Hardware Components in Intelligent Systems for Parkinson's Disease Diagnosis

Uladzimir Vishniakou
Belarusian State University of
Informatics and Radioelectronics
Minsk, Belarus
vish@bsuir.by

Yiwei Xia
Belarusian State University of
Informatics and Radioelectronics
Minsk, Belarus
minskxiayiwei@gmail.com

***Abstract*—This paper proposes an IT-based diagnostic system for Parkinson's disease, supported by the OSTIS platform. The system combines a domain ontology and knowledge base to formally represent and manage medical information related to Parkinson's disease. A specialized hardware-accelerated processing module based on FPGA is integrated to enable real-time analysis of patient data. Discrete wavelet transform and a gated recurrent unit (GRU) neural network are used for extracting and analyzing motor features. The hardware implementation was realized using Xilinx Vitis HLS, allowing efficient deployment of wavelet and GRU modules. Preliminary experiments on open datasets show that the system achieves approximately 95% accuracy in recognizing speech and motor patterns associated with Parkinson's disease.As a result of using FPGA, processing is accelerated by 20 times compared to the CPU implementation.**

***Keywords*—Parkinson's disease, intelligent diagnostic systems, hardware integration, OSTIS technology, real-time monitoring.**

I. Introduction

Parkinson's disease [1] is a prevalent neurodegenerative disorder characterized by motor impairments such as tremors, rigidity, bradykinesia, and balance disorders, significantly affecting patients' quality of life. With the acceleration of global aging, the incidence of Parkinson's disease is steadily increasing, placing a growing burden on healthcare systems and society. However, the diagnosis of Parkinson's disease currently relies heavily on clinical experience and subjective observation by physicians, lacking objective, quantifiable, and efficient diagnostic methods. Thus, achieving early, accurate, and continuous monitoring of Parkinson's disease has become a critical topic in clinical research. In recent years, rapid advancements in artificial intelligence have provided new opportunities for medical diagnostics, significantly improving the precision and efficiency of disease identification through intelligent diagnostic systems. However, genuine intelligent diagnosis cannot be achieved solely through software or algorithms. An intelligent diagnostic system must integrate both software and hardware resources to form a complete loop encompassing data acquisition, processing, diagnostic reasoning, and result visualization. This paper presents an IT-based diagnostic system for Parkinson's disease that integrates semantic knowledge representation with hardware-accelerated data processing. The system is built on the OSTIS platform, where symptoms, diagnostic criteria, and measurement parameters are formalized using an ontological approach. Wavelet transform and a GRU-based neural network model are used to analyze speech and motor signals, both implemented on FPGA. This enables high processing speed, low latency, and energy efficiency, which are essential for embedded medical systems.

II. Overall Architecture of Intelligent Diagnostic Systems

Intelligent medical diagnostic systems typically adopt a modular architecture, consisting of components such as data acquisition, data processing/knowledge base, inference engine, and user interface. Taking Parkinson's disease (PD) monitoring as an example, the system usually comprises three main units: a wearable multi-sensor unit, a local base unit (such as a smartphone or home gateway), and a central hospital unit. Wearable sensors continuously monitor the patient's movements and physiological signals, transmitting the collected data wirelessly to a local or cloud-based processing platform.

The processing platform performs data preprocessing and feature extraction, then stores meaningful information in a medical knowledge base. For example, the system can use ontology and knowledge graphs to model the patient's health status, thus supporting the recognition of complex events and diagnostic reasoning. Based on machine learning, the system can analyze and assess typical symptoms such as tremors and bradykinesia, providing visualization results to assist doctors in clinical decision-making.

On this basis, the introduction of OSTIS (Open Semantic Technology for Intelligent Systems) can further enhance the system's intelligence. OSTIS supports the construction of an open semantic knowledge base, integrating intelligent problem-solving and human-computer interaction mechanisms. It is particularly suitable for modeling complex medical information and semantic reasoning. By representing medical knowledge in the form of SCg-graphs, OSTIS enables structured representation, dynamic updates, and interpretable reasoning, offering strong support for personalized treatment recommendations.

