Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

FPGA-based Modulation Classification using Verilog Simulation

Afsal M M

Department of Electronincs, St:Joseph College of Engineering Palai

Abstract
Automatic modulation classification (AMC) plays a crucial role in modern com- munication systems, particularly in applications such as cognitive radio, spec- trum monitoring, and software defined radios (SDRs). Traditional AMC tech- niques rely heavily on software-based processing, which is computationally ex- pensive and unsuitable for real- time embedded systems. In this work, we present a lightweight, Verilog HDL-based design for modulation classification targeting VLSI/FPGA platforms. The system is designed with modular blocks for feature extraction and classification and is evaluated using Vivado simulator. A zero- crossing rate (ZCR), amplitude based and signal energy feature extraction ap- proach is employed to differentiate between basic modulation schemes such as BPSK and QPSK. The classification is implemented as a simple threshold-based decision logic in Verilog HDL. Simulation results confirm that the proposed de- sign successfully classifies signals with high accuracy under noise-free and mod- erate noise conditions. The work demonstrates the feasibility of integrating AMC algorithms into hardware-efficient FPGA- based systems, paving the way for fu- ture extensions to higher-order modulation schemes.

Keywords: Automatic Modulation Classification (AMC), Verilog HDL, VLSI, FPGA, BPSK, QPSK, Signal Processing, Vivad

1. Introduction

Wireless communication systems employ a wide range of BPSK, QPSK, QAM, and digital modulation schemes such as OFDM to optimize data transmission under varying channel conditions. Identifying the modulation scheme at the re- ceiver, often termed as automatic modulation classification (AMC), is a vital task in applications including and cognitive radios. spectrum sensing, electronic war- fare, Conventional AMC methods typically use software-based implementations in MATLAB or Python, which though flexible, cannot meet the strict real-time and low-power constraints of embedded or VLSI-based systems. With the growing demand for high-speed communication, there is an increasing need for hardware- efficient AMC designs ASICs. that can be implemented in FPGAs or Verilog HDL provides a powerful hardware description mechanism that enables modular design and synthesis of digital systems for FPGA and VLSI applications. This project explores the design and simulation of a Verilog-based AMC framework, where feature extraction and classification modules are built directly in hardware. The work emphasizes resource-efficient design while maintaining classification accuracy, serving as a foundation for more advanced FPGA- based AMC imple- mentations.

2. Literature Review

2.1 Feature-based AMC:

statistical approaches to modern deep Abdel-Meniem et al. [1] demonstrated FPGA implementation of AMC using wavelet transforms, showing real-time feasibility but at higher resource cost. Castro et al. [2] applied cyclostationary features accuracy with complexity. and SVM classifiers in hardware for cognitive radios, balancing

2.2 Verilog/FPGA Implementations:

Hemanth Kumar et al. [3] implemented basic modulation schemes such as BASK, BFSK, BPSK, and QPSK in Verilog HDL, demonstrating feasibility of signal- level designs. Tanawade and Sudhansu [4] presented FPGA implementations of BPSK/QPSK, focusing on resource utilization.

2.3 Deep Learning Approaches:

Recent works [5][6] employ CNNs and autoencoders for AMC, achieving supe- rior accuracy but requiring high hardware resources. For example, Tang et al. [5] implemented deep learning-based AMC on FPGA but faced power and resource challenges.

3. Design and Implementation

The overall system consists of four major stages:

1. **Input Signal Acquisition**
2. **Feature Extraction**
3. **Signal Classification**
4. **Output Encoding**

A conceptual block diagram of the proposed methodology is shown in Figure 3.1.

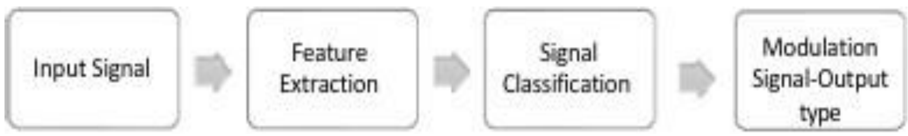

Figure 3.1: Proposed system block diagram showing signal flow from input ac- quisition to classification.

The overall system architecture for the proposed FPGA-based automatic mod- ulation classification (AMC) consists of four major functional blocks — Input Signal Acquisition, Feature Extraction, Classification, and Output Encoding. The block diagram illustrates the sequential flow of data from the signal input to the final classification output. Modulated signals such as BPSK and QPSK are first supplied to the system as input waveforms. These signals are then processed by the Feature Extraction block, which computes measurable parameters like zero- crossing rate, amplitude, and signal energy. The extracted features are passed to the Classifier, which uses a threshold-based decision logic implemented in Verilog HDL to determine the modulation type. The final Output Encoding block con- verts this classification decision into binary labels (e.g., “00” for BPSK and “01” for QPSK). The design follows a modular and hierarchical structure, ensuring that each block can be independently tested, verified, and scaled for higher-order modulation schemes.

