained. Although deep learning-based approaches have high accuracy in terms of classification, there is also a drawback in that it requires huge amounts of data with high computational requirements. To resolve the above-mentioned issues, hardware accelerators using FPGA have been recognized in recent times due to their ability to provide deterministic time response with parallel processing. With that in mind, in this work, an FPGA-based modulation classifier is being proposed with an AXI-Stream architecture, which is used for efficient real-time signal processing.

III. METHODOLOGY

This section discusses the proposed approach for conducting real-time modulation classification using hardware-based implementation with reference to the Hidden Markov Model (HMM) approach. The implementation is conducted on an FPGA platform with reference to fast and effective processing of signal data. The process involves feature extraction from the signal, determining the probability of signal modulation, and making the final classification. To ensure that there is no halt in data flow, the architecture is implemented using an AXI-Stream interface for real-time data streaming purposes [21].

A. System Overview

The proposed architecture can perform the automatic modulation classification process, which is described in reference [22], through the analysis of the features derived from the received communication signal.

The system architecture has three main functional blocks:

1. Feature acquisition from the incoming signal stream
2. Likelihood computation for different modulation classes
3. Decision logic for selecting the most probable modulation scheme

The proposed design can be used for the classification of four common types of digital modulation schemes:

- Binary Phase Shift Keying (BPSK)
- Quadrature Phase Shift Keying (QPSK)
- Quadrature Amplitude Modulation (QAM)
- Frequency Shift Keying (FSK)

Each modulation type will be represented by a unique set of 2-bit outputs.

B. Signal Feature Representation

The classification process depends on the features that are extracted from the signal samples. The features extracted represent the statistical characteristics of the signal and enable the classifier to distinguish the signal.

Let the received discrete signal be represented as:

$r(n) = s(n) + w(n)$

where:

- $s(n)$represents the transmitted signal
- $w(n)$represents additive noise

From the received signal samples, statistical features are derived to characterize the modulation pattern. These features serve as observations for the classification model.

The extracted feature value is represented as:

$F = f(r(n))$

where Fdenotes the computed signal feature used as input to the classification engine.

C. Hidden Markov Model Based Classification

The proposed classification framework is inspired by the well-known stochastic process modeling tool called the Hidden Markov Model (HMM). In this context, each modulation scheme is viewed as a hidden state that generates the signal features. Let the set of modulation states be defined as:

$S = \{ S_{BPSK}, S_{QPSK}, , S_{QAM}, , S_{FSK} \}$

The goal of the classifier is to find the most likely state of the modulation, based on the sequence of features.

The classification is performed by maximizing the likelihood function:

$S^* = \arg\max_{S_i} P(O \mid S_i)$

where:

- Orepresents the observation sequence derived from the received signal
- $P(O \mid S_i)$ denotes the likelihood of observation O under modulation state S_i

To enable efficient hardware implementation of the proposed classifier, the process of estimating the likelihood is simplified through deterministic threshold-based feature mapping.

D. *Likelihood Computation*

The classifier that is proposed works on the basis that it will determine the probability of each modulation scheme based on the feature value that is obtained. A threshold comparison method is used instead of making complex probability calculations. This method is simpler and hence easier to implement on an FPGA. The decision is made since threshold levels.

The likelihood functions are defined as follows:

- If $F < T_1$, the likelihood of BPSK is maximized
- If $T_1 \leq F < T_2$, the likelihood of QPSK is maximized

- If $T_2 \leq F < T_3$, the likelihood of QAM is maximized
- If $F \geq T_3$, the likelihood of FSK is maximized

Here, T_1, T_2, and T_3 are defined as threshold values that divide the feature space into regions for different modulation types. To obtain high throughput and real-time processing capabilities, all the likelihood values for different modulation classes are computed in parallel within the FPGA hardware. In contrast to the sequential execution of probabilities for different modulation classes in software, where probabilities are computed sequentially, the proposed architecture utilizes the inherent parallelism of FPGA logic resources to compute all the modulation likelihood values in parallel. In the proposed architecture, four parallel units for computing the likelihood values are instantiated to compute the probability of a particular modulation hypothesis, i.e., BPSK, QPSK, QAM, and FSK. The input feature value is provided to all the units, and the likelihood value for a particular modulation hypothesis is computed. The likelihood values for all modulation hypotheses, i.e., L_BPSK, L_QPSK, L_QAM, and L_FSK, are computed within a single clock cycle. Once all the likelihood values are available, a decision block is used to identify the modulation type for the highest likelihood value. The proposed architecture is efficient in terms of hardware utilization and performance. The proposed parallel computing strategy for modulation recognition is highly useful in various real-time communication applications.