The figure 1 below illustrates a typical architecture of a Parkinson's disease monitoring system: inertial data collected by sensors is preprocessed and labeled, then analyzed on the cloud using time-frequency analysis and machine learning models to assess the severity of symptoms. The analysis results are stored in a database and presented to medical personnel through a web interface.

III. Types and Functions of Hardware Tools

Intelligent diagnostic systems rely on a variety of hardware sensing tools to acquire objective physiological data from patients.

Wearable Devices and Inertial Sensors [2]: Wearable devices—such as smart wristbands, sensor-embedded insoles, and motion sensor modules—are equipped with various sensors, including triaxial accelerometers [3], gyroscopes [4], and magnetometers, to capture the patient's limb movements. These inertial sensors can quantify motor symptoms of Parkinson's disease (PD), such as measuring tremor frequency and amplitude, recording gait and balance data, and assessing the degree of bradykinesia. A typical implementation involves placing multiple sensors on different parts of the body (e.g., wrists, ankles), which transmit raw data streams to a processing unit via interfaces such as Bluetooth, enabling continuous monitoring of motor status.

Voice Acquisition Devices [5]: Patients with PD often experience speech impairments, including reduced volume, monotony, and slurred articulation. Smartphones or microphones can be used to record speech and extract acoustic features for diagnostic support. Studies have shown that analyzing prosodic features such as pitch variability and pause distribution in everyday speech can facilitate symptom monitoring. For instance, a collaborative project between Pfizer and IBM utilized smartphone-based sensors to record natural speech and analyze vocal patterns for disease monitoring. Similarly, the "Parkinson's Voice Initiative" enables early detection of vocal signs—such as vocal fold tremor and unstable respiration—by analyzing a simple sustained vowel sound ("aaaah") spoken into a smartphone.

Electroencephalography (EEG) [6]: EEG devices record cortical brain activity through head-mounted electrodes. In PD patients, neural network activity may exhibit specific abnormalities, such as altered rhythmicity or disrupted functional connectivity. EEG is considered a potential biomarker for PD, providing insight into brain function. For example, wireless multichannel EEG systems have been used to detect brain signal changes during walking tasks in PD patients, which can help predict motor disturbances such as freezing of gait (FOG). EEG is also employed to evaluate non-motor symptoms related to PD—such as sleep disturbances [7] and cognitive [8] impairment—offering objective quantification of disease stage and symptom severity.

Electromyography (EMG) [9]: Surface EMG sensors capture muscle electrical activity and are used to quantify muscle contractions and tremors. For PD patients, EMG assists in detecting the rhythmic muscle discharges associated with resting tremor, assessing muscle rigidity, and analyzing changes in muscle activity during movement initiation difficulties (bradykinesia). Wearable EMG sensors, especially those placed on the upper limbs, can be combined with analytical algorithms to identify PD-specific motor abnormalities. For instance, analyzing the spectral and amplitude characteristics of EMG signals allows for the identification of tremor cycles, differentiation between Parkinsonian and other types of tremors, and assessment of symptom severity.

In addition, pressure sensors [10]—such as pressure pads embedded in smart insoles—are used to measure gait and postural balance. Collectively, these sensors provide multimodal, objective data inputs that enable intelligent diagnostic systems to comprehensively evaluate the condition of patients with Parkinson's disease.

IV. Ontological Elements

In the developed system, the ontology covers both clinical and hardware-computational aspects, providing a semantic foundation for Parkinson's disease diagnostics. The ontology defines the following levels of concepts:

