

2011

A special purpose embedded system for neural machine interface for artificial legs

Xiaorong Zhang
University of Rhode Island

He Huang
University of Rhode Island

Qing Yang
University of Rhode Island, qyang@uri.edu

Follow this and additional works at: https://digitalcommons.uri.edu/ele_facpubs

Citation/Publisher Attribution

X. Zhang, H. Huang and Q. Yang, "A special purpose embedded system for neural machine interface for artificial legs," *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011, pp. 5207-5210, doi: 10.1109/IEMBS.2011.6091288
Available at: <https://doi.org/10.1109/IEMBS.2011.6091288>

This Article is brought to you by the University of Rhode Island. It has been accepted for inclusion in Electrical, Computer, and Biomedical Engineering Faculty Publications by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu. For permission to reuse copyrighted content, contact the author directly.

A special purpose embedded system for neural machine interface for artificial legs

The University of Rhode Island Faculty have made this article openly available.
Please let us know how Open Access to this research benefits you.

This is a pre-publication author manuscript of the final, published article.

Terms of Use

This article is made available under the terms and conditions applicable towards Open Access Policy Articles, as set forth in our [Terms of Use](#).



Published in final edited form as:

Conf Proc IEEE Eng Med Biol Soc. 2011 ; 2011: 5207–5210. doi:10.1109/IEMBS.2011.6091288.

A Special Purpose Embedded System for Neural Machine Interface for Artificial Legs

Xiaorong Zhang [Student Member, IEEE], He Huang [Member, IEEE], and Qing Yang [Fellow, IEEE]

The authors are with the Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881, USA

Abstract

This paper presents a design and implementation of a neural-machine interface (NMI) for artificial legs that can decode amputee's intended movement in real time. The newly designed NMI integrates an FPGA chip for fast processing and a microcontroller unit (MCU) with multiple on-chip analog-to-digital converters (ADCs) for real-time data sampling. The resulting embedded system is able to sample in real time 12 EMG signals and 6 mechanical signals and execute a special complex phase-dependent classifier for accurate recognition of the user's intended locomotion modes. The implementation and evaluation are based on Altera's Stratix III 3S150 FPGA device coupled with Freescale's MPC5566 MCU. The experimental results for classifying three locomotion modes (level-ground walking, stairs ascent, and stairs descent) based on data collected from an able-bodied human subject have shown acceptable performance for real-time controlling of artificial legs.

I. Introduction

The technology of neurally controlled artificial limbs has advanced rapidly in recent biomedical research [1–6]. Compared with computerized prostheses without neural control, neurally controlled artificial limbs perform and feel like natural limbs. A neural-machine interface (NMI) that deciphers neural signals from amputees to identify the users' intended movements is the center of the neural control system for artificial limbs. The NMI needs to be realized in an embedded computer so as to be carried by amputees.

Electromyographic (EMG) signals recorded from muscles are effective electrical signals for expressing movement intent [7]. While EMG-based NMI has been tested clinically for artificial arm control [1–2], there has been no commercial EMG-controlled prosthetic leg available. This is partly because the EMG signals recorded from leg muscles are highly non-stationary. Accurately decoding the user intent from such signals is difficult. Furthermore, the accuracy in identifying the lower-limb movement is essential because any incorrect decision may cause the user to stumble and even fall. To address these challenges, our research group has developed a phase-dependent EMG pattern recognition (PR) strategy to dynamically classify the user's locomotion modes [3]. A novel, neuromuscular-mechanical fusion technique that incorporates neuromuscular control information in the form of EMG signals and mechanical forces/moments acting on prostheses has been proposed to further improve the accuracy of the PR algorithm [4].