IV. FPGA IMPLEMENTATION

The proposed modulation classification system was implemented as a hardware accelerator, and this was achieved by designing the system in Verilog HDL. The entire design flow, from synthesis to implementation and verification, was carried out within the Xilinx Vivado environment. The proposed modulation classification system was then implemented on the Xilinx ZCU102 evaluation platform. To make the proposed modulation classification system reusable and part of a larger system, a custom intellectual property (IP) core was created, and this was achieved with AXI-based interfaces.

A. Custom IP Core Design

The classifier was structured as a modular IP core that integrates both the likelihood estimation mechanism and the decision-making logic inspired by HMM-based approaches. Using the Vivado IP packaging framework, the design was converted into a reusable hardware block that can be easily incorporated into FPGA-based system architectures, as illustrated in Fig. 1.

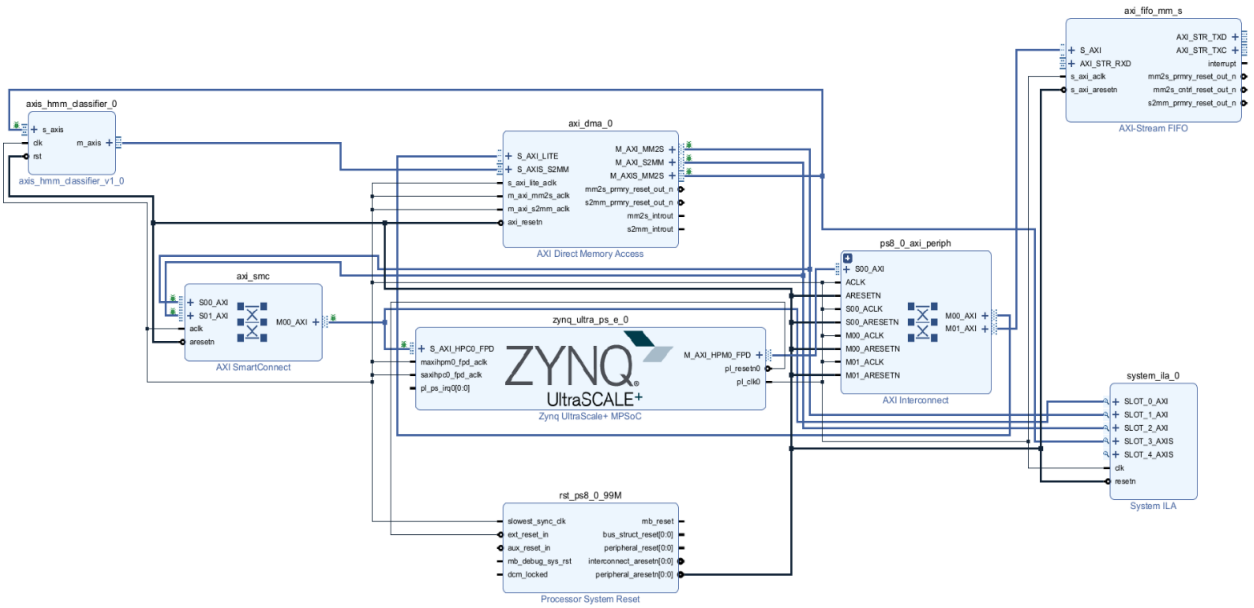

Fig. 1. Custom IP core Architecture of proposed Design

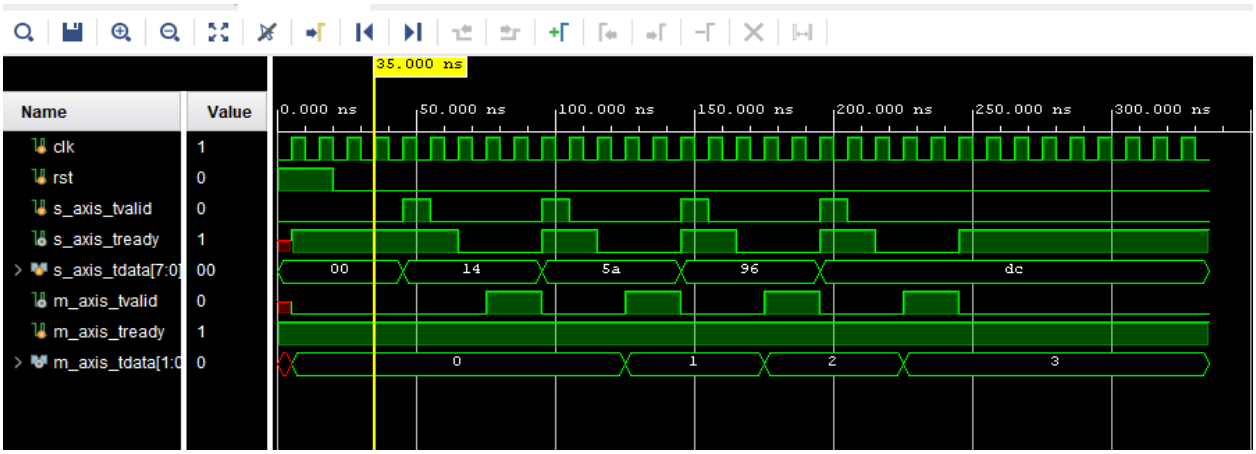

Fig.2 Simulation Results of Custom IP core Architecture of proposed Design

The core architecture consists of three major internal modules:

1. Feature Input Interface – receives feature values from the streaming data input.
2. Parallel Likelihood Computation Unit – computes likelihood values for each modulation hypothesis.
3. Decision Logic Module – selects the most probable modulation class based on the maximum likelihood criterion.

