Journal of Computer Applications and Information Technology
Vol. 2, Issue. 3, Sep 2026, pp. 1~11
ISSN: 3107-6424, DOI: 10.32595/jcait/v2i3.2026.37

FPGA-Prototyped Application-Specific RTL Architecture for Real-Time Sign Language Translation Using Dual-Glove Sensor Fusion

Heng-Wei Lee[1], Racheal Poh Su Ying[2]

[1]School of Digital Technology, Wawasan Open University (WOU), George Town 10050, Penang, Malaysia. E-mail: hwlee@wou.edu.my
[2]School of Digital Technology, Wawasan Open University (WOU), George Town 10050, Penang, Malaysia. E-mail: rachealpoh@wou.edu.my

Article Info

Article History:

Received Jun 20, 2026
Revised Jul 17, 2026
Accepted Aug 28, 2026

Keywords:

Sign Language Recognition,
FPGA,
RTL,
Wearable Sensors,
Flex Sensor,
Sensor Fusion,
Application-Specific Hardware,
Real-Time Text-to-Speech Output

ABSTRACT

This paper proposes an FPGA-prototyped, application specific Register Transfer Level (RTL) architecture that translates hand gestures captured by a dual-glove wearable sensor system into real-time text and audio output. Each glove incorporates five flex sensors and one accelerometer, providing ten flex and two motion inputs for a total of twelve; the binary flex states provide 2^10 = 1,024 possible finger configurations, while the five discretized motion states of each of the accelerometers provide 5^2 = 25 combined motion states, yielding a theoretical sensor-state space of 25,600 combinations. The proposed hardware pipeline executes sensor interface, finger-state encoding, motion-state classification, sensor fusion, command decoding, and message generation completely in dedicated RTL, as opposed to using software on a general purpose or embedded processor. The final output of the system is a real-time, human-readable translation of the signer's hand gesture, displayed as text and rendered as audio speech -- generated directly from raw glove sensor data with no intermediate processor or ML inference stage in the recognition pipeline. The architecture is functionally verified through RTL simulation, and demonstrated on an FPGA prototype, laying a hardware-validated foundation for a subsequent ASIC implementation in which area, power, and timing can be quantitatively optimized.

Corresponding Author:

Heng-Wei Lee,
School of Digital Technology, Wawasan Open University (WOU),
George Town 10050, Penang, Malaysia.
E-mail: hwlee@wou.edu.my

1. INTRODUCTION

Sign language recognition can ease the communication barrier between sign-language users and people unfamiliar with sign language. Wearable systems using flex and inertial/motion sensors provide a method of acquiring bending and motion data from the fingers and hands independent of camera position, illumination, and background, which are known limitations of vision-based

recognition [1]. Wearable sensor-based sign-language recognition has been reviewed comprehensively [2], and glove-mounted stretchable sensor arrays have demonstrated real-time sign-to-speech translation [3]. FPGA approaches have been explored to accelerate sign-language recognition and reduce processing latency using hardware parallelism [4].

Most existing wearable solutions have been built using microcontrollers, embedded processors, or software-implemented machine-learning models for classification. While such platforms are flexible, a general-purpose processor incurs overhead in the form of decode of instructions, cache hierarchy, and general arithmetic units, not needed for a specific recognition task. Application-specific hardware encodes the needed operations directly as digital logic, eliminating that overhead, which is the central motivation for domain-specific architectures [5]. Quantitative comparisons have shown that an ASIC implementation can be substantially smaller, faster and more power-efficient than the same logic on an FPGA [6], and FPGA-based design space exploration can be utilized as a low-risk path to explore such hardware before committing to an ASIC [7].

This paper proposes an application-specific RTL architecture for a dual-glove sign-language translation system. The architecture utilizes 10 flex sensors and 2 accelerometers across both gloves, and executes sensor encoding, motion classification, sensor fusion, command decoding, and output generation entirely in FPGA logic. To verify the core recognition-to-output pipeline before committing to the full two-hand fusion stage, a single-glove sub system of this architecture was first implemented and functionally verified in RTL simulation and on FPGA hardware. This paper describes the complete dual-glove architecture along with the single-glove hardware verification, and outlines the extension path to the full two-glove system.

