IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 June 2023, accepted 1 July 2023, date of publication 7 July 2023, date of current version 14 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3293422

RESEARCH ARTICLE

Reconfigurable VLSI Design Architecture for Deep Learning Established Forelimb and Hindlimb Gesture Recognition for Rehabilitation Application

ANAGHA NIMBEKAR[1], Y. V. SAI DINESH[1], ARVIND GAUTAM[1], VIDHUMOULI HUNSIGIDA[2], APPA RAO NALI[2], AND AMIT ACHARYYA[1], (Member, IEEE)
[1]Indian Institute of Technology Hyderabad, Hyderabad 502285, India
[2]Xilinx, Hyderabad 500081, India

Corresponding author: Amit Acharyya (amit_acharyya@iith.ac.in)

This work was supported in part by the Ministry of Electronics and Information Technology (Government of India), and in part by AMD (Xilinx).

ABSTRACT Recently, a large body of work has revealed a very accurate deep learning established movement categorization algorithmic technique for assistive technology applications. However, minimum seriousness is given to its relative hardware execution. But direct mapping these algorithms onto battery-operated devices is a point of care, edge devices would not yield optimum results necessitating an algorithm-architectural holistic design methodology. In the current study, we offered a method for creating reconfigurable VLSI architectures for deep learning established gesture recognition for rehabilitative technology applications. The ZYNQ ultrascale + MPSoC zcu102 FPGA was used to implement the design. The on-chip power requirements for LoCoMo-Net and MyoNet on FPGA are 3.5 and 5 Watts, respectively. Furthermore, gesture recognition on the FPGA takes both networks at 1.876 ms and 61.988 ms, respectively. The accuracy comparison of the most advanced networks is carried out on the CPU, GPU, FPGA, and ASIC platforms. The suggested architecture was further synthesized utilizing GF 40-nm technology, which resulted in a 2.5× improvement in performance in terms of speed and a 12–15× decrease in power consumption compared to FPGA in 2.046 *mm*2 of space and 300.8423 mW of power at 1.21 V.

INDEX TERMS Convolutional neural networks (CNN), long-term convolutional networks (LRCN), long short-term memory (LSTM), LoCoMo-Net, MyoNet.

I. INTRODUCTION

Majority of the forelimb (hand) and hindlimb (leg) amputations occur due to sports and accidents. A finger amputation is a simple procedure that involves working with nerves, tendons, and skin to restore fine motor ability while limiting hand use [1]. In accidents such as anterior cruciate ligaments (ACL), sciatic nerve, and meniscus injuries an individual can get a knee injury responsible for the dysfunction. Therefore making the subject's life easy and self-dependent on assistive technology solutions plays a major role.

The associate editor coordinating the review of this manuscript and approving it for publication was Ahsan Khandoker.

In the field of assistive technology applications deep learning has made a promising improvement. In this context, many attempts have been made. Park and Lee in [2] proposed a Convolutional Neural Network to predict hand movement using sEMG-based data with an accuracy of 90%. Soman et al. in [3] proposed a solution using a twin Support Vector Machine (SVM) to recognize 15 wrist and finger flexion movements from the muscles of the forelimb achieving an average accuracy of 82%. Ulysse et al. [4], executed transfer learning for hand gesture recognition with the slow fusion model of CNN and reported 97.8% of accuracy. Atzori et al. [5] introduced a CNN model for 50 movement classification and reporting 55% of average accuracy. Work

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

by Zhai et al. [6], introduced a self-modifying sEMG pattern recognition technology using CNN reporting accuracy of 78.71%.

Similarly, for hindlimb, Zhang et al. [7] developed an ARI (auto-regressive integrated) network to infer hinge joint angle. In [8] Kianifar et al. utilized an inertial sensor to evaluate knee joint angle. Kianifar et al. also proposed the possibility of knee injury in the course of a one-leg squat. Huang et al. [9] introduced a recurrent neural network to estimate the real-time hinge joint movement fusing of sEMG and inertial data. Lui et al. [10] utilized sEMG time-domain data to estimate hinge joint angle. Chen et al. [11] proposed an EMG signal-based artificial neural network to predict hinge joint angle and also used parameters such as positive pulse width, negative pulse width, and positive pulse amplitude called as functional electrical stimulation (FES) parameters. In our recently published research by Gautam et al., two deep learning established networks, LoCoMo-Net [12] for forelimb amputation and MyoNet [13] for hindlimb amputation, were proposed. LoCoMo-Net and Myonet have recorded accuracies of 93.6% and 95.2%, respectively. In the view of above it can be seen that deep learning has made a tremendous improvement in designing an algorithm for assistive technology applications. However, owing to the computationally intensive nature of DNN algorithms, implemented onto the wearable device running on a battery backup or harvested energy, may not be efficient with regards to low power consumption and low area overhead. For real-time performance, these algorithms need to be mapped onto their corresponding hardware platform. Application-Specific Integrated Circuit (ASIC) is a device that is designed for a specific application. As the device is application specific and not a general-purpose device hence consumes low power. Hence a low-complexity VLSI architecture design without compromising algorithmic accuracy will play a major role. Having separate chips/systems for forelimb and hindlimb amputation costs more for designing two different chips/systems. It also increases the design time and effort for having a separate architecture for the forelimb and hindlimb. Hence a single hardware architecture accommodating both the DNN algorithm to serve hindlimb and forelimb respectively in a reconfigurable fashion would even reduce power and area consumption further. To the best of our knowledge, there have only been a few attempts to develop such a unique reconfigurable hardware design, which is essential to the work that drives this research. Recently our preliminary results in this context have been published in [14] however in this article in addition to those preliminary results the following contributions are made:

Our main contributions are highlighted as follows:

- proposal of reconfigurable, low complexity VLSI architecture design for deep learning established forelimb and hindlimb gesture recognition for rehabilitation applications (LoCoMo-Net [12] and MyoNet [13]) also, it's detailed and elaborated architectural description.
- succeeding implementation of the proposed architecture on Zynq Ultra-scale+ FPGA and a thorough comparison with state-of-the-art GPU platform affirming low complexity and power efficiency (4-4.6× of reduction in power), real-time (1.876ms for LoCoMo-Net and 61.988ms for MyoNet) implementation.
- subsequent design space exploration of the proposed architecture has been done in the Application Specific Integrated Circuit domain and the gate level netlist has been synthesized using GF 40-nm technology utilizing 2.046 mm^2 of the area and 300.8423 mW of power at 1.21 V. Power and area results are reported which to the best of our knowledge is first of its kind.

The remainder of the paper is structured as follows: section II provides the theoretical background, section III outlines our suggested reconfigurable hardware architectural design, section IV provides experimental findings and analysis, and Section V closes with the scope of its future development.

II. THEORETICAL BACKGROUND

A. LoCoMo-NET

A Low-Complex Deep Learning Framework for sEMG-Based Hand Movement Recognition for Prosthetic Control [12] was created to differentiate between forelimb tasks for amputee patients. It is a binary classifier with an average accuracy of 93.6%. It classifies among 15 various forelimb tasks of finger and wrist movements with the rest position of the forelimb. LoCoMo-Net consists of two convolutional layers with one fully-connected layer at last as shown in Fig 1. The input layer provides 250 ms windows of 1-D single-channel sEMG sensor data. LoCoMo-Net has two 1D convolutional layers with seven 5 × 1 filters and a stride of one. Following convolution, both layers use a ReLU activation function (Rectified Linear Unit) followed by a 2 × 1 maximum pooling operation. After that, the Fully Connected layer is executed at the last layer. Because LoCoMo-Net is a binary classifier, each task is divided into two categories: tasks that are categorized (labeled class A) and tasks that are not classified (labeled class B).

B. MyoNet

Myo-Net is one of the transfer-learning established Long-term Recurrent Convolutional Network (LRCN) motion recognition networks with an accuracy of 95.2% for hindlimb movement categorization and joint angle prediction. MyoNet distinguishes three hindlimb movements: class A (sitting), class B (standing), and class C. (walking). It also forecasts the knee joint's angle. MyoNet uses convolutional layers to obtain features from given data and Long Short Term Memory with a fully connected layer for hinge angle prediction. At the output, the softmax loss function is generated for forecasting the feature over time steps for classification [13]. From the input layer, four-channel 1-D surface-Electromyography sensor data having windows of 256 ms is fed to the convolution layer. Before the ReLU (Rectified Linear Unit activation) operation and 4 × 1 maximum pooling function, the

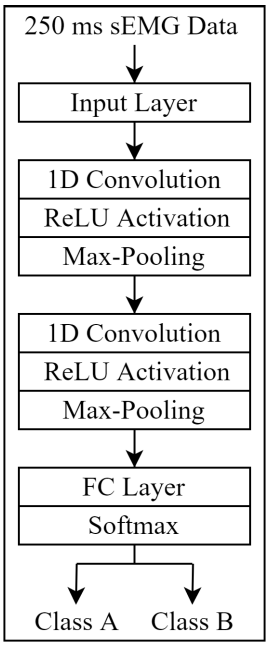


FIGURE 1. LoCoMo-Net network architecture.

convolution layer conducts 1D convolution with 20 kernels each of 11 × 1 size and stride of 1. The movement categorization and joint angle prediction block are then computed in parallel. For classification among classes A, B, and C, the movement classification block computes a dense layer with the softmax function. Two LSTM layers make up the joint angle prediction block. The first LSTM layer was 32 memory units in size, while the second LSTM layer was 64 memory units in size. Fig 2 depicts the MyoNet network topology.