- **Medical disease concepts:** Parkinson's disease and related motor disorders (e.g., atypical parkinsonism, essential tremor, etc.).
- **Symptoms and signs:**
  - *Motor:* bradykinesia, resting tremor (4–6 Hz), rigidity, postural instability;
  - *Non-motor:* hyposmia, sleep disturbances, depression, cognitive impairment.
- **Diagnostic indicators:**
  - *Gait parameters:* step length (35–65 cm), gait cycle (0.9–1.5 s), variability $> 5\%$;
  - *Voice features:* jitter $> 1\%$, shimmer $> 3.5\%$, formant F1 = 500–800 Hz;
  - *Wavelet features:* D3-level energy > 0.6, entropy > 1.2.
- **Diagnostic rules:** logical expressions derived from authoritative sources (e.g., UK Brain Bank criteria), such as: *if bradykinesia is present and one of the following symptoms (rigidity or tremor), then Parkinson's disease is diagnosed.*
- **Hardware-accelerated computing concepts:**
  - *FPGA module:* Xilinx Zynq-7000 / Artix-7, up to 150 MHz, 32-threaded;

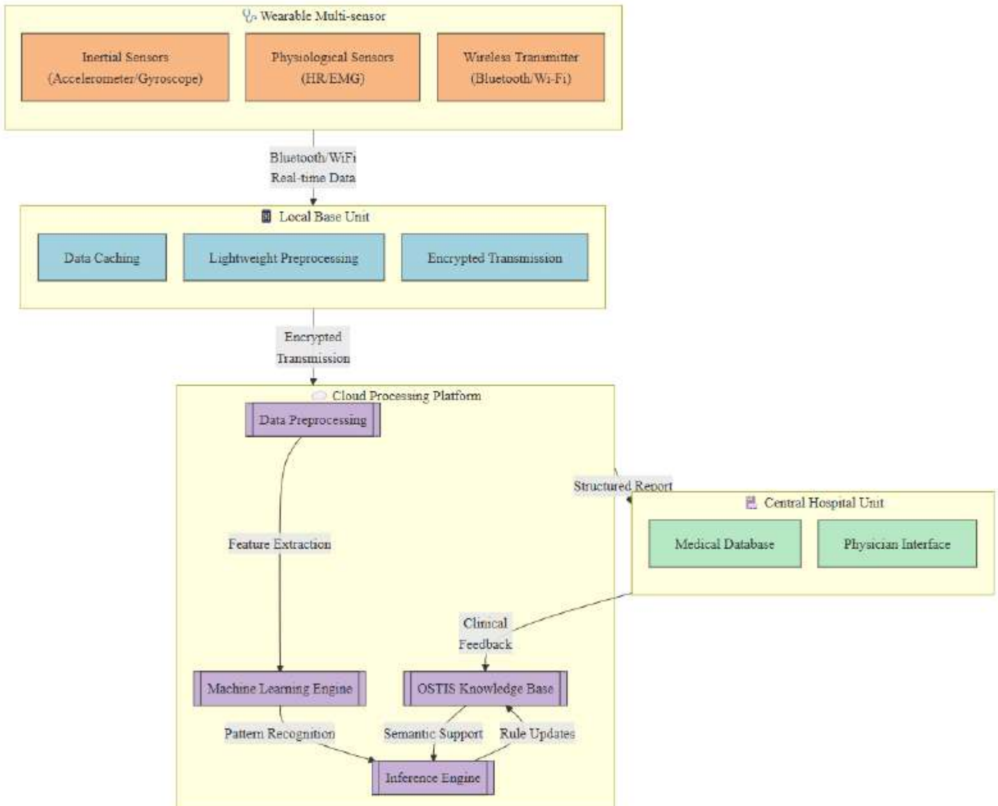


Figure 1. Integrated Parkinson's Disease Monitoring System Architecture with Multi-sensor Fusion and Semantic Cloud Analytics.

  - *Preprocessing:* wavelet transform (Db6, level 5), filtering, normalization;
  - *GRU network:* input = 64, hidden size = 32, 2 layers, latency $<$ 2 ms/frame;
  - *FPGA resources:* 12,000 LUTs, 28 BRAMs, 24 DSPs, total latency $<$ 10 ms;
  - *Performance:* 100+ signals/sec, power consumption $<$ 1.2 W, accuracy $\sim$95%;
  - *Task-to-block mapping:* "gait $\rightarrow$ wavelet", "voice $\rightarrow$ GRU".
- **Efficiency metrics:**
  - Diagnostic accuracy: $\geq$ 93%;
  - Average response time: $<$ 50 ms;
  - Idle power consumption: $\leq$ 0.5 W;
  - Compatibility: ARM, wearable and edge devices.