The remainder of this paper is organized as follows. Section 2 reviews related work on wearable sign-language systems and FPGA-based recognition. Section 3 presents the proposed dual-glove methodology and system architecture. Section 4 describes the FPGA/RTL implementation. Section 5 discusses the simulation and hardware prototype results. Finally, Section 6 concludes the paper and outlines future work.

2. LITERATURE REVIEW

Wearable sign-language systems are increasingly using flex and IMU sensors to capture hand posture and motion, and sensory-glove systems developed over the last decade have been surveyed in detail [8]. Cheok et al. reviewed hand gesture and sign-language recognition techniques more broadly [1]. Kudrinko et al. reviewed wearable sensor-based sign-language recognition, covering the sensor types, recognition methods and open challenges of glove-based systems [2]. Zhou et al. demonstrated a glove fitted with stretchable sensor arrays that translates signs into speech with machine-learning assistance [3], and Wen et al. combined a triboelectric smart glove with artificial-intelligence models for sign-language recognition and bidirectional communication in virtual-reality space [9]. However, the wearable systems are mostly devoted to sensing, software processing, and communication, and not hardware recognition.

The FPGA-based studies have predominantly accelerated vision-based recognition models. Yang et al. developed an FPGA accelerator for high-accuracy sign-language recognition based on a residual network [4]. Such FPGA studies emphasize computational acceleration, and not sensor driven RTL architectures.

Hennessy and Patterson argued that domain-specific architectures, which tailor hardware to a single application domain, offer large gains in performance and efficiency over general-purpose

processors [5]. Kuon and Rose quantified the area, delay and power gap between FPGA and ASIC implementations [6], whereas Hammerquist and Lysecky explored the design space of application-specific FPGAs in system-on-a-chip designs [7].

Research Gap: The existing works are mostly focused on either wearable sensing and software processing or FPGA acceleration of vision-based recognition models. There is limited work that focuses on a dedicated flex/IMU sensor-to-output RTL architecture with a staged validation from a single-sensor module to a complete multi-glove system. This work fills this research gap by addressing a hardware oriented sensor-to-output design.

3. PROPOSED METHODOLOGY

The proposed system consists of two sensor gloves. Each glove contains five flex sensors and one accelerometer. The flex sensors capture individual finger bending, while the accelerometers provide hand-motion information.

For ten binary flex inputs,

$$N_F = 2^{10} = 1024 \tag{1}$$

possible finger configurations are available. Each accelerometer is classified into five motion states. Therefore,

$$N_A = 5^2 = 25 \tag{2}$$

combined motion states are obtained. The theoretical sensor-state space is therefore

$$N_{total} = 2^{10} \times 5^2 = 25{,}600. \tag{3}$$

This value represents the **theoretical sensor-state combinations**, not 25,600 experimentally validated signs.

3.1. Proposed System Architecture

The overall block diagram of the proposed dual-glove sign-language translation system is shown in Figure 1.