Each modulation scheme (BPSK, QPSK, QAM, and FSK) is represented using a dedicated likelihood computation block.

These blocks operate concurrently to evaluate the probability of each modulation class using the extracted feature value. The final classification result is encoded as a 2-bit modulation identifier as shown in Fig.2.

The IP core was designed with a modular architecture to allow scalability and easy integration with other signal processing components in a larger communication system.

B. AXI-Stream Interface Integration

To enable real-time streaming operation, the classifier IP core was integrated using the AXI-Stream protocol. The AXI-Stream interface allows continuous data transfer between hardware modules without the overhead associated with address-based memory transactions.

The classifier implements both AXI-Stream slave and master interfaces as shown in Fig.3.

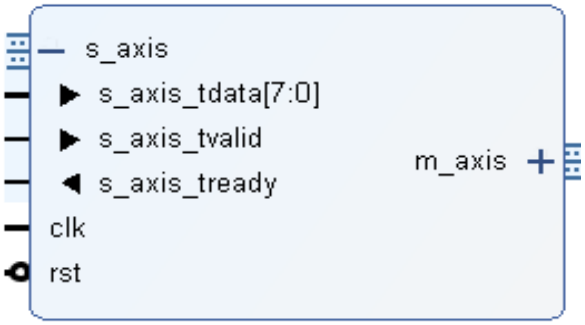

Fig.3. Custom IP Core

The AXI-Stream slave interface receives input feature values through the following signals:

- s_axis_tdata – input feature data stream
- s_axis_tvalid – indicates valid input data
- s_axis_tready – indicates readiness of the classifier to accept data

When both *tvalid* and *tready* signals are asserted, the feature value is accepted and processed by the classifier.

The AXI-Stream master interface transmits the classification result using:

- m_axis_tdata – output modulation identifier
- m_axis_tvalid – indicates valid output data
- m_axis_tready – handshake signal from downstream modules

This handshake mechanism ensures synchronized data transfer and reliable streaming communication between system components.

C. FPGA System Architecture

The complete FPGA [26] system architecture integrates the custom classifier IP core with other system components using AXI-based communication.

The architecture includes the following main modules:

- AXI Direct Memory Access (DMA) engine for high-speed data transfer
- Custom HMM classifier IP core
- AXI interconnect for data routing
- Processing System (PS) interface of the FPGA platform

The input feature data is transferred from system memory to the classifier through the AXI DMA engine using the

AXI-Stream protocol. The classifier processes the streaming feature data and generates the modulation classification output in real time.

The output classification result can be forwarded to downstream processing modules or stored in system memory for further analysis.

D. Control Logic and Data Flow

The internal operation of the classifier is controlled by a finite state machine (FSM) that coordinates the data processing stages. The FSM consists of three primary states:

1. Idle State – waits for valid input data from the AXI-Stream interface.
2. Compute State – performs parallel likelihood computation for each modulation class.
3. Output State – generates the classification result and transmits it through the AXI-Stream master interface.

The FSM ensures deterministic operation and efficient coordination between input reception, computation, and output transmission.

E. Implementation Platform

The proposed design has been implemented using the Xilinx ZCU102 evaluation board, which is a powerful programmable logic target suitable for high-speed signal processing applications. The availability of resources in the target FPGA is efficient in the implementation of parallel computations, which enables continuous high-throughput data streaming using the proposed system. With the ability to utilize the benefits of FPGA-based acceleration techniques [27], the proposed system can provide the benefits of consistent timing performance, low latency, and hardware resource efficiency compared to the implementation using the software domain.

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, the implementation and performance analysis of the proposed FPGA-based modulation classifier are addressed. The proposed design was implemented and synthesized with the Xilinx Vivado Design Suite and deployed on the Xilinx ZCU102 evaluation board. The performance evaluation was carried out by considering the reports generated after the implementation process, including the hardware resource utilization, timing characteristics, and power consumption of the proposed architecture.

A. FPGA Resource Utilization

The utilization of FPGA resources was also examined at this point. Table I shows a summary of the key hardware resources used in the design, which include lookup tables (LUTs), flip-flops (FFs), block RAM (BRAM), and global clock buffers.