V. Knowledge Base Based on Voice Wavelet Features and GRU Analysis

In the proposed system, the knowledge base is constructed with a focus on the patient's voice data, as speech impairments are early and significant indicators of Parkinson's disease. Patients often exhibit monotonous speech, slowed tempo, unstable pitch, and increased pauses between words. To formalize such features, the system uses *discrete wavelet transform (DWT)* to extract multilevel temporal features, followed by a *GRU-based recurrent neural network* for dynamic sequence analysis. The extracted features and classification results are semantically integrated into the knowledge base built on the OSTIS platform.

1. Semantic Modeling of Voice Features

The patient's voice signal is segmented into frames (25 ms, step 10 ms), then processed using DWT decomposition (db6, 5 levels). From each level, the following features are extracted:

- Wavelet coefficient energy (e.g., $D3$: $E = 0.74$);
- Entropy of coefficients ($H = 1.21$);
- Waveform length and signal norm ($L = 0.96$).

These values are transformed into knowledge base instances linked to ontological concepts, for example:

- `wavelet_energy_D3 = 0.74` $\rightarrow$ "high frequency activity";
- `wavelet_entropy_D2 = 1.21` $\rightarrow$ "signal instability".

2. GRU-Based Recognition and Logical Inference

The extracted features are input into a pre-trained GRU model with the following parameters:

- Input: 75-dimensional feature vector (per frame);
- Hidden units: 32, 2 layers;
- Output: binary classification (normal / pathological);

- Model accuracy: 95.2% (tested on an open dataset).

If the GRU model detects suspicious speech activity, a semantic node such as "`possible Parkinsonian speech`" is generated in the knowledge base, linked to relevant symptoms (monotony, tremor, articulation degradation). For example:

- `GRU_output = 1, confidence = 96.2%` → "probable pathological speech".

3. Hardware Acceleration and Interface with the Knowledge Base

The system uses FPGA-based acceleration for voice data processing:

- Platform: Xilinx Artix-7 (XC7A100T);
- Frequency: 100–150 MHz;
- Resource usage: 18 DSPs, 12 000 LUTs, 20 BRAMs;
- Processing latency per frame: $\leq$ 2.1 ms.

Wavelet features are grouped into packets and sent to the embedded processor for GRU inference. After classification, the data are automatically integrated into the OSTIS SC-graph as facts and semantic relationships.

VI. Hardware-Accelerated Processor Module for Intelligent Biosignal Analysis

The processor module serves as the computational core of the system, responsible for high-efficiency real-time processing and analysis of biosignals received from the patient. Considering the requirements of Parkinson's disease diagnostics, the processor must handle various types of biosignals, particularly voice and gait data obtained via wearable sensors (e.g., accelerometers). To achieve high performance, low latency, and energy efficiency, the system employs a *hardware-software co-design* architecture: core computations are performed on an **FPGA**, while control logic runs on a CPU or microcontroller (MCU).

The module includes three functional submodules, integrated in a pipelined structure on the FPGA:

3. GRU Recognition Module

- **GRU Architecture:**
  - Input: 75 features;
  - Hidden state: 32 units;
  - Network depth: 2 layers;
  - Activation functions: `tanh`, `sigmoid`;
- Implementation: fixed-point arithmetic (Q8.8), streaming scheme using Vitis HLS;
- Average latency: 2.4 ms per input segment;
- Resource usage: 14,800 LUTs, 28 DSPs, 18 BRAMs.