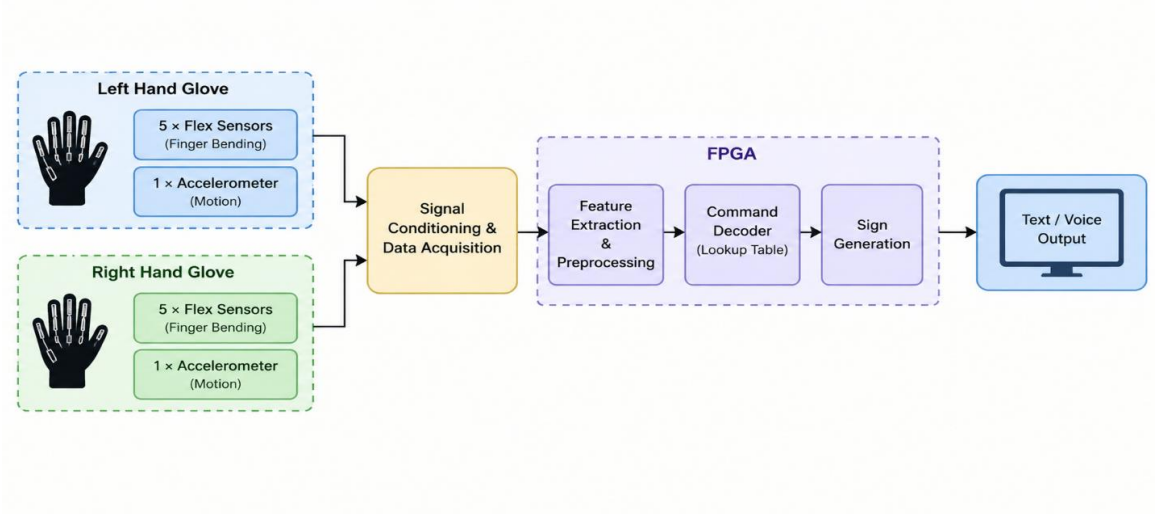


Figure 1. Proposed Architecture Block Diagram

The presented architecture includes two sensor gloves that track the bending of the fingers using flex sensors and the movement of the hands using accelerometers. The signals from the

sensors are conditioned and preprocessed to enter the FPGA. The FPGA extracts the features, decodes the command using a lookup table, and generates the appropriate sign resulting in the given text or voice.

3.2. Working Principle

During operation, the two gloves simultaneously capture finger and hand-motion information. Each flex sensor is represented by a binary state,

$$f_i \in \{0, 1\}, i = 1, \dots, 10 \tag{4}$$

where the state represents the corresponding finger condition.

The left and right glove inputs are represented as

$$F_L = [f_1, f_2, f_3, f_4, f_5] \tag{5}$$

And,

$$F_R = [f_6, f_7, f_8, f_9, f_{10}] \tag{6}$$

The accelerometer outputs are converted into five predefined motion states:

$$D_L, D_R \in \{0, 1, 2, 3, 4\} \tag{7}$$

The complete gesture state can therefore be represented as

$$S = [F_L, F_R, D_L, D_R] \tag{8}$$

The RTL logic processes this state through the encoder, motion classifier, fusion unit, and command decoder. When a valid predefined state is detected, the corresponding message is generated and transferred to the display and audio output.

3.3. Sensor Configuration and Digital Representation

Each glove comprises of five flex sensors, a sensor for each of the five fingers, plus an accelerometer, used to capture hand-orientation information. Consequently, the two gloves yield ten finger-related input features and two motion-related input features. The inherent symmetry in the configuration facilitates the organization of the RTL architecture, as similar processing can take place on the channels corresponding to the left and to the right gloves prior to the combination of their information.

The flex-sensor information is encoded as binary states by the proposed abstraction. Each finger therefore yields a single digit, signifying whether it has been bent or not, according to its predefined condition. The ten binary digits can encode the 1024 possible combinations of the ten fingers. It is worth noting that the encoding is compact since the objective of the architecture is not to preserve the continuous analog measurement along the recognition pipeline, but rather to convert it into a deterministic digital representation.

The accelerometer information is discretized into 5 possible motion states for each glove. Consequently, the combination of two accelerometers produces 25 possible combinations of motion information. Combining both the representation of the finger-related and the motion-related observations yields the theoretical sensor-state space of the existing methodology. These combinations form the addressable representation of the architecture; the command lookup mechanism then identifies which of the combinations has been associated with a vocabulary item.

3.4. Sensor-State Encoding

Sensor-state encoding takes care of providing an interface between the sensory system and the digital recognition pipeline. The aim of this stage is to encode the individual sensor conditions into compact digital representations which can be further processed synchronously by the FPGA logic.

For the flex sensors, the binary representation allows to directly associate the state of each finger with a certain bit position. The resulting vector provides a compact representation of the hand posture. Furthermore as left and right gloves are using the same sensor arrangements their vectors can be processed using equivalent logic structures.