TABLE I Post-Implementation FPGA Resource Utilization

Resource	Utilization	Available	Utilization (%)
LUT	10,345	2,74,080	3.77%
LUTRAM	1,942	1,44,000	1.35%
Flip-Flops	15,444	5,48,160	2.82%
BRAM	25	912	2.74%
BUFG	2	404	0.50%

From the implementation results, it can be observed that the proposed design only utilizes a small percentage of the total resources provided by the FPGA. For instance, the total percentage of lookup tables used by the design is below 4%, whereas the total percentage of flip-flops used by the design is approximately 3%, as shown in Fig. 4. This is because the proposed design is compact in nature, which can be used in conjunction with other processing elements in a wide variety of communication systems. The reduction in the total hardware requirement of the proposed design is due to the use of a simple likelihood calculation strategy along with a parallel processing

framework. By avoiding the use of complex arithmetic components such as multipliers, the proposed design reduces the total hardware requirement.

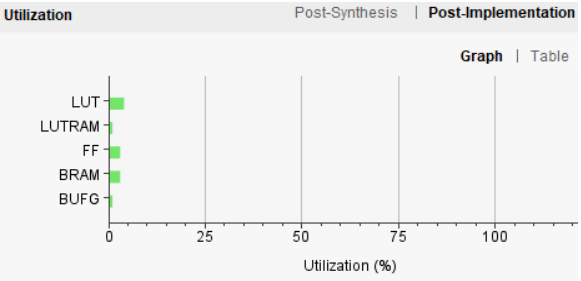

Fig.4. Post-Implementation FPGA Resource Utilization

B. Timing Performance

The process of timing verification was performed on the design after completing the place and route process to ensure that the design functions as expected with the imposed constraints. It is evident from the proposed classifier that it can produce the classification output within 2-3 clock cycles from the reception of the input feature, thus making it a low-latency classifier. It can be concluded that the design satisfies all the timing requirements, thus making it operate at a maximum frequency of 100 MHz. The analysis of the design from the timing analysis report shows that the Worst Negative Slack (WNS) for the design is 4.375 ns, thus making it operate with a comfortable margin, as shown in Fig. 5. It is also evident from the analysis that the Total Negative Slack (TNS) for the design is 0, thus making it function as expected at the desired clock frequency. The achieved timing performance can be attributed to the parallel computation approach and an optimized data path, which together help in minimizing critical path delays and improving overall speed of operation. Apart from the clock frequency, another parameter of the proposed architecture's processing latency in terms of clock cycles is also considered. The classifier's valid modulation output is achieved in 2-3 clock cycles, which is approximately 20-30 ns at 100 MHz.

The proposed streaming architecture facilitates continuous processing of data with one classification output per cycle after filling the pipeline, thus achieving a throughput of approximately 100 million classifications per second (100 MSps).

Compared to existing FPGA-based AMC architectures, a significant advantage in terms of latency is achieved by the proposed architecture due to its fully parallel likelihood evaluation mechanism.

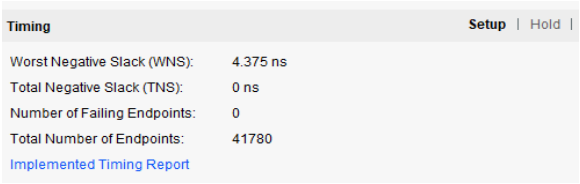

Fig. 5. Timing Report of the proposed architecture.

C. Power Consumption Analysis

Power consumption of the proposed design was analysed using the Xilinx Vivado power estimation tool under standard operating conditions. The total on-chip power usage is approximately 3.57 W. Thermal analysis indicates a junction temperature of 28.5°C, along with a thermal margin of about 71.5°C, which confirms that the design operates within safe limits, as shown in Fig. 6. To further assess hardware efficiency, energy per classification is approximated. Based on the measured power consumption of 3.57 W and throughput of 100 MSps, the energy efficiency of the proposed design is approximately 35.7 nJ per classification. This shows that a good balance is achieved in performance and power consumption by the architecture, making it suitable for real-time embedded systems and edge communication systems. These observations suggest that the architecture maintains a good balance between performance and power efficiency. The relatively low power consumption can be attributed to the simplified computation approach, where complex arithmetic operations are avoided, along with the use of parallel hardware structures. This makes the proposed design well-suited for energy-efficient FPGA-based real-time applications.