The realization of our new PR strategy on PC shows a high accuracy in identifying user's locomotion modes [4]. Although the software implementation on PC is useful to verify the correctness of the neural decoding algorithm, it is not applicable for amputees to wear in real life. Realizing the NMI in an embedded computer is required and challenging because of computation complexity of the PR algorithm coupled with real-time requirement of

controlling artificial legs. Such an embedded system should provide high computation speed for the PR algorithm since any delayed decision may result in unsafe use of the prosthesis. Memory resources need to be carefully managed because fast memory resources on embedded computers are usually very limited. Effective timing control is also required to guarantee smooth control of artificial legs. The execution time of the neural machine interfacing algorithm for one analysis window is usually expected to be less than 20 ms to ensure the safe use of artificial legs.

Our prior NMI design was implemented on a Freescale's MPC5566 132 MHz 32 bits microcontroller unit (MCU) that can accurately classify sitting and standing in real-time [5]. The measured delay for generating a decision in one analysis window with 7 EMG inputs was around 80 ms [5]. If more tasks such as walking on different terrains are considered, more EMG channels and auxiliary mechanical signals need to be collected, and a more complicated phase-dependent classifier needs to be applied for accurately determining the user's locomotion modes. Existing embedded systems generally cannot be directly applied to such a system for NMI to provide real time performance. To tackle this problem, we designed a parallel PR algorithm tailored to the FPGA device for classifying sitting and standing [6]. The offline measurements based on Altera Stratix II GX EP2SGX90 FPGA device showed a speedup of around 280X over the software implementation based on MPC5566 MCU [6], implying that FPGA-based parallel design is a promising approach to realize real-time NMI for artificial legs.

This paper presents an integrated design of a special purpose embedded system realizing a complete NMI for artificial legs. It has an MCU with multiple built-in ADCs for real-time data sampling and dispatching and an FPGA device for fast data decoding. The neuromuscular-mechanical fusion-based phase-dependent PR algorithm is parallelized and mapped to the FPGA device. The implementations are based on Freescale's MPC5566 evaluation board (EVB) and Altera's DE3 education board with a Stratix III 3S150 FPGA device. The experimental results for classifying three locomotion modes (level-ground walking, stairs ascent, and stairs descent) with 12 EMG signals and 6 mechanical signals have shown that the average execution time of PR for one analysis window is 0.25 ms. A 38X speedup is observed over the software implementation on a PC with 3.2 GHz Intel i3 processor and 6GB RAM.

II. Embedded System Architecture

A. System Architecture

The system architecture of the new NMI for artificial legs is shown in Fig. 1. The embedded NMI senses signals from two physical systems a human neuromuscular system and a mechanical prosthetic leg, and decodes these signals to control the prosthesis. The NMI contains two modules: a MCU module for data sampling and dispatching, and an FPGA module for fast data decoding and pattern recognition. Data are transferred between these two modules using serial peripheral interface (SPI).

1) MCU Module—Multichannel EMG signals are collected from multiple electrodes mounted on patient's residual lower-limb muscles. Mechanical forces/moments are recorded from a 6 degrees-of-freedom (DOF) load cell mounted on the prosthetic pylon. The EMG signals and the mechanical signals are preprocessed by filters and amplifiers and then simultaneously streamed into on-board ADCs of the MCU. The direct memory access (DMA) engine transfers the digitized input data from the ADCs to the RAM without direct involvement of the processor. Data are then sent to the FPGA module through SPI bus.

2) FPGA Module—Once the EMG and mechanical data are received by the FPGA device, they are stored in the on-chip RAM and segmented by sliding analysis windows with a fixed window length and a window increment. The FPGA system works in two modes: *classifier training* and *pattern recognition*. In the *classifier training* mode, a large amount of continuous analysis windows are collected to train the classifier. The parameters of the trained classifiers are stored in the RAM for later use in the *pattern recognition* mode. In the *pattern recognition* mode, analysis windows are processed continuously in real-time. One classification decision is produced for each window to identify the user's intended movement.

