Mehran University Research Journal of Engineering and Technology
Vol. 40, No. 1, 241-250, January 2021
p-ISSN: 0254-7821, e-ISSN: 2413-7219
DOI: 10.22581/muet1982.2101.21

# FPGA Implementation of RLSE Algorithm for Multichannel Brain Imaging

Muhammad Shahid Nazir[1a], Haroon-Ur-Rasheed Khan[1b], Abubaker Akram[1c], Bhagesh Maheshwari[1d], Muhammad Aqil[1e]

RECEIVED ON 21.02.2019, ACCEPTED ON 26.07.2019

## ABSTRACT

**This paper describes the implementation of a computationally efficient embedded system on an Field Programmable Gate Array (FPGA) platform for real-time brain activity estimation with multiple channels. The brain signals from multiple channels are considered as output of independent linear systems with unknown parameters representing the brain activity in corresponding channels. Multiple adaptive Recursive Least-Squares Estimation (RLSE) cores are implemented in FPGA to independently estimate the brain activity in each channel concurrently. The proposed RLSE-FPGA system provides dedicated (no time or resource sharing) and parallel processing environment. The universal asynchronous receiver transmitter core is also developed to communicate the measured and estimated parameters supported by storage facility programmed as shared memory. The computational precision is guaranteed by deploying a 32-bit floating point core for all the variables. The validation carried out by real Functional Near-Infrared Spectroscopy dataset and comparative analysis with the previously reported result, demonstrates the effectiveness of the proposed system. The computational cost endorses the effectiveness of concurrent processing of multiple channels' data in a sample before the arrival of the next sample. The proposed methodology has potential in real-time medical, military and industrial applications.**

**Keywords: FPGA, Recursive Least Squares Estimation, Brain Imaging,**

## 1. INTRODUCTION

Functional Near Infrared Spectroscopy (fNIRS) monitors the brain activity by measuring variation in the absorption of near infrared light in brain tissues. It is non-invasive technology that uses the variation in infrared light of wavelength 650–950 nm to monitor the brain activity [1]. The variation in absorption of infrared light is directly related with the concentration of Oxygenated Hemoglobin (HbO) and Deoxygenated Hemoglobin (HbR) in brain tissues. As subject is involved in performing cognitive tasks, the concentration of oxygenated and deoxygenated hemoglobin varies in brain tissues, these variations are then measured by fNIRS techniques. In recent years, many researchers have used fNIRS technique to investigate for biomedical and Brain-Computer Interface (BCI) applications [2-5]. Non-invasive feature of fNIRS, makes it attractive to utilize it for brain imaging (BI) and BCI. In reference to BCI techniques, fNIRS offers several benefits over other non-invasive technologies, like functional magnetic resonance imaging (fMRI) in terms of better temporal resolution, cost and computational efficiency. The fNIRS offers good spatial resolution and noise immunity against electromagnetic interference, when it is compared to Electroencephalography (EEG) technique.

[1] Department of Electrical Engineering, Pakistan Institute of Engineering and Applied Sciences (PIEAS), Nilore, Islamabad, Pakistan. Email: [a]shahidnazir@pieas.edu.pk, [b]haroon@pieas.edu.pk, [c]m.turi21@gmail.com, [d]bhagesh.c.maheshwari@ieee.org, [e]aqil@pieas.edu.pk (Corresponding Author).

Studies have been carried out to investigate for the possibility of a fast fNIRS response [6, 7]. The averaging of multiple trials besides burdening the computational cost limits the use especially for an online environment. In recent years, works have been reported for recursive estimation of brain activity as coefficients of a linear system by the RLSE algorithm [2-4], and by a Kalman Filter (KF) [3, 8]. These methodologies, having potential of being utilized for real-time applications, were validated online in a time sharing environment by a sequential processing strategy of each channel. The online results provide a fast (compared to an offline) solution to the experimental investigation. But the outcome of each sample of multichannel brain data cannot be ensured before the occurrence of the next multichannel sample. As a consequence, the delay between the occurrence of a sample and its processing grows. Accumulation in this latency drags the online processing away from the real-time results.  This is a swear issue for event related real-time studies, *e.g.*, BCI and BI for rehabilitation and gamming, etc. A real-time solution would be having the potential for complex, multichannel, and multitasking BCI studies [9,10].

The RLSE is used as adaptive filter that is meant to estimate the model iteratively [2-4, 11]. It is advantageous over other iterative algorithms: E.g., fast convergence compared to least mean square filtering and computational efficiency over KF. The RLSE exhibits extremely fast convergence due its second-order nature and consistently performs to a higher accuracy than the other iterative algorithms [12].