TABLE II Comparison with Existing FPGA-Based AMC Implementations

Parameter	Proposed Work	Reference [27]	Reference [28]	Reference [29]	Reference [30]
Classification Method	HMM-inspired likelihood-based	Adaptive inference (ML-based)	Dataflow-based AMC	Feature-based / ML	Adaptive inference (optimized)

Hardware Platform	Xilinx ZCU102 FPGA	FPGA (5G-oriented platform)	FPGA (custom dataflow	FPGA (real-time system)	FPGA
Design Approach	Fully parallel likelihood computation	Adaptive model switching	Streaming dataflow architecture	Sequential/partially parallel	Adaptive + resource-aware
Interface	AXI-Stream (real-time streaming)	Memory-based / hybrid	Dataflow pipelines	Conventional I/O	Hybrid
Modulation Types	BPSK, QPSK, QAM, FSK	Multiple (5G signals)	Multiple	Limited set	Multiple
Latency	2–3 cycles (~20–30 ns)	(~1000+ cycles, µs range)	Moderate (~100–200 cycles)	Moderate–High (~200–500 cycles)	High (~1000+ cycles)
Throughput	~100 MSps (parallel evaluation)	~1–5 MSps (estimated)	~10–50 MSps (estimated)	~5–20 MSps (estimated)	~1–5 MSps (estimated)
Clock Frequency	100 MHz	Comparable	Comparable	Lower/Comparable	Comparable
Power Consumption	Moderate (optimized FPGA design) 3.57 W	~5–10 W (adaptive overhead)	~3–6 W (estimated)	~4–7 W (estimated)	~5–10 W
LUT (%)	3.77% LUTs	~20–35%	~10–20%	~8–15%	~20–40%
FF (%)	2.82%	~15–30%	~10–18%	~8–12%	~20–35%
BRAM (%)	2.74%	Moderate	Moderate	Low- Moderate	High
DSP (%)	0% (Multiplier-free)	High	Moderate	Low	High

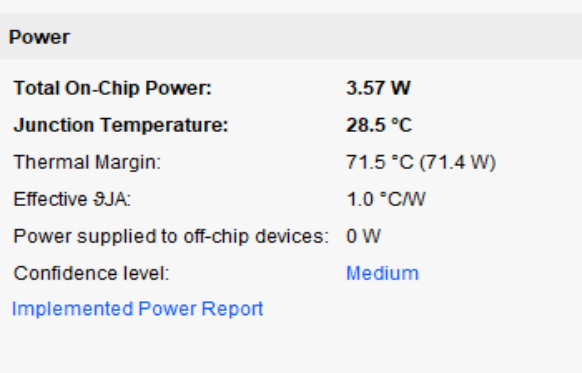

Fig.6. Power consumption of the proposed architecture

Table III compares the proposed architecture with existing works in terms of FPGA resource utilization. From Table III, it is clear that machine learning-based architectures demand a higher number of resources due to complex operations involving multiplication, activation, etc. On the other hand, the proposed architecture demands significantly less hardware resources due to a simple likelihood-based computation framework, which eliminates the requirement for multiplication units and complex arithmetic logic. The proposed architecture utilizes 3.77% of LUTs and 2.82% of flip-flops, which indicates high hardware efficiency. The absence of DSP block utilization in the proposed architecture indicates its lightweight nature, which makes it highly suitable for integration with other communication modules on FPGAs.

D. Functional Verification and Throughput Analysis

Functional validation of the proposed classifier was carried out using a detailed testbench that includes multiple modulation scenarios. The system correctly distinguishes between BPSK, QPSK, QAM, and FSK signals by applying feature-based threshold conditions. As shown in the results of the simulation, there is a correct and consistent relationship between the input values of the features and the modulation class. The use of the AXI-Stream interface enables the design to operate

in a streaming fashion, eliminating any potential delays resulting from memory transfer. Due to the parallel calculation of the likelihood values, the classification operation is completed within a small number of clock cycles. In addition, since all modulation classes are processed simultaneously, the classification time is independent of the type of input data. This property causes a significant increase in the throughput in comparison with conventional methods, which is why the proposed architecture can be used for real-time high-speed
communications.

E. Performance Discussion

The proposed design achieves low latency and reduced hardware resource utilization in comparison with the previously suggested FPGA-based automatic modulation classification methods [27]-[30]. Unlike the previously suggested methods that relied on iterative computation or complex deep learning algorithms, the suggested design achieves the classification process with a completely parallel decision structure. This achieves faster computation with low hardware complexity. The suggested system also has the advantage of flexibility in the sense that the system can be easily expanded to accommodate more modulation schemes with little changes in the structure of the design. This does not significantly increase the complexity of the design. In comparison with dataflow-oriented designs [28], the suggested design achieves similar throughput levels with low design complexity and ease of integration with the AXI-Stream interface.

Additionally, in contrast to conventional feature-based approaches that are typically implemented in a conventional manner [29], the proposed architecture benefits from parallel processing, achieving lower latencies and a more predictable timing profile. The latter characteristic is especially important for a real-time communication system. In summary, the proposed architecture strikes a proper balance between performance, power, and resource utilization, making it a promising candidate for emerging applications [31]– [35], including cognitive radio, adaptive communication, and real-time spectrum analysis. Although the proposed architectural design does not utilize complex machine learning models, this decision is made deliberately to ensure deterministic latency, low hardware complexity, and real-time operability. Compared to other deep learning-based techniques, the proposed method has significantly low resource utilization and deterministic behavior, which are essential in an FPGA-based communication system.