B. Phase-Dependent Pattern Recognition

The architecture of the neuromuscular-mechanical fusion-based phase-dependent PR strategy for artificial legs is shown in Fig. 2. To identify the user intent, we need first extract features from each input channel and then choose a classifier to assign the intended locomotion mode. The phase-dependent classifier consists of a gait phase detector and multiple classifiers [3]. Each classifier is trained for a specific gait phase. In the real-time pattern recognition process, current gait phase is first determined by the gait phase detector in each analysis window, and then the classifier associated with that specific phase is adopted to do the classification.

1) Gait Phase Detection—In this study, four gait phases are used: initial double limb stance, single limb stance, terminal double limb stance, and swing [4]. The real-time gait phase detector is built based on the vertical ground reaction force (GRF) measured from the 6-DOF load cell.

2) Feature Extraction—Features are extracted in every analysis window from each input channel. In this study, four EMG time-domain (TD) features (mean absolute value, number of zero crossings, waveform length, and number of slope sign changes) are used [8]. For the mechanical forces/moments, the mean value is computed from each individual DOF as the mechanical feature. The features extracted from each channel are fused and normalized into one feature vector for each analysis window. The feature vector is then sent to the classifier for pattern recognition.

3) Pattern Recognition—A simple linear discriminant analysis (LDA) classifier is adopted in this study because of its computation efficiency for real-time prosthesis control and the comparable accuracy to more complex classifiers [5, 9].

III. System Implementation & Prototype

The system implementation is based on Freescale's MPC5566 132 MHz 32 bits MCU with the Power Architecture™ and Altera's DE3 education board with a Stratix III 3S150 FPGA device. The prototype board for our NMI system is shown in Fig. 3. This paper only presents the implementation and experimental results of the PR algorithm for testing phase. We choose the window length and the window increment to be 160 ms and 20 ms, respectively. The implementation of the classifier training algorithm will be presented in the future work.

A. Timing Control and Memory Management

The MPC5566 MCU has 40 on-chip ADC channels with 12 bit resolution, 32 KB unified cache and 128 KB SRAM. It also has four SPI modules that each can be configured as either an SPI master or a slave, and a DMA controller that supports up to 64 channels. The ADC channels samples EMG signals and mechanical forces/moments at the rate of 1000 Hz. Therefore, for each channel every analysis window contains 160 data samples. Twenty new

samples are streamed into the MCU in every window increment. To guarantee the smooth control of the prosthesis, efficient timing control and memory management is required. Fig. 4 shows a simplified diagram of our strategy for one data channel. In the diagram, we use two RAM blocks, $B1$ and $B2$ to store the input data. Both blocks have the capacity of 20 data samples. While $B1$ is used for storing new incoming data from the ADC module, the old data in $B2$ are being transferred to the FPGA device through the SPI bus for pattern recognition. In this manner, real-time data sampling for new analysis window and pattern recognition for the current window can be done simultaneously. Since the sampling time for filling up one RAM block is 20 ms, the total execution time of SPI transfer and PR computation must be less than 20 ms to ensure the smooth data streaming and prosthesis control. As shown in Fig. 4, if $B1$ is filled up with new samples, then it will switch roles with $B2$. At this time, new data will be stored in $B2$, and data in $B1$ will be sent to the FPGA device. On the FPGA device, a similar circular buffer is also designed to efficiently utilize the memory resources.

B. Task Parallelism and Pipelining

A new phase-dependent PR algorithm is designed and implemented based on Altera's Stratix III 3S150 FPGA device, to make the best use of the parallelism of FPGAs. The algorithm is implemented using Impulse C C-to-HDL CoDeveloper software [10]. Fixed-point data format is adopted for non-integer data in the implementation.

Fig. 5 shows the data flows and task stages of the PR algorithm. In our design, tasks are divided into multiple processes that can be executed in parallel if there are no data dependencies or in pipeline if a sequence of small processes are executed repeatedly. The data streaming between different processes is done by dual-port first-in-first-out (FIFO) RAM buffers. A single process can be associated with multiple input and output FIFO buffers. Signals are used to synchronize a group of processes if needed.