C. CONVOLUTIONAL NEURAL NETWORKS

The exceptional performance offered by CNNs made them an unavoidable choice for supervised learning tasks and computer vision applications [15]. Convolutional Neural Networks (CNNs) are symmetrical to traditional Artificial Neural Networks (ANN) and composed of interconnected neurons that are fine-tuned through learning. Every neuron performs an operation (viz. scalar product followed by a non-linear function) on the input received on the basis of countless Artificial Neural Networks [16]. Starting with the input image vectors to the category score as the final output, the whole neural network will be expressing a single incisive score function (the weight) [16]. The final layer of the convolutional neural network contains the loss function to evaluate the accuracy of the classes and all of the tuning methods of ANN still apply [16]. Till now people have used CNN for image-processing applications. In this paper, we propose a reconfigurable hardware architecture for assistive technology applications based on sEMG sensor data. LoCoMo-Net [12] and MyoNet [13] are based on Convolutional Neural Network (CNN). LoCoMo-Net consists of two convolutional layers after that a fully connected layer and softmax function. MyoNet consists of 2 convolutional layers and 2 LSTM layers.

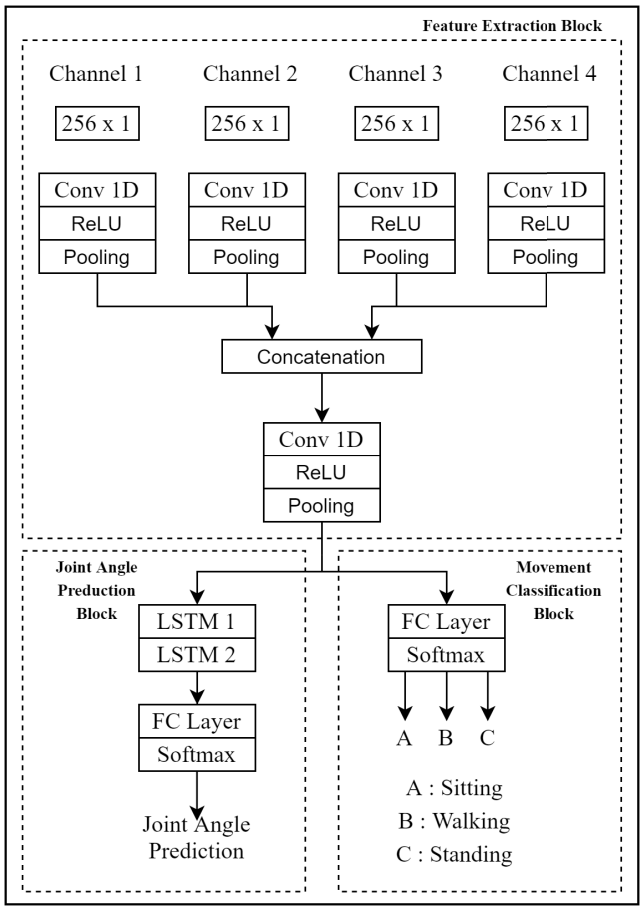


FIGURE 2. MyoNet network architecture.

D. LONG-TERM RECURRENT CONVOLUTIONAL NETWORK (LRCN)

MyoNet [13] is based on LRCN (an architecture of DNN), shown in Fig. 2 comprised of LSTM-based RNN preceded by a CNN. CNNs are used to take out the complex character from the incoming input data which is followed by a fully connected layer and softmax to evaluate the loss at the output and to predict the task movement classification. A transfer-learning-based approach is utilized for the movement classification, wherein the trained sEMG data is directly passed to the dense layer [13]. Features of the sEMG data are already learned while performing the angle prediction task, this avoids redoing the feature extraction again for movement classification [13]. The output feature maps are obtained through the extracted features of input data which are described by kernel weights of convolution layers. This is succeeded by an activation function (for introducing the non-linearity for learning complex features), then the pooling layer followed by a densely connected layer which gives the classified output [17].

LSTM units, on the other hand, are a form of recurrent neural network (RNN) [18] that are good at capturing long-term temporal relationships while avoiding optimization challenges. LSTM units are typically composed of a hidden

state triggered by a nonlinear function, that employs a learned gating operation to enable the condition to propagate without alteration, be reset, or be updated [13]. LSTM's have recently obtained respectable results in biological domain [19], voice identification [20], language conversion [21], [22], and computer-vision applications [23].