The motion classifier performs a similar abstraction on the accelerometer information. Instead of retaining unrestricted continuous motion values in the command-decoding stage, the information is mapped to one of the predefined motion categories and therefore reduces the complexity in following fusion and command lookup operations.

The separation between sensor-state encoding and command decoding also allows modularity optimizations. Changes to the sensor interface or state thresholds can be isolated from the command mapping logic, provided that the resulting encoded state follows the defined interface.

3.5. Motion-State Classification

Motion classification is required because the information provided by the accelerometer cannot be directly utilized as a binary state of a finger. The proposed architecture therefore classifies the accelerometer information into five predetermined states. The states allow for a compact representation of the motion component of a gesture.

The classification stage can be seen as an abstraction layer between the physical sensor and the digital fusion unit. Its output is not to be taken as to represent every possible physical movement. Rather it provides the limited number of motion categories that are needed by the predefined recognition vocabulary. This allows for later stages of the architecture to operate with discrete digital states.

The use of discrete motion categories also makes the architecture compatible with a lookup-based command decoder. Once the motion state has been identified it can be combined with the finger configuration to establish a deterministic gesture representation. A valid representation can then be mapped to a corresponding command without requiring iterative software inference.

3.6. Dual-Glove Sensor Fusion

The sensor-fusion stage combines information from both gloves. This is necessary for gestures in which left and right hands contribute to a single meaning. Each glove individually encodes a local representation of the finger positions and movements. These local states are then combined to a global gesture state:

The fusion-process can be seen as concatenated encoding of both representations, which are then merged into one representation containing information from both gloves. Since both gloves share the same sensing configuration, the field of each hand retains a predictable arrangement in which each element occupies a specific position.

The suggested architecture does not consider all possible fused states as valid signs. Instead, the command decoder only recognizes representations that explicitly assign to a certain

command. Other states are thus not defined as valid signs by the system and hence can be ignored instead of having to be recognized as experimental gestures.

3.7. Command Decoding

After the sensor fusion process, the resulting digital representation must be passed on to a command decoding stage. The proposed design makes use of a lookup-table approach to associate a set of predefined gesture states with their respective output messages. This is particularly suitable to an application, where the to-be-recognized vocabulary is fixed.

A lookup-table based decoder has the inherent property of determinism. Given a valid input state, it returns the associated predefined message. The same input condition will always yield the same message, as long as the lookup contents are held constant. This is particularly beneficial from an RTL point of view, as it eliminates the runtime variability, which is inherent to any software-based model execution.

The single-glove approach illustrates this concept in form of a ROM-based command decoder. The relevant input state is used as an address, and the stored message is read and passed on to the LCD controller. The same principle can be applied to the complete dual-glove representation.

3.8. Output Generation

The output stage transforms the decoded command into a human-readable representation. The current prototype demonstrates text output via the LCD path and associated wireless/message path. Output stage is thus the final step in the transition from internal digital recognition state to user-visible information.

The separation of command decoding and output generation is an additional modular boundary. Same decoded command can potentially be connected to a variety of output interfaces without affecting the fundamental recognition logic. Text display and audio-output interface can consume the decoded message via two separate output modules.

This modular organization is very useful for future expansion since it allows the recognition core to remain independent of the particular human-machine interface chosen for deployment.

4. FPGA/RTL IMPLEMENTATION

The principal hardware evaluation parameters of the proposed dual-glove architecture, including the sensor configuration and the resulting sensor-state space, are summarized in Table 1.

Table 1. Hardware Evaluation Parameters

Parameter	Value
Flex sensors per glove	5
Total flex sensors	10
Binary finger configurations	$2^{10} = 1024$
Accelerometers	2
Motion states/accelerometer	5
Combined motion states	$5^2 = 25$
Theoretical sensor states	25,600

The proposed RTL architecture is simulated as a verification step prior to FPGA deployment. The simulation checks sensor-state transition, reset behavior, generation of valid-state output, and command decoding, plus output timing.