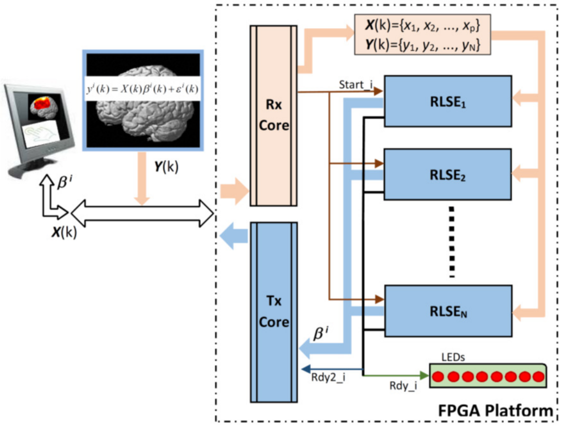
Reprogrammable hardware attracted a growing interest of researchers for the obtainment of enormous gain in speed and saving of energy besides its capability to be reprogrammed to desired application or functionality requirements after deployment [13-16]. The ease of providing parallel processing environment makes FPGA a high performance computing hardware for real-time processing of multichannel (multi-input, multi-output) systems [17], [18]. An FPGA required minimum expertise of a Hardware Description Languages (HDL; e.g., Verilog or VHDL) to program it. Third party software (like MatLab) introduced HDL code generator to convert the scientific coding into an HDL code for FPGA deployment [19]. Recently, the FPGA vendors presented an open computing language (OpenCL) [20] to get two-fold benefits: (i) A generalized coding environment for different vendors' FPGAs, and (ii) Abstract (high-level) coding environment for an FPGA. Although the programming of an FPGA by this high level language/software significantly reduces the development time compared to the traditional low-level languages (e.g., Verilog or VHDL) but results in significantly under-utilizing the computing capabilities of the device [21]. Thus, a trade-off exist between ease of coding and optimized coding. Researchers are focusing on investigating appropriate optimizers with the high-level software to reduce this trade-off [21].

The current study is aimed to provide a hardware solution for the estimation of brain activity, modelled as coefficients of a linear recursive model, in real-time. For this purpose, the RLSE algorithm [2-4, 11] is implemented on a Field Programmable Gate Array (FPGA) platform. Its multiple cores are instantiated to process multiple channels’ data of a sample simultaneously. A communication core is implemented to receive the sampled data serially (with RS-232 protocol) and place it as global variables. Each RLSE core is signaled by the communication core once its liable data is received and tapped at the designated place. Upon completion of each RLSE process, the computed parameters are transmitted back in real-time for display purposes. The proposed FPGA based embedded environment provides the parallel processing of multiple channels in a sample well before the arrival of the new sample. Thus, ensures the actual real-time processing of brain signals. The computational efficiency of the proposed system is described with quantitative details. On the basis of which, the sampling frequency of the imaging modality is aided for real-time processing of multiple channels. The entire methodology is realized on an FPGA kit (Spartan-6 LX150T) and validated with an fNIRS dataset. A t-statistics is formed to signify the results. A t-score based interpolated brain activation map is draw over the range of the channels. The accuracy of the obtained results are compared with the previously reported offline and online results.

This paper is organized as follows: Materials and methods section provides the modelling and estimation theory of the activity parameters. It further provides the FPGA implementation and computational efficiency for the targeting estimation theory. The subsequent section covers the Results and Discussion of proposed embedded system realized as octal-core on an FPGA platform, which follows the concluding remarks.

## 2. MATERIALS AND METHODS

Block diagram of the proposed RLSE-FPGA embedded system is shown in Fig. 1. The brain signals, modeled as the coefficients of a linear model, are measured in real-time and received into the platform along with the modeling regressors. The platform stores the data sample to a shared memory and inform the RLSE cores to start the estimation task. Upon completion, the RLSE cores intimate the transmitter to send the estimated parameters for display purpose. The descriptive detail of each block is provided in the following subsections.



$y^i(k) = X(k)\beta^i(k) + \varepsilon^i(k)$
$\boldsymbol{Y}$(k)
$\beta^i$
$\boldsymbol{X}$(k)
Rx Core
Tx Core
$\boldsymbol{X}(k)=\{x_1, x_2, \ldots, x_p\}$
$\boldsymbol{Y}(k)=\{y_1, y_2, \ldots, y_N\}$
Start_i
RLSE₁
RLSE₂
RLSE_N
$\beta^i$
Rdy2_i
Rdy_i
LEDs
FPGA Platform

Fig. 1: Schematic Diagram of Fnirs-RLSE Brain Activity Estimation for a Single Channel

### 2.1: Brain Activity Model

A linear brain activation model with coefficient representing the brain activity is considered as [2-4]

$$y^{i}(k) = x_1(k)\beta_1^i(k) + x_2(k)\beta_2^i(k) \cdots + x_p(k)\beta_p^i(k) + \varepsilon^i(k), \quad (1)$$

where k is discrete sampling time instant, $y^{i}(k)$ is the brain signal acquired from i-th channel ($i = 1,2,\ldots,N$) at k sampling instant, $x_1(k)$, $x_2(k)$, …, $x_p(k)$ are the k-th samples of p regressors, $\beta_1^i(k), \beta_2^i(k), \ldots, \beta_p^i(k)$ are the coefficients of corresponding regressors representing their strength in channel i at k sampling instance, and $\varepsilon^i$ (k) is the Gaussian noise with zero mean. Equation (1) can be rewritten in vector form as

$$y^{i}(k) = \mathbf{X}^T(k)\boldsymbol{\beta}^i(k) + \varepsilon^i(k). \quad (2)$$

The estimation method and its deploying hardware will decide the accuracy, precession, and speed of the modelled activity parameters.