E. HARDWARE-SOFTWARE CO-DESIGN

It involves hardware and software interaction to complete a specific task using concurrent and coordinated design [24]. The system is composed of hardware and software that is specifically designed for it and is mapped to a central processing unit (CPU). The co-design-dependent method benefits from the hardware's power, speed, and parallelism, as well as the software's flexibility and modularity for constraint optimization [25]. Extensive research in hardware-software co-design with a focus on partition strategies [25] has led to various developments in architectures, multiprocessing, multithreading, and multi-core environments. System-on-chip (SoC) [26], which unifies all the elements of computation and communication on a single chip, has just emerged as a new paradigm for co-design. The Zynq-7000 SoC FPGA family, the technology employed in this method, has the ability to construct the entire hardware-software system on a single platform, [26]. By offering a one-chip solution, these FPGAs do away with the need for individual integrated circuits (IC) for the CPU and hardware. The co-design method uses a variety of soft-core processors that are readily available, ready to use, and completely configurable to the needs of the application processors in addition to the Zynq family with an on-chip ARM processor, enabling quick prototype and design implementation.

III. PROPOSED METHODOLOGY

To achieve the proposed reconfigurable VLSI architecture for deep learning networks, we divided the approach into two stages. First, we designed dedicated VLSI architectures for LoCoMo-Net (described in subsection A) and MyoNet (described in subsection B). Second, we figured out the overlapping and separate operations of both networks. Based on these operations we propose reconfigurable VLSI architecture with common hardware units for overlapping operations and dedicated hardware for separate operations as described in subsection C. First the VLSI architecture for LoCoMo-Net.

A. PROPOSED VLSI ARCHITECTURE FOR LoCoMo-NET

The VLSI architecture for LoCoMo-Net as shown in Fig 3 is described as follows: first, from the processor, the trained model and the test data are loaded to primary storage (Block RAM). For the first convolution, layer input is selected from the primary storage, and for the next layer from the local storage. Switching between the storage is done with a 2:1 multiplexer based on the layers. After receiving the data, the convolution block starts the MAC computation. The partial products are passed to bias addition. Bias weights are also stored in primary memory. Following bias addition,

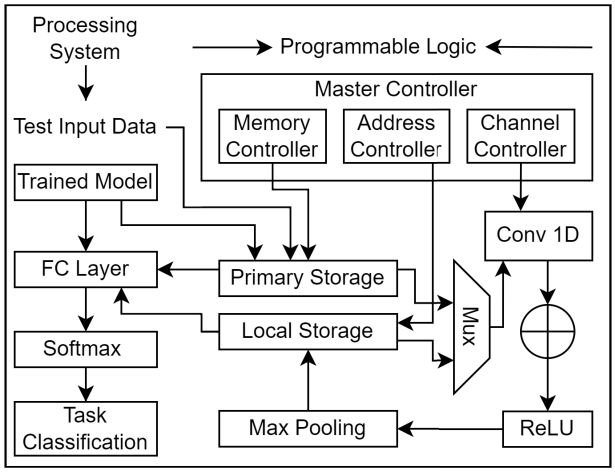


FIGURE 3. Proposed VLSI architecture for LoCoMo-Net implemented on FPGA.

ReLU activation is computed, followed by max pooling. LoCoMo-Net [12] employs 2×1 max-pooling, necessitating the usage of a two-input comparator. The address controller is used to hold two input values in registers for 2×1 pooling. The comparator receives these two values. The result of the comparison is saved in the block RAM.

The results of max pooling of the first layer are stored in the local storage. For the second layer inputs are selected from the local storage and similar computations are performed in the first layer. Once we get the max pooling results for the second layer, data is passed to a fully connected layer. After a fully connected softmax function is executed. In the end, we get the final result whether the task is classified or not.

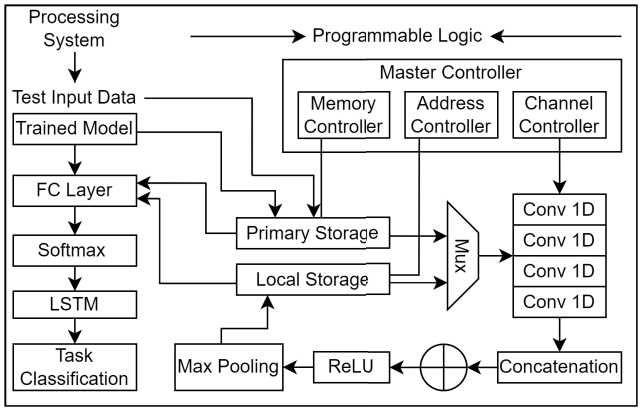


FIGURE 4. Proposed VLSI architecture for MyoNet implemented on FPGA.