The FPGA is used as a hardware platform to prototype the system; it allows testing the RTL code with real-life wearable sensor inputs. The prototype implements the full signal processing chain, from the input sensors to the user outputs. The architecture is optimized around the required sign-recognition functions, rather than a generic processor core. This is important for future-hardware exploitation: the RTL architecture could benefit from hardware-specific optimization. Application-specific design methodologies usually explore the design space around area, timing, and power constraints [7]. FPGA prototyping also provides a practical step towards ASIC implementation, where much of the area, speed and power gap between the two technologies can be recovered [6].

4.1. RTL Module Organization

The proposed RTL architecture involves a set of processing stages, which are functionally separated. The first group of modules processes sensor input data and converts it into states. The second group classifies motion and performs sensor-fusion, and the command decoder interprets the determined state as a particular message. Finally, the output-control logic transfers the message to the display or communication interface.

The described architecture implies clear interfaces between the modules, while their functional separation allows for easier verification. In fact, the given structure was applied to the single-glove implementation, where the input sensor information was registered, decoded using a ROM-like structure, and sent to the LCD controller (Figure 3).

Accordingly, the architecture can be considered as a pipeline of data processing, where each stage implements a limited set of functions. This is explained by the application-specific nature of the given design, which does not require any unnecessary general-purpose operations.

4.2. Clock and Synchronization Considerations

Digital hardware requires a careful treatment of signals arriving from different clock domains. In particular, sensor-valid or asynchronous control signals may have an undefined relation to the internal FPGA clock. Directly applying such a signal may lead to metastability or improper switching, which is conventionally prevented by passing it through a multi-flop synchronizer [10].

The presented single-glove subsystem is using the two-flop synchronizer and the rising edge detector for the given signal to synchronize the register holding the sensor state. The former circuit is necessary to keep the FPGA clock domain control, while the latter is detecting the rising edge for the registered update event. The given implementation is especially useful for the described scenario when the external sensor and clock domains are not naturally aligned. Furthermore, such a structure allows avoiding unnecessary state updates between the clock cycles.

4.3. Rom-Based Decoder Implementation

The command decoder uses a ROM-oriented structure. The single-glove subsystem uses the concatenated direction/flex information as a compact address; the corresponding message is read out of a message register for feeding to the LCD controller. This organization provides an efficient use of the hardware for a fixed vocabulary since the recognition commands are predetermined and the decoder does not need to perform any significant arithmetic or iterative classification. The output is simply selected according to the encoded sensor state.

The ROM-based organization also provides for a simple modification of the recognition vocabulary. A different set of recognized commands can be obtained by changing the mapping corresponding to a set of states without affecting the overall data-processing structure.

4.4. Display Interface

The LCD controller provides the interface between the decoded internal message and the actual LCD display. The simulation waveform in Figure 2 shows the initialization sequence followed by the data transfer. The controller generates the necessary LCD data and control signals such as register-select and enable.

The use of a dedicated display controller prevents the recognition logic from having to manage the low-level timing requirements of the display. The decoding logic therefore generates the message while the display module handles the conversion to the interface transactions. This modularity allows the recognition logic to focus on sensor interpretation and selection of command.

4.5. FPGA as a Prototyping Platform

FPGA implementation allows for an intermediate step between RTL design and possible ASIC realization. Due to its reconfigurable nature, the FPGA allows for architectural modifications and testing without dedicated silicon fabrication [11]. This is particularly beneficial for early hardware development, where interface behavior, state encoding, and command mapping may require refinement. As such, this work utilizes the FPGA deployment to confirm that the designed RTL can work with actual sensor inputs and generate a physical output. In this way, the hardware prototype serves to complement the RTL simulation: the former allows observing the behavior of the architecture beyond a purely simulated environment.

The FPGA should not be considered as the final application-specific chip. On the contrary, this is a proof-of-concept implementation, which allows for practical validation of the described architecture. Once the full dual-glove system is implemented and characterized, the same RTL concepts can be used as a basis for synthesis and future ASIC-specific optimizations. The FPGA prototype provides a physical validation of the sensor-to-output path for the single-glove subsystem. Flex sensors mounted on the glove were connected to the FPGA development board, allowing the physical sensor condition to enter the implemented RTL pipeline.