### 2.2. RLSE Estimation of Activity Parameters

For each value of input and output at discrete time instant, RLSE recursively estimates the coefficients of an adaptive filter. In our case, the RLSE estimates the optimal values of $\boldsymbol{\beta}$(k) of a measured channel by minimizing the following cost-function,

$$V^i(\beta, k) = \frac{1}{2}\lambda^{k-j}\sum_{j=1}^{k}[e^i(j)]^2, \quad (3)$$

where $e^i$ is the estimation error and $\lambda$ is the forgetting factor whose value can be varied from 1 to 0 to rely on recent measurements more than the previous ones. The estimation error at time instance k is given by,

$$e^i(k) = y^i(k) - \mathbf{X}(k)^T\boldsymbol{\beta}^i(k-1). \quad (4)$$

Solving for minimizing the cost-function in (3), new set of equations are obtained as in [2-4, 11]

$$\mathbf{K}(k) = \mathbf{P}(k-1)\mathbf{X}(k)(\lambda + \mathbf{X}^T(k)\mathbf{P}(k-1)\mathbf{X}(k))^{-1}, \quad (5)$$

$$\boldsymbol{\beta}^i(k) = \boldsymbol{\beta}^i(k-1) + \mathbf{K}(k)e^i(k), \quad (6)$$

$$\mathbf{P}(k) = \{\mathbf{P}(k-1) - \mathbf{K}(k)\mathbf{X}^T(k)\mathbf{P}(k-1)\}/\lambda, \quad (7)$$

where $\mathbf{K}(k) \in \mathbb{R}^p$ is known as weighting vector and $\mathbf{P}(k) \in \mathbb{R}^{p\times p}$ is the recursive inverse of the input covariance matrix. The above mentioned RLSE equations (4)-(7) recursively estimate the optimal values of $\boldsymbol{\beta}$(k) for a single channel. For multichannel estimation, multiple RLSE processes are needed to workout independently, as illustrated in Fig. 1. Thus, it becomes the responsibility of the implementing hardware to provide independent processing, precise estimation, and fast (without latency: processing of the last sample before the arrival of the next sample) environment.

## 3. PROPOSED HARDWARE

Field programmable gate arrays are reprogrammable semiconductor devices based on configurable logic blocks. These logic blocks can be programmed in any configuration to perform desired arithmetic and logical operations through programmable interconnects. Nowadays, the Matlab based high level coding is advantageous for user prospective [19]. But the Matlab converted verilog/VHDL code is generalized in nature and is not optimal [21]. Such implementation uses more resources and introduces computational complexities as compare to the same algorithm implemented with low level Verilog/VHDL coding. Unlike microcontrollers and digital signal processors where the execution of the code is performed sequentially amongst the channels' estimation, FPGA provides the concurrent and order independent execution of different core instances. This approach drastically reduces the computational time.

The lower (near to machine) level coding of programmable interconnects of an FPGA is not easy especially for a real-time interfacing with high density data computations. But the FPGA platform is preferred owing to its concurrent, order independent, and fast performance. Furthermore, the coding based re-configurability is advantageous for further advancement of the design in future. A block diagram of the proposed FPGA based system is shown in Fig. 1. The fast processing speed of the FPGA along with the low level programing methodology reduces the computation time of an RLSE core. Multiple RLSE cores process multiple channels simultaneously and elevate the computational efficiency.

## 4. IMPLEMENTATION OF EMBEDDED RLSE FPGA

This section describes the hardware considerations of FPGA platform for the targeting blocks: a serial receiver core, N RLSE cores, and a serial transmitter core. The communication interface is defined for two single precision 32 bit data inputs $X(k) \in \Re^{p}$ and $y(k) \in \Re^{1\times N}$ by RS232_RXD, and an output $\beta(k) \in \Re^{p\times N}$ by RS232_TXD. Whereas RS232_RXD and RS232_TXD provides single bit serial receive and transmit interfaces, respectively, based on standard RS-232 protocol. The communication is facilitated by RESET and CLK as controlling inputs while LED[7:0] as status indicators. To make a serial interface compatible with universal asynchronous receiver transmitter (UART) protocol, a CLKX is opted for a desired baud-rate of 115200 bps. The receiver CLK16X is defined 16 times faster to avoid the bit error of the receiving data. Receiver module uar_top receives N channels' sampled dataset $y(k) \in \Re^{1\times N}$ followed by the regression vector $X(k) \in \Re^{p}$ and generates start signals $\mathrm{rls\_starti}$ (for $i = 1$ to N). Upon receiving the start signals, the RLSE cores initiate the processing to compute activity parameters. Finally, these cores generate $\mathrm{rdy2\_i}$ (for $i = 1$ to N) signals for the transmitter core uat_top to transmit the computed parameters $\beta(k) \in \Re^{p\times N}$ for real-time presentation. The $\mathrm{rdy\_i}$ (for $i = 1$ to N) signals, connected with LEDs, are used as test points to verify RLSE cores' computations. A typical realization of the proposed system for eight channel (N=8) is provided in Fig. 2, only selected elements are shown. An RS-232 based UART module is instantiated in top level module including two cores: (i) uar_top to receive the data inputs, and (ii) uat_top to transmit back the estimated parameters. The received data is stored in a memory, not shown in Fig. 2 for simplicity of illustration, and shared to all RLSE cores deployed in FPGA. Though a single RLSE core is shown for simplicity, the complete port list for multiple RLSE cores, working simultaneously, can easily be seen from Fig. 2.