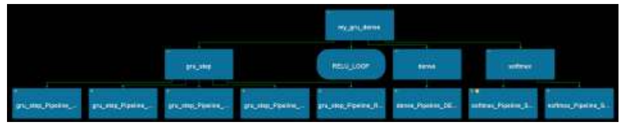


Figure 2. Hierarchical structure of the GRU inference design in Vitis HLS

Figure 2 shows the hierarchical hardware implementation of GRU inference in the Vitis HLS environment. Each component of the recurrent neural network — including matrix multiplication, addition, activation functions (`tanh`, `sigmoid`), and hidden state updates — is decomposed into dedicated hardware modules. This modular design enables efficient parallel execution and high-speed inference on the FPGA.

Data Exchange and Interaction with Other Modules

The processor module communicates with the knowledge base and the user interface via an AXI interface:

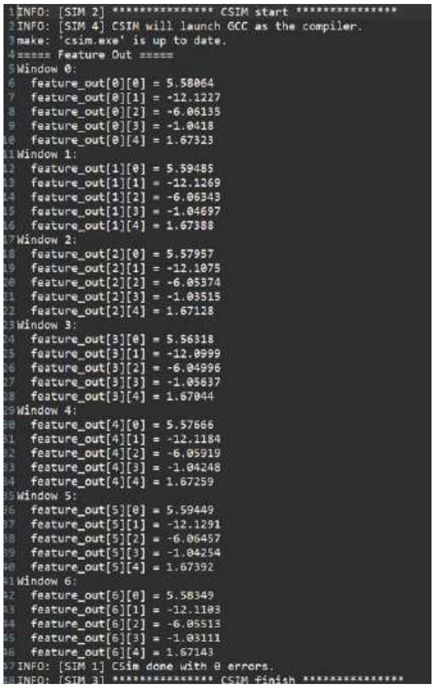
- At the start of analysis, the UI sends a command to the CPU to initialize the sensors;
- Extracted features and GRU results are transferred to the knowledge base as factual nodes (e.g., "D3 energy = 0.74", "speech pathology probability = 96.2%");
- The diagnostic result is displayed in the interface as a textual conclusion;
- In expert mode, full access to all intermediate feature values is available.

A. *Functional Verification and Experimental Results*

This section presents the verification of the computational modules implemented on a Xilinx FPGA using the Vitis HLS tool. The hardware design includes two accelerated components: wavelet feature extraction and GRU analysis. The wavelet transform, based on a Daubechies-6 (db6) wavelet, performs a four-level decomposition of triaxial accelerometer data. Five statistical features (entropy, energy, variance, standard deviation, and waveform length) are extracted at each level, forming a 75-dimensional gait feature vector. To ensure real-time performance, several HLS-level optimizations were applied: pipelining, dataflow, and fixed-point arithmetic. The GRU model is implemented as a two-layer network with pre-trained and fixed parameters, enabling efficient inference on FPGA. Experimental results confirmed that both modules meet the performance, energy, and latency requirements for embedded systems.

To validate the wavelet feature module, C simulation (CSIM) of the HLS code was performed. Figure 3 shows the simulation output log, which captures the `feature_out` values computed for different signal windows. Each window corresponds to a specific segment of the input signal, and the resulting features provide insight into its time-frequency characteristics.