B. PROPOSED VLSI ARCHITECTURE FOR MYO-NET

The VLSI architecture for MyoNet as shown in Fig 4 is described as follows: firstly the trained weights and test data are loaded first to the primary storage. As MyoNet has 4 channel sEMG data hence 4 convolution modules are utilized (each module for each channel). The four-channel convolved

output is concatenated and converted to a single dimension. After concatenation bias addition after ReLU operation and maximum pooling is performed. For Myonet two 2 input comparators are required as it has a pooling size of 4×1. With the help of the address controller, 4 input data samples are saved in registers. These 4 data samples are fed to 2 comparators in the first iteration. The two comparison outputs from the first iteration are supplied to one comparator again in the second iteration, while the other is deactivated. The output is then stored in the block RAM.

For task classification, a fully connected layer is performed with the softmax function. In the end, we have 3 task classifications as shown in Fig 4. Parallel to the FC layer LSTM layer is executed on the processor. After LSTM again a fully connected layer is performed to get the joint angle prediction.

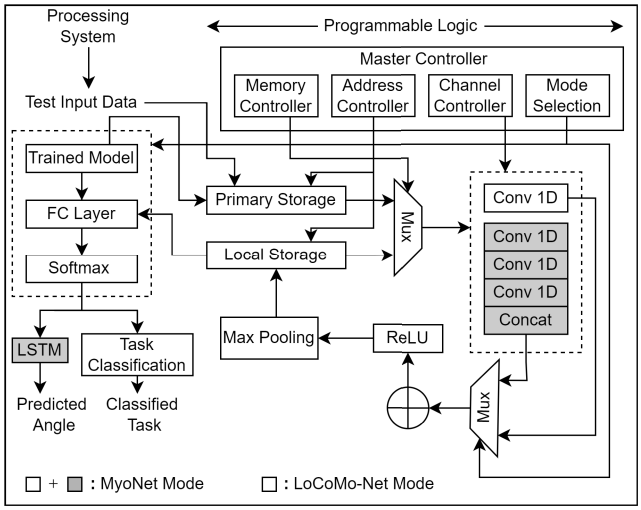


FIGURE 5. Proposed re-configurable, low-Complexity VLSI design architecture for deep learning established forelimb and hindlimb gesture recognition for rehabilitative technologies (LoCoMo-Net [12] and MyoNet [13]).

C. PROPOSED RECONFIGURABLE VLSI ARCHITECTURE

As shown in Fig1 and Fig 2 that both networks have few common operations. The first two layers of MyoNet and LoCoMo-Net are convolutional layers that perform 1D convolution after ReLU and maximum pooling operations. And MyoNet [13] has two LSTM layers whereas LoCoMo-Net [12] does not use LSTM. Hence the common operation is kept in the programmable logic part and the varying operations are executed on the processor (processing system). Both the networks have fully connected layers but with a different number of classes hence we executed fully connected layer operation on the processor. The main idea of partitioning is to keep the static operations on the programmable logic side and the dynamic operations to be executed at the processor end. The proposed reconfigurable machine learning hardware architecture for assistive technology applications is as shown in Fig 5. The flow of architecture is as follows. First, from the processor, the trained model and the test input data are stored in the primary storage. Depending on the mode selection architecture is reconfigured for the input-trained network. The input from the primary storage is passed to the convolutions. The output of a convolution is passed to further blocks for concatenation, bias addition after that ReLU activation, and max pooling. The output of max pooling is stored in the local storage. Next data from local storage is passed to the FC (Fully Connected) layer. The FC layer also receives the trained weights from the processor and starts computing the FC layer operations. FC layer output is then passed to the softmax activation function which generates the probabilities for the classification classes. At last, we get the final output class at task classification.