The largest benefit obtained from the FPGA design is the high parallelism of the PR algorithm. It is observed that the task procedure for each individual channel from data sampling, storing, and loading, to feature extraction is independent and almost identical so that all the channels can be processed in parallel. This greatly reduces the computation time for feature extraction, which is very important because in the software implementation of the PR algorithm, we found the computation time for feature extraction counted for 90% of the total execution time.

IV. Experimental Results

This study was conducted with Institutional Review Board (IRB) approval at the University of Rhode Island and informed consent of subjects. To verify the correctness of the FPGA-based PR algorithm and compare the performance of the new NMI design with our previous software implementation on PC, we used the same dataset to run on both platforms. The software implementation is based on a PC with Intel i3 3.2 GHz processor and 6 GB RAM, designed using Matlab programming tool. The testing dataset was previously collected from an able-bodied subject for identifying three locomotion modes including level-ground walking, stairs ascent, and stairs descent. The parameters of the trained classifiers were manually loaded into the NMI systems beforehand. Twelve input channels of EMG signals and six channels of mechanical forces/moments were used as the baseline configuration. The SPI clock was set to 1 MHz and synchronized between the MPC566 EVB and the DE3 board. The SPI data transferring time for every window increment was less than 6 ms. The resource utilization summary of the FPGA implementation is listed in Table 1.

We tested both the FPGA implementation and the software implementation for 1000 continuous analysis windows and found their classification results were completely matched. Table 2 shows comparison of the average execution time of the PR algorithm for one analysis window. Besides the baseline configuration with 12 EMG channels and 6 mechanical channels, we also tested the performance of another configuration with 6 EMG channels and 3 mechanical channels on both platforms to make a better comparison.

From Table 2 we can see, to classify one analysis window, the computation of PR for our new NMI system took less than 0.3 ms. The total execution time of SPI data transfer and PR computation was significantly less than the window increment (i.e. 20 ms), which ensured the smooth control of prosthesis. When the NMI system had 6 EMG channels and 3 mechanical channels, our new FPGA-based design gave a 27X speedup over the software implementation. When the number of channels doubled, the performance of the new design was even better, a 38X speedup was observed compared with the software implementation. This is because of the parallelism of FPGAs. The results are promising and imply that even more neural signals and mechanical signals can be effectively handled by our designed embedded system for identifying more complex activities, which is one of our future research tasks.

V. Conclusions

In this paper, a new embedded system has been designed and implemented for neuromuscular-mechanical fusion-based NMI for artificial legs. It integrates an MCU for real-time data sampling of multichannel EMG signals and mechanical signals and an FPGA device for fast PR computation. A parallel phase-dependent PR algorithm has been developed and implemented on Altera's Stratix III 3S150 FPGA device. The functionality of the new design for accurately classifying three locomotion modes including level-ground walking, stairs ascent, and stairs descent have shown great improvements over our prior work that can only classified sitting and standing. The performance of the FPGA-based implementation of PR algorithm was 38X faster than the software implementation on a PC with Intel i3 3.2 GHz processor. Future work includes real-time testing of our new NMI system on amputee subjects, minimizing power consumption, and increasing reliability.

Acknowledgments

This work is supported by National Science Foundation NSF/CPS #0931820, NIH #RHD064968A and NSF/CCF #0811333.