## 5. COMPUTATIONAL EFFICIENCY

The computational efficiency of the proposed embedded system is assessed for a real-time sample. For a baud-rate of 115200bps, the bit communication time is 8.6805 μs and the transmission time for a single precision parameter (4 bytes × 10 bits/bytes = 40 bits) is 0.347 ms. A receiving sample of the dataset consists of p parameters of regression vector X(k) and one measured (HbO/HbR) parameter each for N channels. Whereas, transmitting sample consists of p estimated parameters, $\beta_1(k), \beta_2(k), \ldots, \beta_p(k)$, each for N channels. Thus, total time consumed in communicating data of N channels is summarized in Table 1.

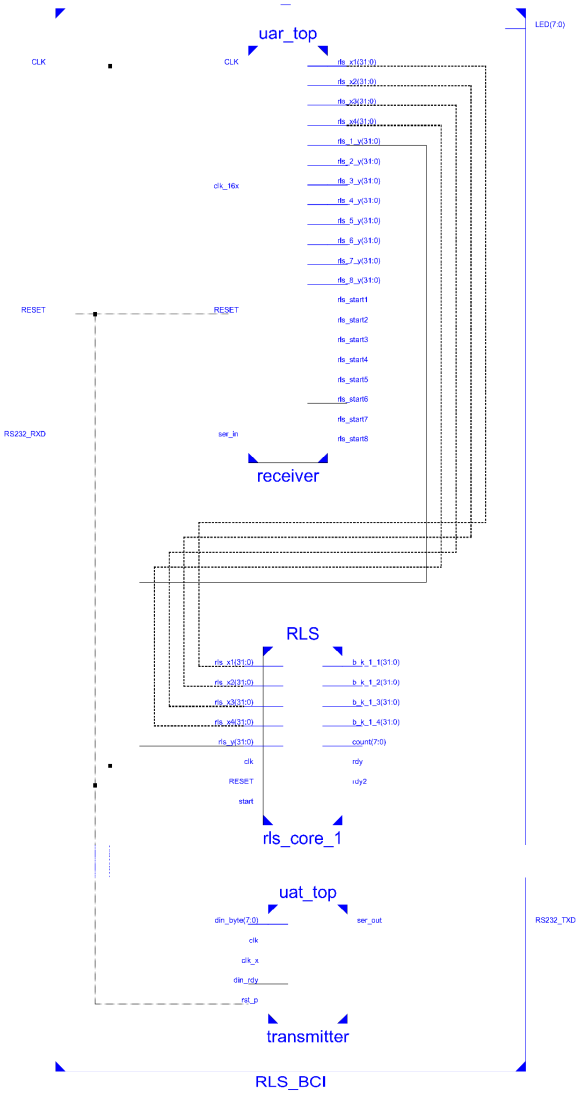


Fig. 2: RTL Schematic of the fNIRS-RLSE Embedded System Implemented in FPGA (Selected Elements Shown)

Table 1: Communication Time Summary With Different Number of Channels Implemented on FPGA with p = 4

| Number of RLSE Cores | Communication Time (ms) |
|---|---|
| N-Core | $[p + (p+1)N] \times 0.347$ |
| Single core | 3.123 |
| Dual-core | 4.858 |
| Quad-core | 8.328 |
| Octal-core | 15.268 |

## 6. FNIRS DATASET FOR VALIDATION

The validation of the proposed methodology is carried out by utilizing the block-design finger-tapping data made available by Ye *et al.* [22]. The fNIRS signals were measured from the left motor cortex at 24 locations, Fig. 3 shows the channels' configuration. The behavioral protocol consisted of an initial 42 sec for signal equilibrium followed by ten repetitions of 21 sec right-index-finger-tapping alternated with 30 sec rest periods. The Oxymon MK III, Artinis instrument was used to acquire the fNIRS data with a sampling rate of 9.75Hz (sample time of 102.6 msec). The availability of the complete dataset provides controllability of the imitated sampling rate to verify the speed of the computations of the system under test.

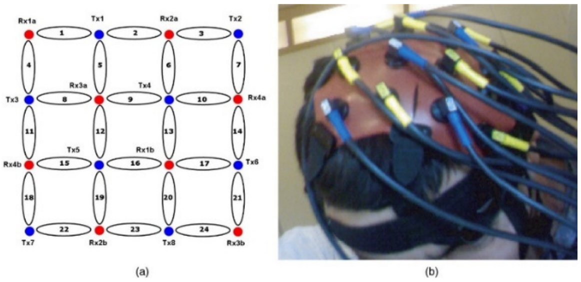


Fig. 3: Configuration of 24 Channels Covering the Left Motor Cortex for Right-Index-Finger-Tapping Task [22]