- **Master Controller:** The master controller plays an important role. It consists of controller logic for memory, channel, address, and mode selection with the network's information like the number of layers in the network, filter size, stride, pooling size, and the number of input channels. Depending on the mode selected, the master controller will configure/reconfigure the entire architecture.
- **Memory Controller** Advanced memory layout and data flow are critical for optimal hardware implementation. The memory should be huge as CNN includes millions of parameters. To preserve accuracy, all operations are conducted with a 16-bit floating point. As a result, all parameters are recorded with 16-bit precision. It costs a lot of RAM to store all the parameters. DDR memory is a viable option for such memory needs, although it uses more power. However, in the proposed design, DDR memory is used as the main memory whereas Block RAMs are used to store interim findings.
- **Channel controllers** The channel controller manages the channel computation based on the input data. Only one convolution block is active since LoCoMo-Net uses single-channel data. Since Myo-Net uses four-channel sEMG data, four convolution blocks are also enabled for this algorithm.
- **Address controller** Address controller controls the flow of Writing data from the processor and reading it back to the processor from memory for the next processing.
- **Mode selection** Mode selection determines whether architecture will operate in LoCoMo-Net [12] or Myo-Net [13] mode. It's a one-bit signal.
- **Convolution:** In any network accelerator design convolution operation governs the computation complexity and therefore engages the most computations. In the proposed architecture there are four identical convolution modules each module having one multiply-and-accumulate (MAC) unit. Also, each module is re-configurable for 1D convolution. Fig 6 shows block evel representation of the convolution module. An internal BRAM memory is utilized for storing the intermediate results between input data convolved with filter coefficients. Due to a shortage of memory space, the hardware is reused to compute the MAC (multiplication

and addition) operations at the cost of a bit more latency to output the feature map.
This results in a reduction in memory usage.

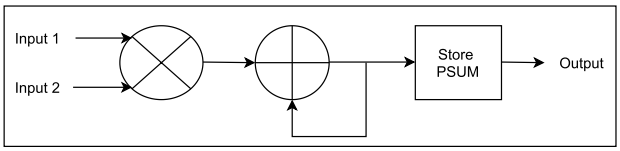


FIGURE 6. Block level representation of convolution operation implemented on the proposed architecture.

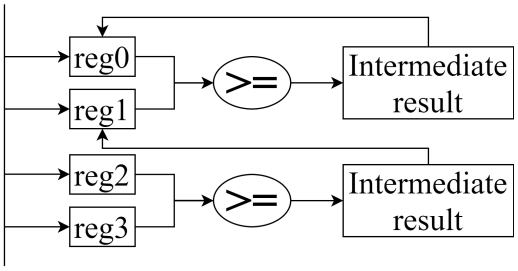


FIGURE 7. Comparator based max-pooling logic to work in LoCoMo-Net and MyoNet Mode.

- **Rectified Linear Activation Function:** The convolution procedure is immediately followed by the ReLU Activation function. The negative data samples are set to zero by this activation function, while non-zero data samples are sent on exactly as they are. The activation function receives the output of the convolution process. The max-pooling algorithm is used to handle the output of the ReLU activation function.
- **Max-Pooling:** To decrease computing complexity, the maximum pooling layer is used. It also improves efficiency by down-sampling the incoming data samples. It returns the greatest data sample value of the chosen region. The output of the activation function is initially stored in registers. Input for Max-Pooling is chosen from these registers. The address controller is used to choose activation outputs. As illustrated in Fig 7, the proposed architecture consists of two comparators to obtain the max-pooled output. If the architecture works in LoCoMo-Net mode then only one comparator is used. If the architecture is configured to work in MyoNet mode then both comparators are used.
- **Fully Connected Layer:** Neurons which are in Fully connected layers will have complete associations with the past layers. It is utilized to arrange data into different classes in a neural net. We use the Softmax function for the classification of labels based on the probabilities of each input component [27]. The multiplication and accumulation (MAC) operations are performed by the fully connected layer. The maximum pooling outputs are sent into the FC layer. FC-layer outputs the accumulated values based on the number of classes. The probabilities are generated using the softmax loss function after the aggregated outputs are prepared. The categorization result is provided by the fully connected layer based on the highest likelihood.

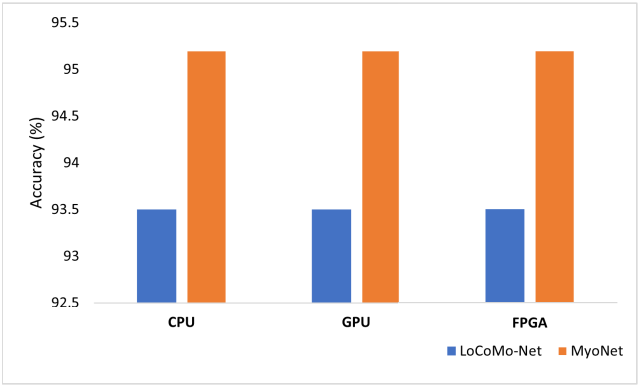


FIGURE 8. Accuracy comparison of LoCoMo-Net and MyoNet on CPU, GPU, and FPGA platform.