References

1. Parker PA, Scott RN. Myoelectric control of prostheses. Critical reviews in biomedical engineering. 1986:283–310. [PubMed: 3512166]
2. Englehart K, Hudgins B. A robust, real-time control scheme for multifunction myoelectric control. IEEE Trans Biomed Eng. 2003:848–854. [PubMed: 12848352]
3. Huang H, Kuiken TA, Lipschutz RD. A strategy for identifying locomotion modes using surface electromyography. IEEE Trans Biomed Eng. 2009:65–73. [PubMed: 19224720]
4. Zhang, F.; DiSanto, W.; Ren, J.; Dou, Z.; Yang, Q.; Huang, H. A Novel CPS System for Evaluating a Neural-Machine Interface for Artificial Legs. presented at ICCPS'11; Chicago, USA. April 2011;
5. Huang, H.; Sun, Y.; Yang, Q.; Zhang, F.; Zhang, X.; Liu, Y.; Ren, J.; Sierra, F. Integrating neuromuscular and cyber systems for neural control of artificial legs. ICCPS'10; Stockholm, Sweden. April 2010;
6. Zhang, X.; Huang, H.; Yang, Q. Design and Implementation of A Special Purpose Embedded System for Neural Machine Interface. ICCD'2010; Amsterdam, the Netherlands. October 2010;

7. Basmajian, JV.; De Luca, CJ. Muscles alive: their functions revealed by electromyography. 5. Baltimore: Williams & Wilkins; 1985.
8. Hudgins B, Parker PA, Scott RN. A new strategy for multifunction myoelectric control. IEEE Trans Biomed Eng. 1993;82–94. [PubMed: 8468080]
9. Huang H, Zhou P, Li G, Kuiken TA. An analysis of EMG electrode configuration for targeted muscle reinnervation based neural machine interface. IEEE Trans Neural Syst Rehabil Eng. 2008;37–45. [PubMed: 18303804]
10. Impulse, C. CoDeveloper from Impulse Accelerated Technologies. <http://www.impulseaccelerated.com/>

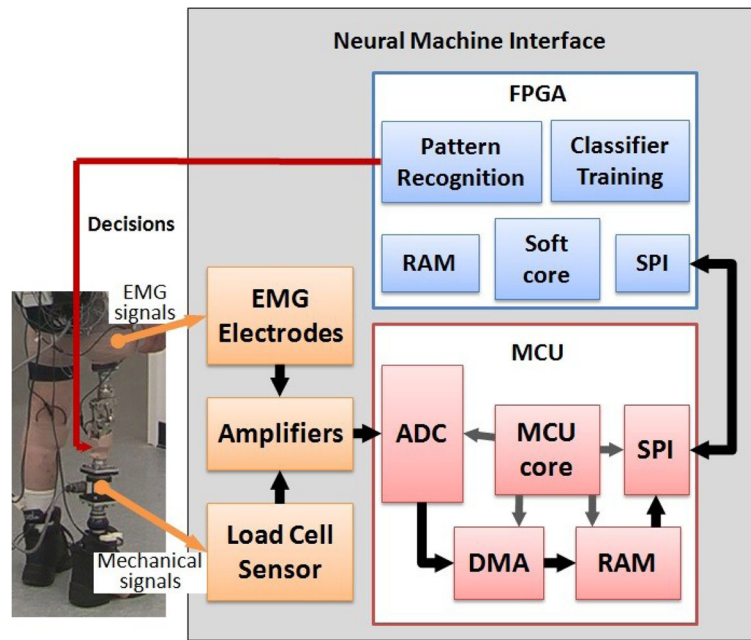


Fig. 1. System architecture of designed NMI for artificial legs.