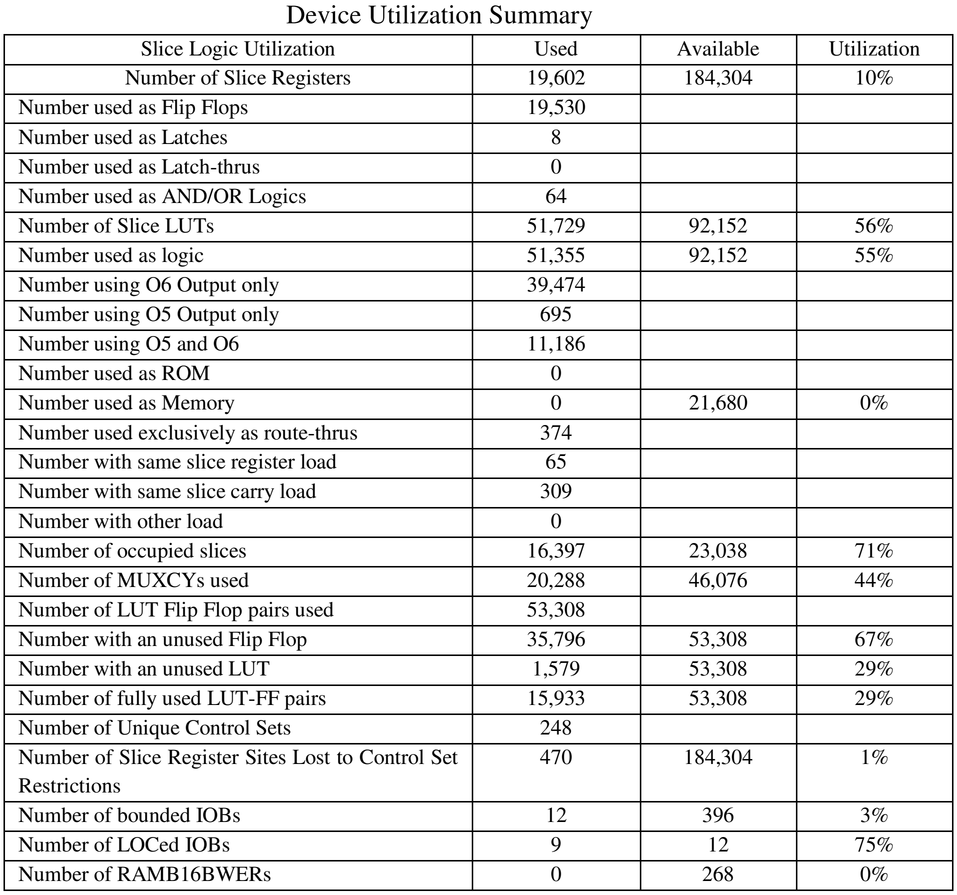
## 7. RESULTS AND DISCUSSION

For convenient comparison with the previously reported results, four regressor functions ($p = 4$) have been used [2, 3]. To model brain activity for a cognitive task, $x_1$ is taken to be the Hemodynamic Response Function (HRF) of the experiment with $x_2$ and $x_3$ are the delay and dispersion derivatives of $x_1$, respectively, while $x_4$ is taken as unity to counter the offset.

### 7.1. Typical Realization of FPGA-RLSE System

The validation is carried out by realization the proposed FPGA-RLSE system on Spartan-6 LX150T FPGA development platform for eight channels (N=8). The eight channels are concurrently processed by eight RLSE cores at the available CLK speed of 100 MHz. Top level RTL schematic of the selected elements is shown in Fig. 2. At the beginning, the receiver core receives the regression vector X(k) which would be common to all the channels for model driven approach. Then receiver core receives the NIRS measurement y(k) for channel 1 and generates rls_start1 while receiving the measurements of channel 2. This enables RLSE core 1 to start its processing promptly. Similarly, each of the eight RLSE cores is interrupted by its respective start signal once its dataset is being completed. Likewise, when the first RLSE core finishes its processing, it generates the ready signals: (i) rdy2_1 for transmitter core to send the computed parameters serially, and (ii) rdy_1 to indicate the task completion on LED0. Similarly, the other RLSE cores will respond to the transmitter as soon as their parameters are ready to be transferred. This approach reduces the computation time further even with a serial interface.

The stated hardware aspects described above are programmed on the FPGA platform by Verilog HDL. The device utilization summary of the platform is described in Fig. 4 as obtained in ISE Design Suit 14.2. The implemented system processes eight channels concurrently in 1.32 μs. It is worth mentioning that more than eight RLSE cores will take the same processing time because they will be working in parallel.