3.1 Input Signal

In digital communication systems, modulation plays a vital role in transmitting digital information through analog carrier waves. The proposed design focuses on classifying two fundamental phase modulation schemes — Binary Phase Shift Keying (BPSK) and Quadrature Phase Shift Keying (QPSK) — which form the basis of many modern wireless communication standards.

3.1.1 Binary Phase Shift Keying (BPSK):

BPSK is the simplest form of phase modulation where each data bit is represented by one of two possible phase states of a carrier signal — typically 0° and 180°. When the transmitted bit is "1," the carrier phase remains unchanged (0°), and when the bit is "0," the phase is inverted (180°). This results in a waveform that alternates polarity according to the binary data sequence. Mathematically, the modulated signal can be expressed as: Mathematically:

$$s(t) = A\cos\ \ 2\pi ft + \pi b \tag{3.1}$$

where:

- A is the carrier amplitude,
- f is the carrier frequency, and
- $b \in \{0, 1\}$ represents the binary data bit.

The output waveform thus alternates between $+A$ and $-A$, making BPSK ro- bust against noise and suitable for low-bandwidth, high-reliability communication links.

3.1.2 Quadrature Phase Shift Keying (QPSK):

QPSK extends the principle of BPSK by encoding two bits per symbol, effectively doubling the data rate without increasing the bandwidth. It uses four distinct phase shifts — 0°, 90°, 180°, and 270° — each representing a unique two-bit combina- tion such as "00," "01," "10," or "11." The QPSK signal can be mathematically written as:

$$s(t) = A\cos\ \ 2\pi ft + \phi \tag{3.2}$$

where ϕ takes one of four possible values $(0, \pi/2, \pi, 3\pi/2)$ corresponding to the input bit pairs. By modulating both the in-phase (I) and quadrature (Q) compo- nents of the carrier, QPSK achieves higher spectral efficiency compared to BPSK.

3.1.3 Signal generation:

In this project, the BPSK and QPSK signals are generated externally using Python signal processing libraries such as NumPy and SciPy. The generated continuous- time signals are sampled and stored as discrete amplitude values in text (.txt) files. These files are imported into the Verilog HDL testbench in Vivado, where they serve as the system's input. This separation between signal generation (software domain) and signal classification (hardware domain) ensures that the Verilog mod- ules are tested with realistic input data that closely resemble real communication signals

3.2 Feature Extraction

Feature extraction is a critical stage in the modulation classification process, re- sponsible for converting raw signal samples into measurable and distinctive pa- rameters that can be used to identify the modulation type. In other words, instead of analyzing the entire waveform directly, specific numerical features are derived that capture the signal's underlying behavior — such as its amplitude variations, phase transitions, or frequency changes.
The main features used in this design are:

1. **Zero Crossing Rate (ZCR):** Represents how often the signal crosses the zero-amplitude axis, which reflects phase change frequency. BPSK, having fewer phase shifts per symbol, exhibits a lower ZCR compared to QPSK.

2. **Average Amplitude:** Captures the overall magnitude variation of the signal samples.

3. **Signal Energy:** Represents the cumulative power of the signal within an observation window. QPSK typically shows higher energy fluctuation due to its multi-phase transitions.

1. **Zero Crossing Rate (ZCR):** The Zero Crossing Rate measures the number of times the signal waveform crosses the zero amplitude axis within a de- fined observation window. It captures the frequency of phase changes in the signal. BPSK has only two phase states per symbol (0° and 180°), result- ing in fewer zero crossings, whereas QPSK has four phase states (0°, 90°, 180°, 270°), producing more zero crossings. This makes ZCR a simple yet effective feature for initial modulation discrimination.

The ZCR for a discrete-time signal $x[n]$ of length N is mathematically de- fined as:

$$\mathrm{ZCR} = \frac{1}{2(N-1)} \sum_{n=1}^{N-1} \left|\mathrm{sign}(x[n]) - \mathrm{sign}(x[n-1])\right| \qquad (3.3)$$

where

$$\mathrm{sign}(x[n]) = \begin{cases} +1 & x[n] > 0 \\ -1 & x[n] < 0 \end{cases}.$$

2. **Average Amplitude:** Average amplitude represents the mean magnitude of the signal samples within an observation window. It captures overall signal strength and helps distinguish modulation types based on amplitude variations.