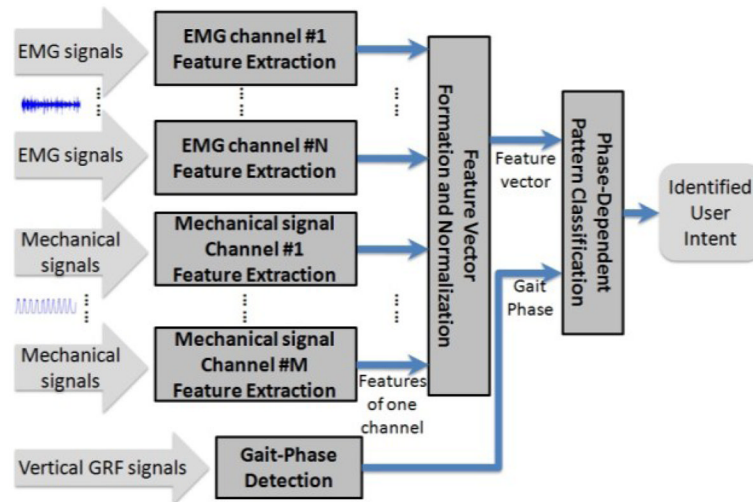


Fig. 2. Architecture of neuromuscular-mechanical fusion-based PR algorithm for artificial legs.

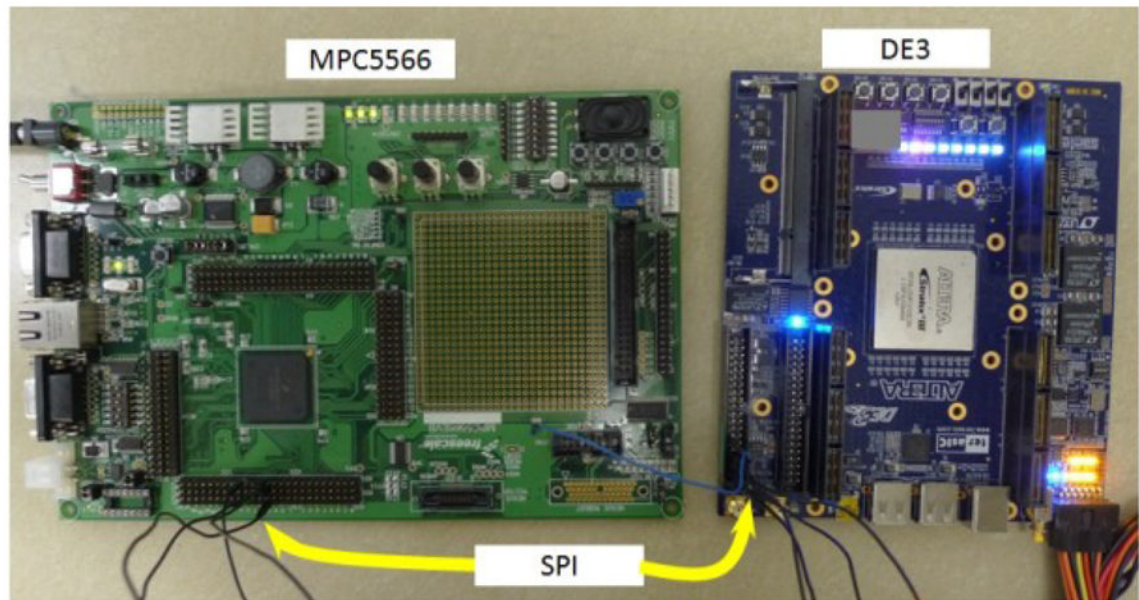


Fig. 3.
The prototype board based on MPC5566 EVB and DE3 education board.