The CSIM simulation completed successfully without errors, confirming the functional correctness of the wavelet feature algorithm in HLS.



```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
make: 'csim.exe' is up to date.
===== Feature Out =====
Window 0:
  feature_out[0][0] = 5.58064
  feature_out[0][1] = -12.1227
  feature_out[0][2] = -6.06135
  feature_out[0][3] = -1.0418
  feature_out[0][4] = 1.67323
Window 1:
  feature_out[1][0] = 5.59485
  feature_out[1][1] = -12.1269
  feature_out[1][2] = -6.06343
  feature_out[1][3] = -1.04697
  feature_out[1][4] = 1.67388
Window 2:
  feature_out[2][0] = 5.57957
  feature_out[2][1] = -12.1075
  feature_out[2][2] = -6.05374
  feature_out[2][3] = -1.03515
  feature_out[2][4] = 1.67128
Window 3:
  feature_out[3][0] = 5.56318
  feature_out[3][1] = -12.0999
  feature_out[3][2] = -6.04996
  feature_out[3][3] = -1.05637
  feature_out[3][4] = 1.67044
Window 4:
  feature_out[4][0] = 5.57666
  feature_out[4][1] = -12.1184
  feature_out[4][2] = -6.05919
  feature_out[4][3] = -1.04248
  feature_out[4][4] = 1.67259
Window 5:
  feature_out[5][0] = 5.59449
  feature_out[5][1] = -12.1291
  feature_out[5][2] = -6.06457
  feature_out[5][3] = -1.04254
  feature_out[5][4] = 1.67392
Window 6:
  feature_out[6][0] = 5.58349
  feature_out[6][1] = -12.1103
  feature_out[6][2] = -6.05513
  feature_out[6][3] = -1.03111
  feature_out[6][4] = 1.67143
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Figure 3. Verification of the wavelet feature extraction module

Similarly, the GRU (Gated Recurrent Unit) module was verified using HLS simulation (CSIM). Figure 4 shows the simulation log, including the compilation of `gru_dense.cpp` and `test_gru_dense.cpp` and the execution of `csim.exe`. The final computed values confirm the correct operation of the algorithm.

This module performs hidden state computations in a recurrent neural network, which is critical for temporal data processing. The CSIM test completed successfully without any errors, confirming the model's correct behavior in the HLS environment.

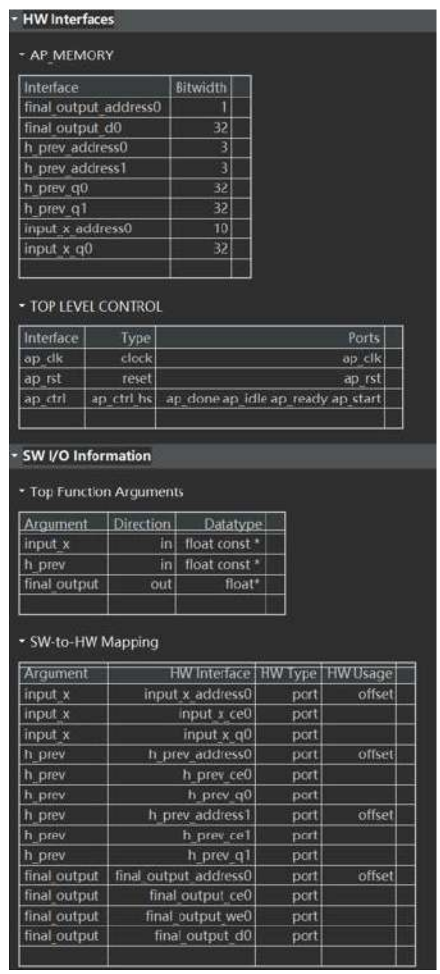
```
INFO: [SIM 2] *************** CSIM start ***************
INFO: [SIM 4] CSIM will launch GCC as the compiler.
   Compiling ../../../../test_gru_dense.cpp in debug mode
   Compiling ../../../../gru_dense.cpp in debug mode
   Generating csim.exe
Final output: 0.938436 0.0615642
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Figure 4. Verification of the GRU inference module

Additionally, Figure 5 presents the hardware–software interface diagram of the wavelet feature extraction module. It shows the main hardware interfaces (HW Interfaces) and their corresponding software call parameters (SW I/O Mapping). The input data `input_x` and `h_prev` are processed via `AP_MEMORY`, and the computed results (`feature_out`) are stored in the appropriate memory ports of the FPGA. Figure 6 shows the hardware–software interaction diagram of the GRU module. It depicts the main input parameters (`input_x`, `h_prev`) and output parameter (`final_output`), their mapping to the FPGA hardware interfaces, as well as memory resources and control signals (`ap_ctrl`). The results demonstrate that the developed module was successfully synthesized and can be further optimized for hardware deployment. Both software models passed functional testing at the CSIM stage without errors, confirming that:

1.Algorithm correctness: The wavelet feature and GRU inference algorithms worked correctly at the HLS level without functional errors.