The mathematical expression for average amplitude is:

$$\text{Average Amplitude} = \frac{1}{N}\sum_{n=0}^{N-1} |x[n]| \qquad (3.4)$$

3. **Signal Energy:** Signal energy quantifies the total power contained in the signal over the observation window. BPSK signals, having fewer phase transitions, generally exhibit lower energy fluctuation than QPSK signals, which makes energy a robust feature for modulation classification, even in noisy environments.
Signal energy is defined as:

$$E = \sum_{n=0}^{N-1} |x[n]|^2 \qquad (3.5)$$

These features are computed in the feature extractor module of the Verilog design. The module outputs the calculated feature values along with a feature valid flag to indicate when the data is ready for classification.

3.3 Signal Classification

The classification block is responsible for taking the numerical features extracted from the input signal and determining the modulation type. In this work, a **Threshold- Based Finite State Machine (FSM) Classifier** is implemented in Verilog, which provides a simple, deterministic, and hardware-friendly solution for distinguish- ing between BPSK and QPSK signals.

Threshold-Based Classifier

The classifier receives three numerical features from the featureextractor module:

- Zero Crossing Rate (ZCR)
- Average Amplitude
- Signal Energy

The FSM compares these features against predefined thresholds to assign a modulation label. This approach avoids complex arithmetic operations such as multiplication or division, making it suitable for resource-constrained FPGA plat- forms.
Classification Steps:

1. Read the feature values (ZCR, Average Amplitude, Signal Energy) when the feature valid flag is asserted.

2. Compare each feature with its corresponding threshold:

- If ZCR < ZCR_{th}, classify as BPSK; else QPSK.
- If Average Amplitude > Amp_{th}, it supports the decision.
- If Signal Energy > $Energy_{th}$, it supports the decision.

Threshold Selection:

For each feature, a threshold value is chosen to best separate the two mod- ulation classes (BPSK and QPSK). The thresholds are empirically deter- mined from representative signal samples.

- Zero Crossing Rate threshold (ZCR_{th}): chosen as the midpoint be- tween the average ZCR values of BPSK and QPSK signals.
- Amplitude threshold (Amp_{th}): chosen as the midpoint between the mean amplitudes of BPSK and QPSK signals.
- Energy threshold ($Energy_{th}$): chosen as the midpoint between the mean energies of BPSK and QPSK signals.

These thresholds ensure minimal overlap between the feature distributions of the two modulation types, thereby maximizing classification accuracy.

3. Assert resultvalidflag along with the binary output:

- 2’b00→ BPSK
- 2’b01→ QPSK

Finite State Machine (FSM) Design

The FSM consists of the following states:

- **IDLE:** Waits for the feature extraction to complete.
- **FEATURE READ:** Reads and latches feature values.
- **CLASSIFY:** Compares features against thresholds and determines modula- tion type.
- **OUTPUT:** Asserts result and resets the FSM to IDLE for the next observa- tion window.

The classifier is integrated with the feature extraction module in the topmodule, where the data flow is controlled by a central control unit to ensure proper se- quencing of operations.

3.4 Top level Integration

The Top Module combines all submodules (Feature Extractor, Classifier, and Con- trol Unit) into a single, functional design. It defines clock, reset, and signal inter- faces for simulation. Each module communicates through valid and ready signals, promoting modular verification.
Input waveform samples (BPSK and QPSK) are loaded from text files gener- ated using Python. These files emulate real-world signals and provide a realistic test environment for the Verilog design.

4. Results and Discussions

4.1 Area and Power Utilization Analysis

Efficient use of FPGA resources and power is a critical metric in evaluating the performance of any hardware-based design. The proposed Verilog HDL-based Automatic Modulation Classification (AMC) system was implemented and syn- thesized using the Xilinx Vivado toolchain. The synthesis and implementation reports provide detailed information regarding the utilization of logic resources such as Look-Up Tables (LUTs), Flip-Flops (FFs), DSP slices, and on-chip mem- ory blocks (BRAM).

4.1.1 Area Utilization

The proposed design demonstrates very low area utilization, owing to its simpli- fied threshold-based feature extraction and classification mechanism. Since no complex arithmetic operations (like multipliers or divisions) are employed, DSP usage remains negligible. The modular Verilog structure enables compact syn- thesis and optimal mapping onto FPGA resources. These results confirm that the AMC system occupies less than 2leaving significant space for scalability—such as adding higher-order modulation schemes (8-PSK, QAM) or multiple input chan- nels in future extensions.

Resource Type	Available on FPGA	Used by Design	Utilization (%)
Look-Up Tables (LUTs)	53200	1045	1.96%
Flip-Flops (FFs)	106400	980	0.92%
DSP Slices	220	0	0%
Block RAM (BRAM)	120	2	1.6%

Figure 4.1: Area Utilisation

4.1.2 Power Utilization

Power estimation was carried out using the Vivado Power Analyzer. The total power consumption is primarily due to dynamic switching activity in the feature extraction module and clock distribution network. Since the design uses simple comparison and counting logic, overall power remains minimal.