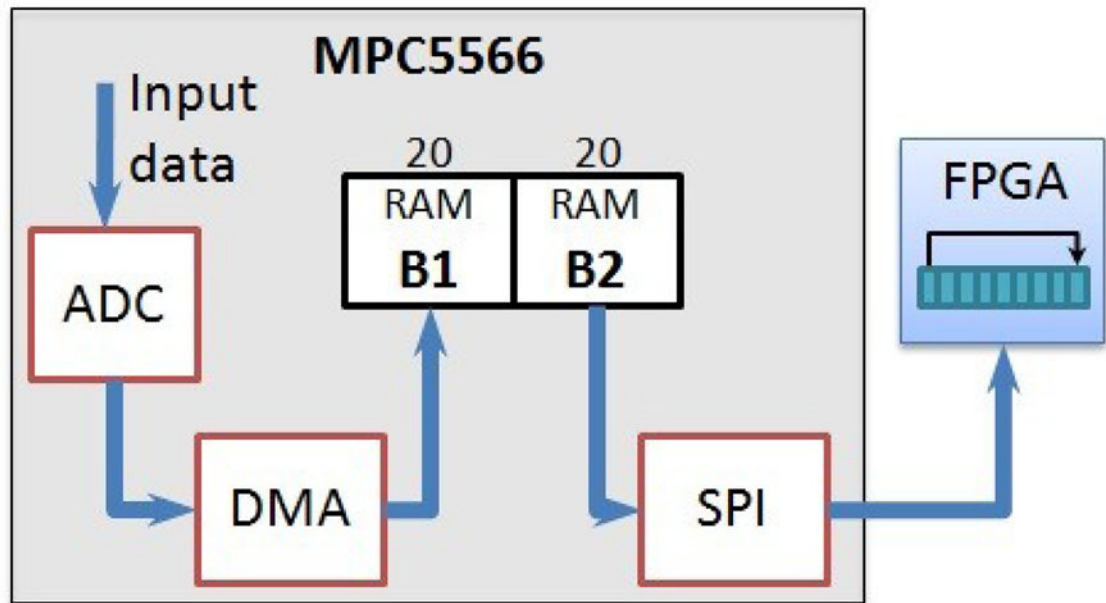


Fig. 4.
Timing control and memory management of real-time control algorithm for one channel.

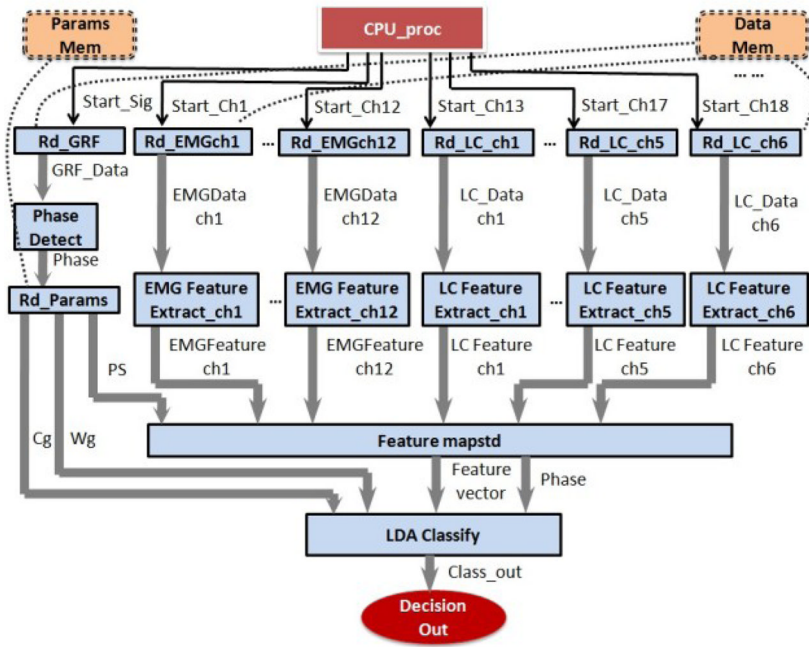


Fig. 5. Task stages and data flows of phase-dependent PR.

Table 1

Stratix III 3S150 Resource Utilization

Resource	Testing
Combinational ALUTs	36,656/113,600 (32%)
Memory ALUTs	1,504/56,800 (3%)
Registers	46,713/113,600 (27%)
Block Memory bits	902,866/5,630,976(16%)
DSP blocks	104/384 (27%)

Table 2

Comparison of PR execution time

	FPGA	Software	Speedup
6 EMG 3 Mech.	0.22 ms	6.0 ms	27x
12 EMG 6 Mech.	0.25 ms	9.5 ms	38x