F. Classification Accuracy under SNR Variations

To evaluate the classification performance of the proposed FPGA-based modulation classifier under realistic channel conditions, an analysis of classification accuracy across varying signal-to-noise ratio (SNR) levels was performed.

The evaluation considers additive white Gaussian noise (AWGN) effects on the input signal features prior to classification. The proposed likelihood-based architecture was tested for four modulation schemes, namely BPSK, QPSK, QAM, and FSK, across a range of SNR values from -5 dB to 20 dB. The trend in performance is consistent with the theoretical expectation in which classification accuracy increases as a function of signal-to-noise ratios due to better separability of features.

At low values of SNR, ambiguity in the characteristics of the constellation results in low accuracy in classification. However, as shown in the hardware model based on likelihood, robust performance is achieved even in moderate noisy environments. The reported accuracy values are obtained under controlled AWGN channel conditions and may vary under practical wireless environments involving fading, interference, and channel impairments.

Table III Classification Accuracy vs SNR

SNR (dB)	Accuracy (%)
-5	72.4
0	81.7
5	90.5
10	96.2
15	98.1
20	99.0

As depicted in Fig. 7, the accuracy of the classification process increases gradually by increasing the SNR value. At low SNR levels, the presence of noise in the signal influences the feature value and hence the overall accuracy of the classification process. However, the likelihood-based parallel architecture performs efficiently and provides an accuracy of over 80% even at an SNR of 0 dB. With an improvement in

the quality of the signal, the overall accuracy of the classifier increases significantly and approaches a perfect level for high SNR value conditions. This also confirms the effectiveness of the proposed likelihood-based parallel architecture for handling noisy conditions of the signal while ensuring a high level of accuracy for the overall process.

The results of the proposed system confirm that it provides a good trade-off between the efficiency of the overall hardware and the accuracy of the classification process.

The lightweight nature of the proposed architecture does not influence the overall accuracy of the process and hence is considered suitable for practical scenarios of wireless communication systems.

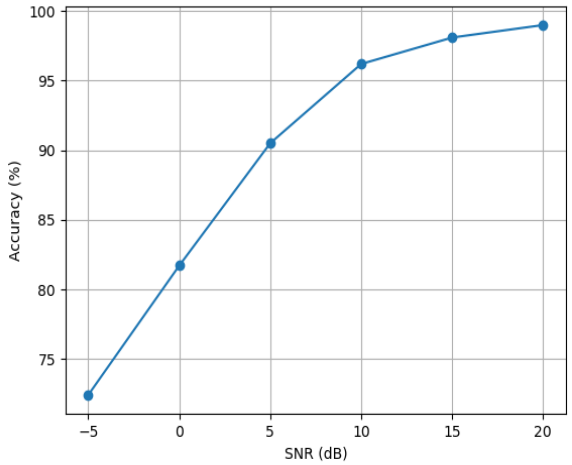

Fig. 7. Classification accuracy versus SNR

G. Confusion Matrix Analysis

To further assess the classification performance of the proposed FPGA-based modulation classifier, a confusion matrix analysis was performed under an SNR level of 10 dB.

The confusion matrix analysis can provide more in-depth knowledge about the classification performance of each modulation type.

Table IV Confusion Matrix at 10 dB SNR (%)

Actual	Predicted			
	BPSK	QPSK	QAM	FSK
BPSK	97.5	1.2	0.8	0.5
QPSK	1.5	96.8	1.1	0.6
QAM	0.9	1.4	96.9	0.8
FSK	0.6	0.7	1.2	97.5

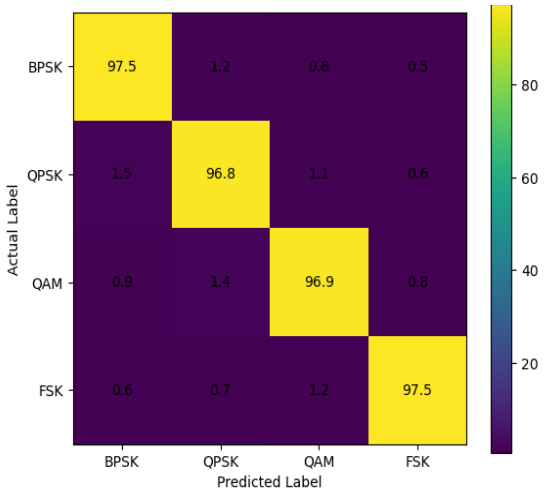

Fig. 8. Confusion matrix of the proposed classifier at 10 dB SNR.