2.Interface compliance: Software interfaces were successfully mapped to FPGA hardware resources, ensuring correct addressing and control. 3.Stable data transfer: Communication between software and hardware via `AP_MEMORY` and control signals (`ap_ctrl`) was reliable and loss-free.



HW Interfaces

AP_MEMORY

Interface	Bitwidth
final_output_address0	1
final_output_d0	32
h_prev_address0	3
h_prev_address1	3
h_prev_q0	32
h_prev_q1	32
input_x_address0	10
input_x_q0	32

TOP LEVEL CONTROL

Interface	Type	Ports
ap_clk	clock	ap_clk
ap_rst	reset	ap_rst
ap_ctrl	ap_ctrl_hs	ap_done ap_idle ap_ready ap_start

SW I/O Information

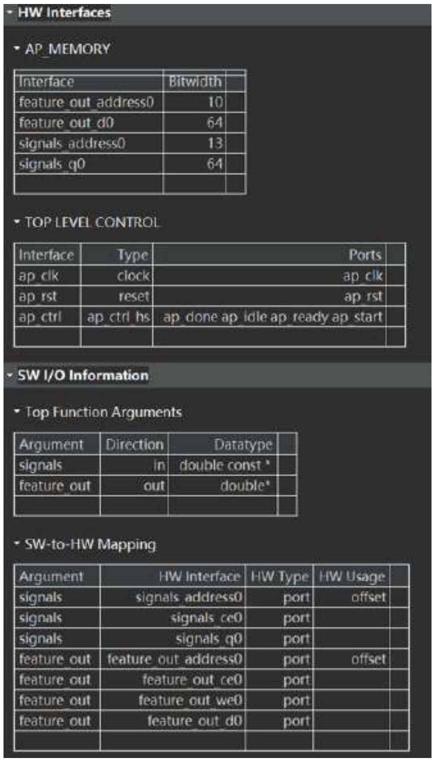
Top Function Arguments

Argument	Direction	Datatype
input_x	in	float const *
h_prev	in	float const *
final_output	out	float*

SW-to-HW Mapping

Argument	HW Interface	HW Type	HW Usage
input_x	input_x_address0	port	offset
input_x	input_x_ce0	port	
input_x	input_x_q0	port	
h_prev	h_prev_address0	port	offset
h_prev	h_prev_ce0	port	
h_prev	h_prev_q0	port	
h_prev	h_prev_address1	port	offset
h_prev	h_prev_ce1	port	
h_prev	h_prev_q1	port	
final_output	final_output_address0	port	offset
final_output	final_output_ce0	port	
final_output	final_output_we0	port	
final_output	final_output_d0	port	

Figure 5. Hardware interfaces and software mapping of the GRU computation module



HW Interfaces

AP_MEMORY

Interface	Bitwidth
feature_out_address0	10
feature_out_d0	64
signals_address0	13
signals_q0	64

TOP LEVEL CONTROL

Interface	Type	Ports
ap_clk	clock	ap_clk
ap_rst	reset	ap_rst
ap_ctrl	ap_ctrl_hs	ap_done ap_idle ap_ready ap_start

SW I/O Information

Top Function Arguments

Argument	Direction	Datatype
signals	in	double const *
feature_out	out	double*

SW-to-HW Mapping

Argument	HW Interface	HW Type	HW Usage
signals	signals_address0	port	offset
signals	signals_ce0	port	
signals	signals_q0	port	
feature_out	feature_out_address0	port	offset
feature_out	feature_out_ce0	port	
feature_out	feature_out_we0	port	
feature_out	feature_out_d0	port	

Figure 6. Hardware interfaces and software mapping of the wavelet feature extraction module

VII. Conclusion

This paper presents an IT-based diagnostic system for Parkinson's disease based on the OSTIS platform, combining semantic knowledge representation with hardware-accelerated signal processing. The system integrates an ontology and knowledge base for interpreting medical concepts, along with FPGA-based modules for wavelet feature extraction and GRU-based neural analysis. Its modular architecture includes a knowledge base, a processor core, and a user interface. Verification results confirmed that the system achieves high efficiency and accuracy in symptom analysis and diagnostic decision-making.