Power Component	Estimated Power (mW)	Percentage Contribution
Dynamic Power	18 mW	72%
Static (Leakage)	7 mW	28%
Total Power	25 mW	

Figure 4.2: Power Utilisation

4.2 Classification Performance

Noise-free conditions: BPSK classified as 2’b00 and QPSK as 2’b01 with 100Mod- erate noise (SNR = 15 dB): Classification accuracy remained above 95due to am- plitude fluctuations near threshold boundaries. Threshold-based logic ensured de- terministic, real-time decision-making.

4.3 Feature Extraction Accuracy

The extracted features for both BPSK and QPSK closely matched theoretical ex- pectations. Table summarizes the average feature values observed during sim- ulation (over 1,000 samples per signal). These values confirm that ZCR serves

Feature	BPSK (mean ± std)	QPSK (mean ± std)	Observed Difference
Zero Crossing Rate (ZCR)	0.18 ± 0.02	0.35 ± 0.03	~48% higher in QPSK
Average Amplitude	0.98 ± 0.01	0.95 ± 0.02	Slightly lower in QPSK due to quadrature mixing
Signal Energy	0.96 ± 0.02	1.10 ± 0.03	~15% higher in QPSK

Figure 4.3: Feature Extraction

as a strong discriminator between the two modulations, while energy variations provide supporting evidence for classification. The feature extraction module achieved >99% match with Python-generated reference data, validating the cor- rectness of the Verilog implementation

5. Conclusions and Future Scope for Improvement

The present work establishes a lightweight FPGA-based automatic modulation classification (AMC) system using Verilog HDL with zero-crossing, amplitude, and energy features. Future developments can focus on the following areas:

1. **Extension to Higher-Order Modulations:** Expanding the classifier to sup- port additional modulation schemes such as 8-PSK, 16-QAM, and OFDM to enhance versatility in modern communication systems.

2. **Real-Time Hardware Implementation:** Deploying the design on a physi- cal FPGA development board and interfacing it with real RF signal sources to evaluate performance under practical channel conditions.

3. **Optimization for Power and Resource Efficiency:** Refining the design architecture through pipelining, clock gating, or feature reduction to further minimize area utilization and power consumption while maintaining classi- fication accuracy.

Bibliography

1. M. Abdel-Meniem et al., "FPGA Implementation of Automatic Modulation Recognition System for Advanced SATCOM System," 2020. Focuses on employing Wavelet Transform (Discrete Haar WT) to differentiate modu- lation schemes (PSK, FSK, QAM) using Xilinx Virtex-4 FPGA. Good for showing feature-driven hardware implementation.
2. A. R. Castro et al., "Modulation Classification in Cognitive Radio," Cog- nitive Radio. Covers system architecture using cyclostationary features and SVM-based classifiers in FPGA. Excellent for understanding classification workflow and hardware feasibility.
3. H. Kumar S. N. et al., "FPGA Implementation Of Digital Modulation Schemes Using Verilog HDL," 2021. Verilog implementations of modula- tion schemes (BASK, BFSK, BPSK, QPSK) using simulation on ModelSim and Xilinx-ISE. Great reference for Verilog-based design approach.
4. N. Tanawade and S. Sudhansu, "FPGA Implementation of Digital Modu- lation Techniques BPSK and QPSK using HDL Verilog," 2017. Highlights basic Verilog HDL-based modulation implementations—helpful for under- standing signal generation and handling in Verilog.
5. S. Tridgell et al., "Real-time Automatic Modulation Classification using RFSoC," 2022. Demonstrates real-time modulation classification on Xilinx RFSoC platform using deep learning techniques. Highlights FPGA perfor- mance aspects like throughput and latency.
6. Z. L. Tang et al., "Implementation of Deep Learning-Based Automatic Mod- ulation Classification Methods and Its Implementation Based on FPGA," 2018. Shows how convolutional autoencoder-based classification can be im- plemented on FPGA, including confusion matrix performance across SNRs.
7. B. Gavin et al., "Low-Area and Low-Power FPGA Implementation of a DBSCAN-Based RF Modulation Classifier," 2024. Proposes a pipelined classifier using DBSCAN algorithm, optimized for low latency and resource usage, achieving accuracy comparable to CNNs at high SNR.
8. W. Xiao, "A Review of Research on Signal Modulation Recognition Based on Deep Learning," MDPI Electronics, 2022. A comprehensive survey on how deep learning methods have advanced modulation recognition, includ- ing challenges and future directions. Helps contextualize your hardware- focus