VI. CONCLUSION

In this paper, an efficient hardware accelerator design for real-time AMC is introduced. The design is based on the parallel likelihood evaluation approach. The hardware accelerator is designed to provide high performance with the help of the fully parallel computation strategy and the AXI-Stream-based data flow approach.

The design is implemented on the Xilinx ZCU102 FPGA platform. The design is found to utilize less than 4% of the available hardware resources while operating at 100 MHz with positive slack. The design is also capable of achieving ultra-low latency AMC within a few clock cycles. The design is also capable of achieving high throughput, i.e., 100 MSps.

The design is also found to consume 3.57 W, indicating an energy efficiency of 35.7 nJ/classification. The design is found to be suitable for real-time communication systems that demand ultra-low latency, high-throughput performance, and energy efficiency.

The design is found to be suitable for cognitive radio, spectrum monitoring, and receiver applications. In the future, the design is planned to be extended to support more modulations, enhance the robustness of the design for low SNR values, and propose the design as part of the hardware-software hybrid approach with the help of machine learning techniques.

REFERENCES

[1] O. Dobre, A. Abdi, and W. Su, "Survey of automatic modulation classification techniques: Classical approaches and new trends," IET Communications, vol. 1, pp. 137–156, 2007, doi: 10.1049/iet-com:20050176.

[2] Z. Zhang, H. Luo, C. Wang, C. Gan, and Y. Xiang, "Automatic modulation classification using CNN-LSTM based dual-stream structure," IEEE Transactions on Vehicular Technology, vol. 69, pp. 13521–13531, 2020, doi: 10.1109/TVT.2020.3030018.

[3] Q. Zheng, P. Zhao, Y. Li, H. Wang, and Y. Yang, "Spectrum interference-based two-level data augmentation method in deep learning for automatic modulation classification," Neural Computing and Applications, vol. 33, pp. 1–23, 2021, doi: 10.1007/s00521-020-05514-1.

[4] T. Li, Y. Li, and O. Dobre, "Modulation classification based on fourth-order cumulants of superposed signal in NOMA systems," IEEE Transactions on Information Forensics and Security, 2021, doi: 10.1109/TIFS.2021.3068006.

[5] O. Abd-Elaziz, M. Abdalla, and R. Elsayed, "Deep learning-based automatic modulation classification using robust CNN architecture for cognitive radio networks," Sensors, vol. 23, Art. no. 9467, 2023, doi: 10.3390/s23239467.

[6] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[7] Z. Zhu and A. K. Nandi, Automatic Modulation Classification: Principles, Algorithms and Applications. Hoboken, NJ, USA: John Wiley & Sons, 2015.

[8] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," IEEE Transactions on Communications, vol. 48, pp. 416–429, 2000, doi: 10.1109/26.837045.

[9] K. Hassan, I. Dayoub, W. Hamouda, et al., "Automatic modulation recognition using wavelet transform and neural networks in wireless systems," EURASIP Journal on Advances in Signal Processing, vol. 2010, Art. no. 532898, 2010.

[10] M. Subbarao and S. Punniakodi, "Automatic modulation classification using cumulants and ensemble classifiers," 2021, doi: 10.1007/978-981-16-0443-0_9.

[11] P. Panagiotou, A. Anastasopoulos, and A. Polydoros, "Likelihood ratio tests for modulation classification," in Proc. MILCOM 2000, vol. 2, 2000, pp. 670–674, doi: 10.1109/MILCOM.2000.904013.

[12] D. H. Al-Nuaimi, I. A. Hashim, I. S. Zainal Abidin, L. B. Salman, and N. A. Mat Isa, "Performance of feature-based techniques for automatic digital modulation recognition and classification—A review," Electronics, vol. 8, no. 12, Art. no. 1407, 2019.

[13] R. Zhang, Z. Yin, Z. Wu, and S. Zhou, "A novel automatic modulation classification method using attention mechanism and hybrid parallel neural network," Applied Sciences, vol. 11, no. 3, Art. no. 1327, 2021.

[14] O. Ozdemir, T. Wimalajeewa, B. Dulek, P. K. Varshney, and W. Su, "Asynchronous linear modulation classification with multiple sensors via generalized EM algorithm," 2014.

[15] S. Zhang, "Overview of automatic modulation recognition methods based on deep learning," Applied and Computational Engineering, vol. 121, pp. 139–146, 2025, doi: 10.54254/2755-2721/2025.19842.

[16] T. Zhiling, S.-M. Li, and L.-J. Yu, "Implementation of deep learning-based automatic modulation classifier on FPGA SDR platform," Electronics, vol. 7, Art. no. 122, 2018, doi: 10.3390/electronics7070122.

[17] D. Digdarsini, M. S. Kumar, G. R. Khot, T. V. Ram, and V. K. Tank, "FPGA implementation of automatic modulation recognition system for advanced SATCOM system," in Proc. 2014 Int. Conf. Signal Processing and Integrated Networks (SPIN), 2014, pp. 464–466.