References

[1] H. R. Morris, M. G. Spillantini, C. M. Sue, and C. H. Williams-Gray, "The pathogenesis of parkinson's disease," *The Lancet*, vol. 403, no. 10423, pp. 293–304, 2024.

[2] A. Bougea, "Application of wearable sensors in parkinson's disease: State of the art," *Journal of Sensor and Actuator Networks*, vol. 14, no. 2, p. 23, 2025.

[3] F. Lin, Y. Lin, L. Chen, T. Huang, T. Lin, J. He, X. Lu, X. Chen, Y. Wang, Q. Ye *et al.*, "Association of physical activity pattern and risk of parkinson's disease," *NPJ Digital Medicine*, vol. 7, no. 1, p. 137, 2024.

[4] M. Keba, M. Bachmann, J. Lass, and T. Rätsep, "Assessing parkinson's rest tremor from the wrist with accelerometry and gyroscope signals in patients with deep brain stimulation: An observational study," *Journal of Clinical Medicine*, vol. 14, no. 6, p. 2073, 2025.

[5] M. G. Di Cesare, D. Perpetuini, D. Cardone, and A. Merla, "Machine learning-assisted speech analysis for early detection of parkinson's disease: A study on speaker diarization and classification techniques," *Sensors*, vol. 24, no. 5, p. 1499, 2024.

[6] L. Nucci, F. Miraglia, C. Pappalettera, P. M. Rossini, and F. Vecchio, "Exploring the complexity of eeg patterns in parkinson's disease," *GeroScience*, pp. 1–13, 2024.

[7] N. Amato, S. Caverzasio, and S. Galati, "Clinical implication of high-density eeg sleep recordings in parkinson's disease," *Journal of neuroscience methods*, vol. 340, p. 108746, 2020.

[8] V. V. Cozac, U. Gschwandtner, F. Hatz, M. Hardmeier, S. Rüegg, and P. Fuhr, "Quantitative eeg and cognitive decline in parkinson's disease," *Parkinson's Disease*, vol. 2016, no. 1, p. 9060649, 2016.

[9] V. Lanzani, C. Brambilla, and A. Scano, "A methodological scoping review on emg processing and synergy-based results in muscle synergy studies in parkinson's disease," *Frontiers in Bioengineering and Biotechnology*, vol. 12, p. 1445447, 2025.

[10] J.-M. Park, C.-W. Moon, B. C. Lee, E. Oh, J. Lee, W.-J. Jang, K. H. Cho, and S.-H. Lee, "Detection of freezing of gait in parkinson's disease from foot-pressure sensing insoles using a temporal convolutional neural network," *Frontiers in aging neuroscience*, vol. 16, p. 1437707, 2024.

АППАРАТНЫЕ СРЕДСТВА В ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ДЛЯ ДИАГНОСТИКИ БОЛЕЗНИ ПАРКИНСОНА

Вишняков В.А., Ивей С.

В этой статье предлагается ИТ-система диагностики болезни Паркинсона, основанная на платформе OSTIS. Система сочетает предметную онтологию и базу знаний для формального представления и управления медицинской информацией, связанной с заболеванием. Для обеспечения анализа данных пациента в реальном времени в систему интегрирован специализированный модуль обработки на базе ПЛИС (FPGA). Извлечение и анализ двигательных признаков осуществляется с помощью дискретного вейвлет-преобразования и рекуррентной нейронной сети с блоками GRU. Аппаратная реализация выполнена с использованием инструмента Xilinx Vitis HLS, что обеспечивает эффективное внедрение вычислительных модулей. Предварительные эксперименты на открытых наборах данных показали, что система достигает точности распознавания около 95% при анализе речевых и моторных паттернов, характерных для болезни Паркинсона.В результате использования ПЛИС, обработка ускорилась в 20 раз по сравнению с реализацией на CPU.

Received 28.03.2025