The decoded output was observed on the onboard LCD, while the corresponding text was also transferred to a smartphone through the demonstrated communication path. The displayed message, including the demonstrated “RESTROOM” output (Figure 4), establishes that the recognition result can travel from the physical glove sensors through the digital hardware and finally reach a human-readable destination.

5. RESULTS AND DISCUSSION

5.1. Simulation Waveform

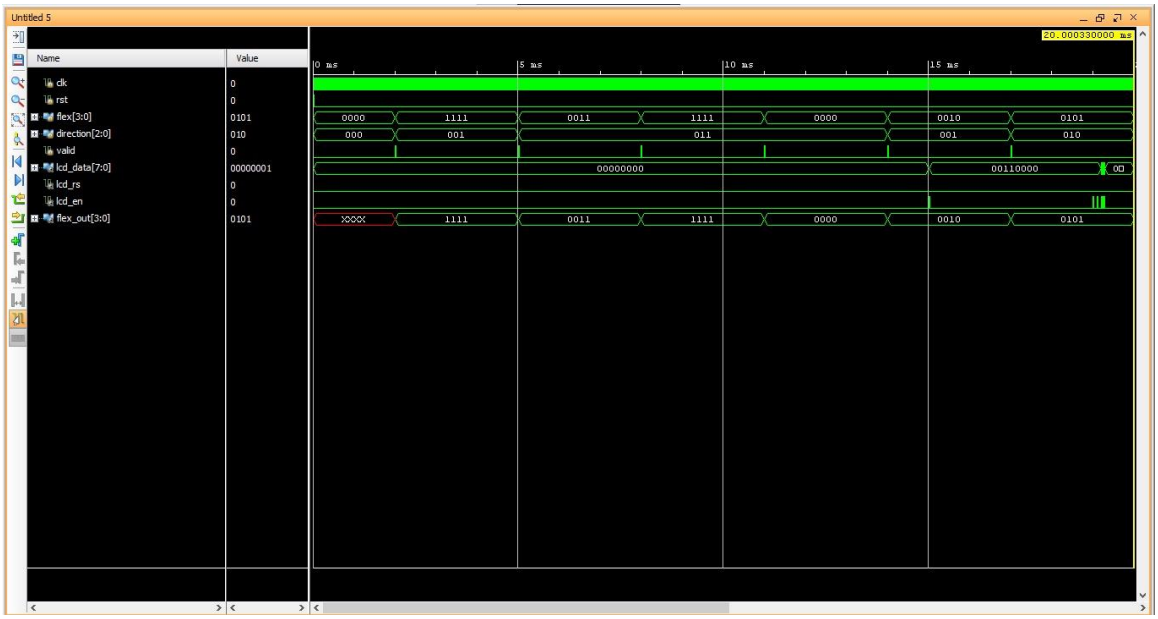


Figure 2. RTL simulation waveform of the single-glove subsystem

Figure 2 shows the RTL simulation waveform of the single-glove subsystem, in which a new flex and direction combination is applied at each valid pulse. The registered output flex_out[3:0] remains undefined until the first valid sample after the reset. It follows the flex[3:0] at every valid occurrence and retains its value in between, implying that the design updates only at the valid synchronized samples. The lcd_data[7:0] bus is initially idle, followed by displaying the LCD command values (0x30, then 0x01) with the lcd_rs low and lcd_en pulsing which is the display initialization sequence.