Device Utilization Summary

| Slice Logic Utilization | Used | Available | Utilization |
|---|---|---|---|
| Number of Slice Registers | 19,602 | 184,304 | 10% |
| Number used as Flip Flops | 19,530 | | |
| Number used as Latches | 8 | | |
| Number used as Latch-thrus | 0 | | |
| Number used as AND/OR Logics | 64 | | |
| Number of Slice LUTs | 51,729 | 92,152 | 56% |
| Number used as logic | 51,355 | 92,152 | 55% |
| Number using O6 Output only | 39,474 | | |
| Number using O5 Output only | 695 | | |
| Number using O5 and O6 | 11,186 | | |
| Number used as ROM | 0 | | |
| Number used as Memory | 0 | 21,680 | 0% |
| Number used exclusively as route-thrus | 374 | | |
| Number with same slice register load | 65 | | |
| Number with same slice carry load | 309 | | |
| Number with other load | 0 | | |
| Number of occupied slices | 16,397 | 23,038 | 71% |
| Number of MUXCYs used | 20,288 | 46,076 | 44% |
| Number of LUT Flip Flop pairs used | 53,308 | | |
| Number with an unused Flip Flop | 35,796 | 53,308 | 67% |
| Number with an unused LUT | 1,579 | 53,308 | 29% |
| Number of fully used LUT-FF pairs | 15,933 | 53,308 | 29% |
| Number of Unique Control Sets | 248 | | |
| Number of Slice Register Sites Lost to Control Set Restrictions | 470 | 184,304 | 1% |
| Number of bounded IOBs | 12 | 396 | 3% |
| Number of LOCed IOBs | 9 | 12 | 75% |
| Number of RAMB16BWERs | 0 | 268 | 0% |

Fig. 4: Device Utilization Summary of Spartan-6 Lx150t for the fNIRS-RLSE System Implemented for 8 Channels

### 7.2 Real-Time fNIRS Estimation

A User-friendly graphical User Interface (GUI), shown in Fig. 5(a) is made on PC to effectively interface the fNIRS data in real-time with the proposed RLSE-FPGA system. Intially the Connect/Disconnect option checks the availability of the system for connection or disconnect the link. Once available, the

Start/Stop button initiates/inhibits the real-time process by sending/stopping the data. The data is sent sample-by-sample with a predefined sampling rate. The Iteration_Number indicates the number of the under processing sample. After reception in PC, the estimated brain activity parameters, $\boldsymbol{\beta}(\mathrm{k})$, of each channel is saved in the local drive besides plotting the channels' HRF coefficients $\beta_1^i(\mathrm{k})$ (for $\mathrm{i} = 1$ to 8) only along with the estimated outputs for quick interpretation of the results. The HRF regressor is plotted alongside for reference. The results of two distinguished channels are plotted in Fig. 5(b-c) for elaboration.

An interpolated *t*-statistics based brain activation map is drawn, illustrated in Fig. 6, by using the methodology of Aqil *et al.* [2,3]. Although the map is drawn over the range of 24 measuring channels for comparative analysis but the t-scores of only first 8 processed channels are utilized by keeping the remaining channels at non-active values. The brain-mapping template (left lateral view) was depicted using the open-source software NIRS-SPM that was made available by Ye *et al.* [22]. It is apparent from Fig. 5-6 that the brain activity parameters obtained in real-time are (i) in accordance with the offline results in Ye *et al.* [22] and (ii) consistent with the online results presented in Aqil *et al.* [2] with the same dataset (Comparative analysis may be carried out with other FPGA implementations related to the field: No such implementation within the targeted BCI field). Same results in comparison with a non-real-time but precision environment proves the obtainment of desired accuracy of the manually implemented data format and their handling (mathematical operations and communication). The proposed FPGA-RLSE system offers beneficial BCI and BI applications in real-time including prompt medical diagnostics and therapeutics.

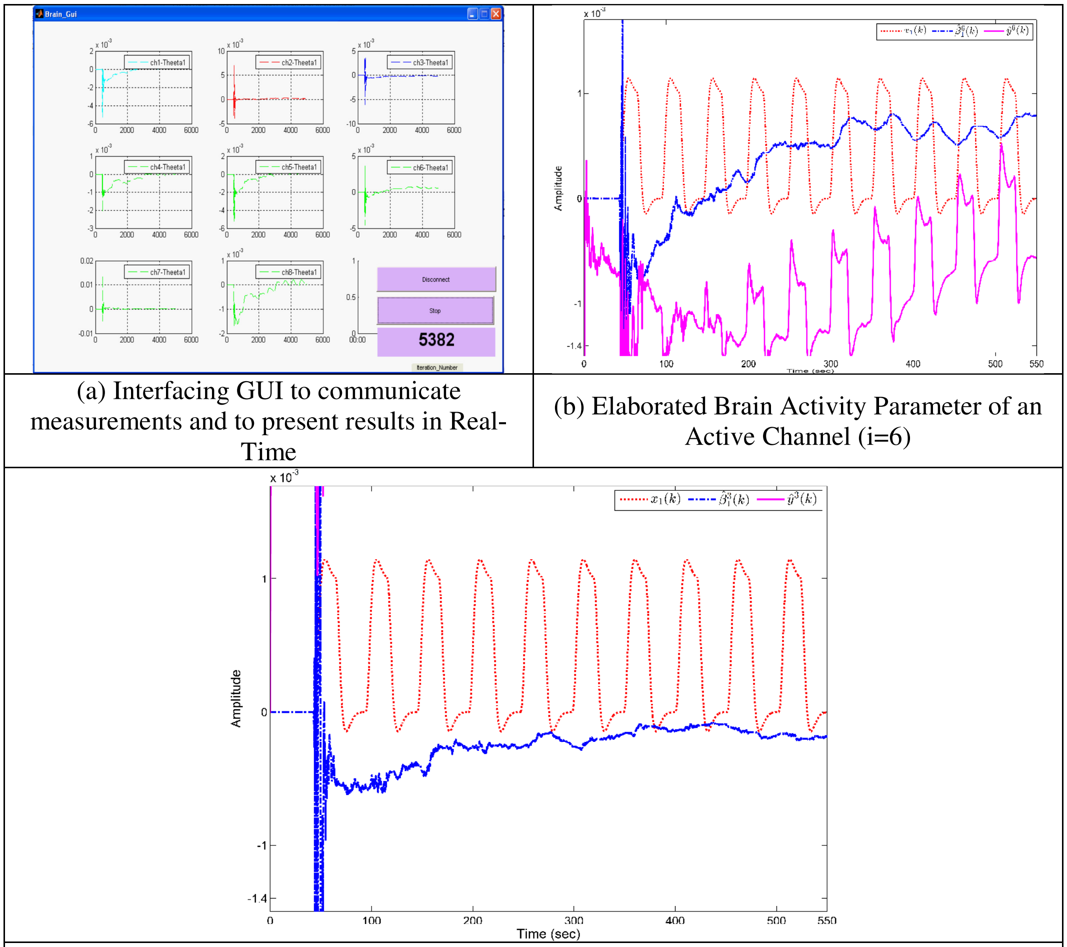
### 7.3. Computational Cost of Octal Core FPGA-RLSE