- **Concatenation:** Concatenation is a function that allows grouping items together in a sequence. Branching is a term used for a few networks, such as [13], in which numerous convolution processes are conducted in parallel. Pooling output must be concatenated before being supplied as input to the next layer. However, three activation and three maximum pooling procedures will be required in this process, at the expense of greater resource use and increased power consumption.
Instead of following the above-mentioned scenario, right after convolution, we concatenate the outputs of four parallel convolution operations, and the concatenated output is fed for activation and max-pooling function. By following the second approach the proposed architecture is able to compute the operation with fewer resources utilization and hence comparatively less power is consumed.

IV. RESULTS AND ANALYSIS

Table 1 shows the lower limb and upper limb movements to be classified by Myonet [13] LoCoMo-Net [12] respectively. Myonet classifies the lower limb task (walking, sitting, and standing). However, LoCoMo-Net classifies among the tasks performed by the upper limb (movement of each finger individually, together movement of combination of two figures, wrist flexion, wrist extension, all figure movement together, hand grasp, pinch grip, and rest position). Fig 8 presents the accuracy comparison of the state-of-the-art algorithms on CPU, GPU, and FPGA platforms. It is observed that there is no reduction in accuracy. Subsequently, the proposed reconfigurable architecture is executed on the ZYNQ ultra-scale+ MPSoC ZCU102 FPGA. Fig 9 shows the dataset collection of the tasks mentioned in Table 1. The entire duration for dataset collection is 50 seconds session of 4 trials. First, the subject is asked to stay in a relaxed state for 5 seconds. Then the subject

TABLE 1. Task to be classified by MyoNet and LoCoMo-Net.

Sr No	Task Description
MyoNet	
Task 1	Walking
Task 2	Sitting
Task 3	Standing
LoCoMo-Net	
Task 1	Little finger motion
Task 2	Ring finger motion
Task 3	Middle finger motion
Task 4	Index finger motion
Task 5	Thumb finger motion
Task 6	Concurrent motion of little and ring fingers
Task 7	Concurrent motion of ring and middle fingers
Task 8	Concurrent motion of little and index fingers
Task 9	Concurrent motion of index and middle finger
Task 10	Wrist flexion
Task 11	Wrist extension
Task 12	Wrist flexion and extension
Task 13	All fingers motion together
Task 14	Hand grasp
Task 15	Pinch grip
Task 16	Rest

is given a time of 2.5 seconds to get ready mentally to perform the task. Later the subject will perform the task for 5 seconds. After finishing the task a break of 5 seconds is given in which the subject has to relax and get ready to perform the next task. The flow for testing the proposed VLSI architecture is as follows. From the processor, we selected the mode in which we wanted to work for example LoCoMo-Net mode which means mode selection is set to zero. After selecting the mode, the architecture is configured to work in LoCoMo-Net mode (mode selection: 0). Now based on the input signal task selected, the architecture will perform the computations, and last it will give the output task classification. Similarly, if the mode is selected as 1 then the architecture will work in Myo-Net mode and gives the output task classification along with join angle prediction. The FPGA is operated at a frequency of 100 MHz. FPGA Resource utilization for LoCoMo-Net and MyoNet is displayed in Table 2. As mentioned in [37], an accumulator comes after a multiplier in the DSP slice. Both multiply and multiply-accumulate operations require at least three pipeline registers to operate at maximum speed. From Fig 1 and Fig 2 it can be observed that LoCoMo-Net processes single-channel data requiring only one multiply and accumulator unit compared to MyoNet. Therefore the proposed architecture utilizes more resources when working in MyoNet mode, however, in LoCoMo-Net mode, the resource utilization is comparatively less.

The computational time is equal to 1.876 ms for LoCoMo-Net and 61.988 ms for MyoNet. The calculation time listed above is within the allowed limit of real-time prosthetic control, which is 300 ms, as stated in the literature [34]. We conducted a comparison of power utilization between the suggested methodology-based architecture execution on FPGA and the direct mapping of algorithms on GPU (Jetson TX2) to demonstrate the low complexity nature of the proposed architecture (zcu102). We used Nvidia Jetson tx2 GPU.The Jetson TX2 is an embedded system-on-module (SoM) developed by NVIDIA. It is designed specifically for artificial intelligence (AI) applications and edge computing. The TX2 module incorporates a powerful GPU (graphics processing unit) and a CPU (central processing unit) along with other essential components, all integrated onto a single board. Table 3 shows the power utilization of the proposed reconfigurable architecture working in LoCoMo-net and MyoNet mode on FPGA, GPU, and ASIC platforms. The proposed architecture consumes 3.6 Watts and 5 Watts of power on FPGA, 16 Watts and 23 Watts of power on GPU, and 258.44 mWatts and 264.88 mWatts of power on ASIC platform working on LoCoMo-Net and MyoNet mode respectively. Reconfigurable architecture has been proven to use 4-4.6 times less power than a GPU. The time consumed by LoCoMo-Net is 90 sec, 1.879 ms, and 17.58 ms therefore, the energy consumed is 1440 J, 0.67 J, and 0.0045 J on GPU, FPGA, and ASIC respectively. For MyoNet the time consumption is 120 sec, 61.988 ms and 50.78 ms energy consumption is 2760 J, 0.31 J, and 0.0134 J on GPU, FPGA, and ASIC respectively. For FPGAs, both energy and power are important considerations. Energy efficiency is crucial to optimize the overall energy consumption while managing power consumption is essential to ensure reliable and stable FPGA operation. For FPGA the power is evaluated from the Xilinx Vivado Tool. For GPU, as we used Nvidia Jetson tx2 GPU, there are power rails available we measured the power rail reading and got the power consumption. For ASIC, we reported the frontend results from Synopsys Design Compiler tool.