5.2. RTL Schematic

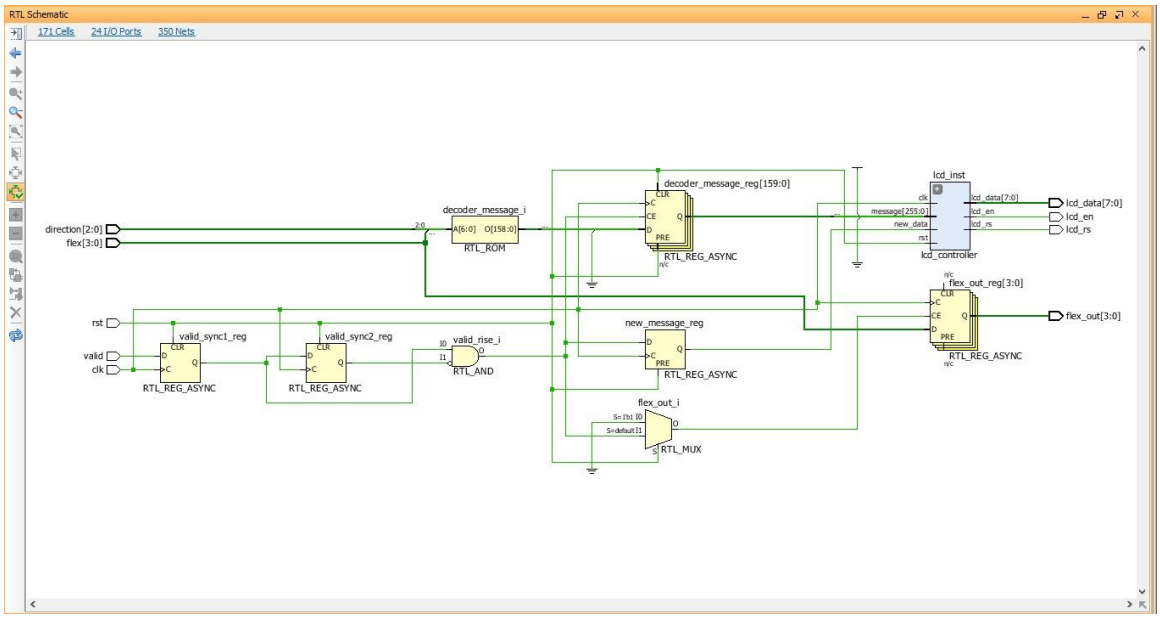


Figure 3. Elaborated RTL schematic of the single-glove subsystem

Figure 3 shows the elaborated RTL schematic of the proposed subsystem (171 cells, 24 I/Os, 350 nets). The 7-bit concatenation of direction[2:0] and flex[3:0] selects an address in the command-decoder ROM, whose output is stored in decoder_message_reg[159:0]. This register feeds the LCD controller which drives lcd_data[7:0], lcd_rs and lcd_en. The asynchronous valid input goes through a two-flop synchronizer and a rising-edge detector before being used to trigger any update.

5.3. Hardware Prototype Validation

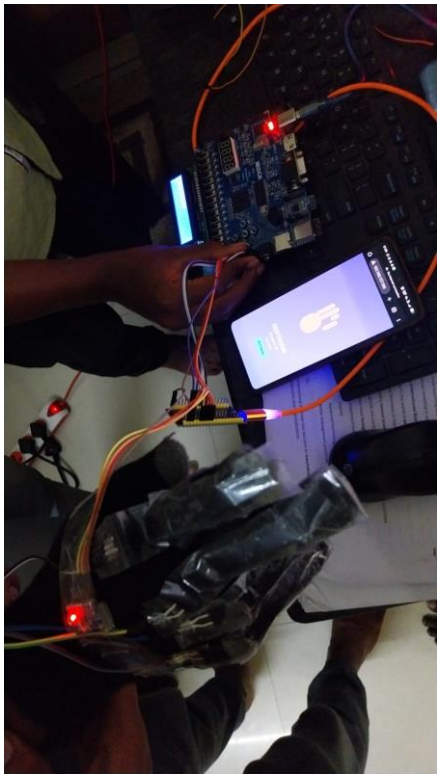


Figure 4. Hardware prototype: sensor glove connected to the FPGA development board

Figure 4 shows the hardware prototype. On the FPGA board, the glove-mounted flex sensors were driving the RTL pipeline. The decoded message was sent to the on-board LCD and was simultaneously output as text ("RESTROOM") on a smartphone. This established the flow of the signal from sensors through the pipeline to a human-readable form on the physical board for the single-glove case. Recognition accuracy, latency, resource usage, and the second glove with a two-hand fusion stage remain subjects for future work.