Considering the bit size of the parameters for octal core realization, the twelve parameters were received in the FPGA in 4.164 msec. The activity parameters were estimated by the octal RLSE cores concurrently in 1.32 μs. In return, the four activity parameters each for eight channels (4 × 8 = 32 parameters) were transmitted from the FPGA in 11.104 msec. Thus, the total time consumed to process a sample of 8 channels was 15.269 msec, which is much lower than the sampling time of a typical fNIRS datasets (in the order of 100 ms). Although, the proposed embedded system has faster sample processing rate than the usual sampling rate of an fNIRS modality, the sample processing rate can be made much faster by reducing the communication time by switching from serial (RS-232) to fast protocols (e.g., USB, Ethernet etc.). It is worth mentioning that the processing time of the proposed fNIRS-RLSE realization is very small as compared to the time consumed in communicating the parameters. The processing time still can be reduced further by realizing the proposed system on a fast clocking FPGA platform.

Currently, the FPGA platform transmits the estimated brain activity parameters to a PC for storage in a file besides real-time plotting. The real-time visualization of the results can be facilitated in FPGA by implementing a video graphic adapter (VGA) or high definition multimedia interface (HDMI) core on FPGA. Once introduced, the VGA/HDMI core can further be advanced to display brain activity on an anatomical brain template for effective brain imaging applications.

## 11. CONCLUSIONS

An embedded RLSE-FPGA system is provided for real-time brain activity estimation. Brain activity of multiple channels, coefficients of independent linear models, are estimated by concurrent RLSE cores implemented in an FPGA platform. The interfacing of the RLSE cores for (i) measuring and modeling inputs, and (ii) estimated activity outputs are facilitated by UART interface and FPGA memory. A 32-bit floating point core is deployed to provide computational accuracy. The real-time processing is ensured by assessing the computational cost of the proposed RLSE-FPGA system. The demonstration is carried out by an octal core realization at Spartan-6 LX150T platform validated by real fNIRS dataset. The proposed system has potential real-time BCI and BI applications.



(a) Interfacing GUI to communicate measurements and to present results in Real-Time

(b) Elaborated Brain Activity Parameter of an Active Channel (i=6)

(c) Elaborated Brain Activity Parameter of an Inactive Channel (i=3)

Fig. 5: Results of the proposed RLSE-FPGA Brain Imaging System:

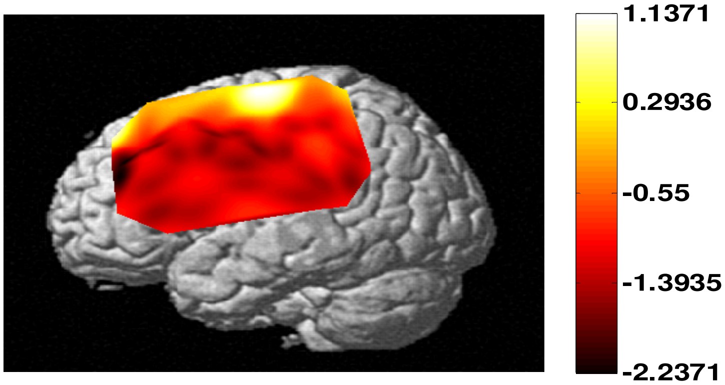


Fig. 6: Brain Activation Map over the range of First 8 Channels Drawn Covering the Left Motor Cortex for Right-Index-Finger-Tapping Task

## ACKNOWLEDGMENT

A part of this project is supported by National ICT R&D fund by the Ministry of Information Technology, Government of Pakistan.

## REFERENCES

[1] Cope M., Delpy D.T., "System for Long-Term Measurement of Cerebral Blood and Tissue

Oxygenation on Newborn-Infants by Near Infra-Red Trans-Illumination", *Medical & Biological Engineering and Computing*, Vol. 26, No.3, pp. 289-294, 1988.