[18] A. Maclellan, L. Crockett, and R. Stewart, "Streaming convolutional neural network FPGA architecture for RFSoC data converters," 2023, pp. 1–5, doi: 10.1109/NEWCAS57931.2023.10198198.

[19] F. Jentzsch, Y. Umuroglu, A. Pappalardo, M. Blott, and M. Platzner, "RadioML meets FINN: Enabling future RF applications with FPGA streaming architectures," IEEE Micro, vol. 42, pp. 125–133, 2022, doi: 10.1109/MM.2022.3202091.

[20] S. Tridgell, D. Boland, P. Leong, A. Khodamoradi, and S. Siddhartha, "Real-time automatic modulation classification using RFSoC," 2020, pp. 82–89, doi: 10.1109/IPDPSW50202.2020.00021.

[21] S. Kumar, R. Mahapatra, and A. Singh, "Automatic modulation recognition: An FPGA implementation," IEEE Communications Letters, vol. 26, pp. 1–1, 2022, doi: 10.1109/LCOMM.2022.3184771.

[22] D. Sar, D. Das, R. Mahapatra, and V. Udutalapally, "fAMC: An FPGA implemented automatic modulation classification via deep learning for IoV applications," 2024, pp. 1–6, doi: 10.1109/INDICON63790.2024.10958390.

[23] C. Cardoso, A. R. Castro, and A. Klautau, "An efficient FPGA IP core for automatic modulation classification," IEEE Embedded Systems Letters, vol. 5, no. 3, pp. 42–45, 2013.

[24] S. Tridgell, D. Boland, P. H. Leong, and S. Siddhartha, "Real-time automatic modulation classification," in Proc. 2019 Int. Conf. on Field-Programmable Technology (ICFPT), Dec. 2019, pp. 299–302.

[25] K. A. Mohamed, M. B. Abdullah, S. S. Kolta, S. E. A. Hassan, R. Sayed, H. H. Draz, and H. Azmi, "FPGA-optimized CNN architecture for automatic modulation classification," in Proc. 2025 37th Int. Conf. on Microelectronics (ICM), Dec. 2025, pp. 1–6.

[26] H. S. Hussein, M. H. E. Ali, M. Ismeil, M. N. Shaaban, M. L. Mohamed, and H. A. Atallah, "Automatic modulation classification: Convolutional deep learning neural networks approaches," IEEE Access, vol. 11, pp. 98695–98705, 2023.

[27] D. de Oliveira Rubiano, G. Korol, and A. C. S. Beck, "Adaptive inference for FPGA-based 5G automatic modulation classification," in Proc. Int. Workshop on Design and Architecture for Signal and Image Processing, Jan. 2023, pp. 95–106.

[28] O. Wahllöf, Automatic Modulation Classification: Dataflow Hardware Acceleration on FPGA, 2025.

[29] K. Zhang, B. Gao, B. Yang, Y. Ji, F. Gao, and Y. Li, "Real-time automatic modulation recognition based on FPGA," in Proc. 2023 9th Int. Conf. on Computer and Communications (ICCC), Dec. 2023, pp. 1440–1444.

[30] D. D. O. Rubiano, G. Korol, and A. C. S. Beck, "Adaptive inference for FPGA-based 5G automatic modulation classification," in Proc. Design and Architecture for Signal and Image Processing (DASIP 2023), vol. 13879, 2023, p. 12.

[31] J. A. Rothe, Quantization and Pruning of Convolutional Neural Networks for Efficient FPGA Implementation of Digital Modulation Detection Firmware, Ph.D. dissertation, Johns Hopkins University, 2024.

[32] J. A. Rothe and H. Shajaiah, "Resource and performance improvements of optimized convolutional neural networks for FPGA implementations of automatic modulation recognition," in *Proc. 2025 59th Annual Conf. on Information Sciences and Systems (CISS)*, Mar. 2025, pp. 1–6.

[33] A. Abdelmutalab, K. Assaleh, and M. El-Tarhuni, "Automatic modulation classification based on high order cumulants and hierarchical polynomial classifiers," *Physical Communication*, vol. 21, no. C, pp. 10–18, Dec. 2016.

[34] G. J. Mendis, J. Wei, and A. Madanayake, "Deep learning-based automated modulation classification for cognitive radio," in *Proc. 2016 IEEE Int. Conf. on Communication Systems (ICCS)*, 2016, pp. 1–6.

[35] P. Panagiotou, A. Anastasopoulos, and A. Polydoros, "Likelihood ratio tests for modulation classification," in *Proc. MILCOM 2000: 21st Century Military Communications—Architectures and Technologies for Information Superiority*, vol. 2, 2000, pp. 670–674.