6. CONCLUSION

This paper introduced an FPGA-prototyped, application-specific RTL architecture for real-time sign-language translation, employing dual-glove sensor fusion, as well as an associated functionally validated single-glove hardware subsystem that implemented its core sensor-to-output pipeline. The complete architecture makes use of ten flex sensors and two accelerometers, in combination, to capture finger configuration and hand motion, over a theoretical 25,600 combinations, using dedicated RTL for sensor interfacing, finger-state encoding, motion-state classification, sensor fusion, command decoding, and output generation. The single-glove prototype establishes that flex-state capture, ROM-based command decoding, clock-domain-safe output gating, and LCD/wireless message delivery all function correctly in hardware, without a

general-purpose processor in the recognition path. Extending this validated pipeline to the full dual-glove system — including the second glove's sensor interface and a two-hand fusion stage — and characterizing its accuracy, latency, and resource utilization on real synthesis data, is the immediate next step of this work, towards an eventual ASIC implementation.

REFERENCES

[1] Cheok, M. J., Omar, Z., & Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *International Journal of Machine Learning and Cybernetics*, *10*(1), 131-153. https://doi.org/10.1007/s13042-017-0705-5

[2] Kudrinko, K., Flavin, E., Zhu, X., & Li, Q. (2021). Wearable sensor-based sign language recognition: A comprehensive review. *IEEE Reviews in Biomedical Engineering*, *14*, 82-97. https://doi.org/10.1109/RBME.2020.3019769

[3] Zhou, Z., Chen, K., Li, X., Zhang, S., Wu, Y., Zhou, Y., Meng, K., Sun, C., He, Q., Fan, W., Fan, E., Lin, Z., Tan, X., Deng, W., Yang, J., & Chen, J. (2020). Sign-to-speech translation using machine-learning-assisted stretchable sensor arrays. *Nature Electronics*, *3*(9), 571-578. https://doi.org/10.1038/s41928-020-0428-6

[4] Yang, D., Li, J., Hao, G., Chen, Q., Wei, X., Dai, Z., Hou, Z., Zhang, L., & Li, X. (2024). Hardware accelerator for high accuracy sign language recognition with residual network based on FPGAs. *IEICE Electronics Express*, *21*(4), 20230579. https://doi.org/10.1587/elex.21.20230579

[5] Hennessy, J. L., & Patterson, D. A. (2019). A new golden age for computer architecture. *Communications of the ACM*, *62*(2), 48-60. https://doi.org/10.1145/3282307

[6] Kuon, I., & Rose, J. (2007). Measuring the gap between FPGAs and ASICs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *26*(2), 203-215. https://doi.org/10.1109/TCAD.2006.884574

[7] Hammerquist, M., & Lysecky, R. (2008). Design space exploration for application specific FPGAs in system-on-a-chip designs. In *2008 IEEE International SOC Conference* (pp. 279-282). IEEE. https://doi.org/10.1109/SOCC.2008.4641527

[8] Ahmed, M. A., Zaidan, B. B., Zaidan, A. A., Salih, M. M., & Lakulu, M. M. b. (2018). A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017. *Sensors*, *18*(7), 2208. https://doi.org/10.3390/s18072208

[9] Wen, F., Zhang, Z., He, T., & Lee, C. (2021). AI enabled sign language recognition and VR space bidirectional communication using triboelectric smart glove. *Nature Communications*, *12*, 5378. https://doi.org/10.1038/s41467-021-25637-w

[10] Ginosar, R. (2011). Metastability and synchronizers: A tutorial. *IEEE Design & Test of Computers*, *28*(5), 23-35. https://doi.org/10.1109/MDT.2011.113

[11] Cong, J., Liu, B., Neuendorffer, S., Noguera, J., Vissers, K., & Zhang, Z. (2011). High-level synthesis for FPGAs: From prototyping to deployment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *30*(4), 473-491. https://doi.org/10.1109/TCAD.2011.2110592