[2] Aqil M., Hong K.-S., Jeong M.-Y., Ge, S.S., "Cortical Brain Imaging by Adaptive Filtering of NIRS Signals", *Neuroscience Letters*, Vol. 514, No. 1, pp. 35–40, 2012.

[3] Aqil M., Hong K.-S., Jeong M.-Y., Ge, S.S., "Detection of Event-Related Hemodynamic Response to Neuroactivation by Dynamic Modeling of Brain Activity", *NeuroImage*, Vol. 63, No. 1, pp. 553-568, 2012.

[4] Aqil M., Jeong M.-Y., "Critical Bounds on Noise and SNR for Robust Estimation of Real-Time Brain Activity from Functional Near Infra-Red Spectroscopy", *NeuroImage*, Vol. 176, No. 1, pp. 321-353, 2018.

[5] Zhang F., Roeyers H., "Exploring Brain Functions in Autism Spectrum Disorder: A Systematic Review on Functional Near-Infrared Spectroscopy (fNIRS) Studies", *International Journal of Psychophysiology*, Vol. 137, No. 1, pp. 41-53, 2019.

[6] Cui, X., Bray S., Reiss A.L., "Speeded Near Infrared Spectroscopy (NIRS) Response Detection", *PLoS One*, Vol. 5, No. 11, pp. e15474, 2010.

[7] Plichta M.M., Heinzel S., Ehlis A.C., Pauli P., Fallgatter A,J., "Model-Based Analysis of Rapid Event-Related Functional Near-Infrared Spectroscopy (NIRS) Data: A Parametric Validation Study", *NeuroImage*, Vol. 35, No. 2, pp. 625-634, 2007.

[8] Abdelnour A. F., Huppert T., "Real-Time Imaging of Human Brain Function by Near-Infrared Spectroscopy Using an Adaptive General Linear Model", *NeuroImage*, Vol. 46, No. 1, pp. 133-143, 2009.

[9] Tak S., Uga M., Flandin G.: "Sensor Space Group Analysis for fNIRS Data", *Journal of Neuroscience Methods*, Vol. 264, No. 1, pp. 103-112, 2016.

[10] Metzger F.G., Ehlis A.-C., Haeussinger F. B., Schneeweiss P., Hudak J., Fallgatter A.J., Schneider S., "Functional Brain Imaging of Walking While Talking - An fNIRS Study", *Neuroscience*, Vol. 343, No. 1, pp. 85-93, 2017.

[11] Haykin S., "Adaptive Filter Theory", Prentice Hall, New Jersey, 4th edn, 2002.

[12] Milanesi M., Martini N., Vanello N., Positano V., Santarelli M.F., Paradiso R., De Rossi D., Landini L., "Multichannel Techniques for Motion Artifacts Removal from Electrocardiographic Signals", *Proceedings of 28th Annual IEEE International Conference Engineering in Medicine and Biology Society*, pp. 3391-3394, 2006.

[13] Pathan A., Balal R., Memon T.D., Memon S., "Analysis of Booth Multiplier based Conventional and Short Word Length FIR Filter", *Mehran University Research Journal of Engineering and Technology*, Vol. 37, No. 3, pp. 595-602, 2018.

[14] Ali S.A., Asif R., Hina S., Fatima Z., "Cloud Based Remote FPGA Lab Platform: An Application of Internet of Things", *Mehran University Research Journal of Engineering and Technology*, Vol. 37, No. 4, pp. 535-544, 2018.

[15] Zahra Q., Malik M.S.A., Batool N., "An Efficient Computer-Aided Diagnosis System for the Analysis of DICOM Volumetric Images", *Mehran University Research Journal of Engineering and Technology*, Vol. 38, No. 3, pp. 835-850, 2019.

[16] Noronha D.H., Torquato M.F., Fernandes, M.A.C., "A parallel implementation of sequential minimal optimization on FPGA", *Microprocessors and Microsystems*, Vol. 69, No. 1, pp. 138-151, 2019.

[17] Torquato M.F., Fernandes M.A.A., "High-performance parallel implementation of genetic algorithm on FPGA", *Circuits, Systems, and Signal Processing*, Vol. 38, No. 9, pp. 1-26, 2019.

[18] Da Costa A.L.X, Silva C.A.D., Torquato M.F., Fernandes M.A.C., "Parallel implementation of particle swarm optimization on FPGA", *IEEE Transactions on Circuits and Systems: Express Briefs*. doi: 10.1109/TCSII.2019.2895343, 2019.

[19] Parmar C.A., Ramanadham B., Darji A.D., "FPGA Implementation of Hardware Efficient Adaptive Filter Robust to Impulsive Noise", *IET Computers & Digital Techniques*, Vol. 11, No. 3, pp. 107-116, 2017.

[20] Khronos Group,The OpenCL Specification, 2010.

[21] Shata K., Elteir M.K., EL-Zoghabi A.A., "Optimized implementation of OpenCL kernels on FPGAs", *Journal of Systems Architecture*, Vol. 97, No. 1, pp. 491-505, 2019.

[22] Ye J.C., Tak S.H., Jang K.E., "NIRS-SPM: Statistical Parametric Mapping for Near-Infrared Spectroscopy", *NeuroImage*, Vol. 44, No. 2, pp. 428-447, 2009.