All the computations are performed on 16-bit format, also the architecture is parameterized. On the GF 40nm, the suggested Re-configurable, Low-Complexity VLSI Architecture architecture for deep learning-based forelimb and hindlimb movement categorization assistive technologies uses 2.046 mm^2 of the area and 300.8423 mW of power.

TABLE 2. FPGA resource utilization of proposed reconfigurable VLSI architecture working in LoCoMo-Net and MyoNet Mode. (FF: Flip-Flops, LUTRAM: Lookup table random access memory, LUT: Lookup Table, BUFG: Global clock buffer, DSP: Digital signal processing, IO: Input/Output, BRAM: Block RAM).

Resource	Available	Utilized	Percentage
MyoNet			
FF	5481602	20493	3.74
LUTRAM	144000	64	0.04
LUT	274080	9339	3.41
BUFG	404	1	0.25
DSP	2520	8	0.32
IO	328	104	31.71
BRAM	912	1.050	1.15
LoCoMo-Net			
FF	548160	3212	0.59
LUTRAM	144000	203	0.14
LUT	274080	2940	1.07
BUFG	404	1	0.25
BRAM	912	0.50	0.05

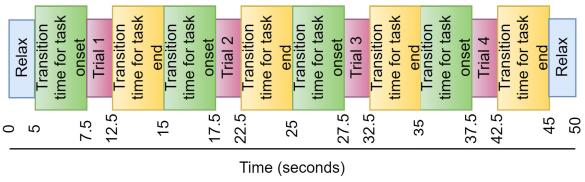


FIGURE 9. Experimental setup for dataset collection of sEMG data.

A. CHALLENGES

Firstly as LoCoMo-Net has a 5 × 1 filter size and MyoNet has an 11 × 1 filter size. Challenges are faced in designing a convolution module supporting various convolution sizes a MAC unit is designed to have 1 multiplier and 1 adder which can be utilized many times as per the requirement. Second a reconfigurable max-pooling logic to support 2 × 1 and 4 × 1 max-pooling operations which are used by LoCoMo-Net and MyoNet respectively. The third control logic is to enable 4 convolution blocks in parallel for MyoNet mode and disable 3 convolution blocks while the first convolution block will be enabled to work in LoCoMo-Net mode.

B. DISCUSSION

The strength of the proposed work is that to date lot of work is done for improving the accuracy of the deep neural network. However minimum efforts are made for designing equivalent VLSI architecture for the same to have a chip/system solution. Having a single chip/system solution that can be used by the amputated patients for performing daily life activities easily. This will make the amputated patients self-dependent. However, the proposed architecture can work in LoCoMo-Net and MyoNet modes sequentially.

TABLE 3. Power utilization of proposed architecture on FPGA, Nvidia GPU and ASIC.

Network	GPU	FPGA	ASIC
LoCoMo-Net	16 Watt	3.6 Watt	258.44 mWatt
MyoNet	23 Watt	5 Watt	264.88 mWatt

V. CONCLUSION

This work offers a reconfigurable, low-complexity VLSI architecture for assistive technology applications that uses deep learning to categorize forelimb and hindlimb movement. The Zynq Ultra-scale+ FPGA has successfully implemented the specified design. Less than 20% of the total FPGA resources are consumed in this paper's analysis of FPGA resource usage. The suggested design consumes 4-4.6× less power on FPGA than on GPU. The design is also synthesized utilizing GF 40-nm technology, yielding 2.046 *mm*2 of the area and 300.8423 mW of power at 1.21 V. The Future scope of the proposed work is to do the front-end and back-end design and get a device. The application of the proposed work is that this device can be used by individuals having hindlimb (upper limb) and/or forelimb (lower limb) amputation to make their daily life activities easier. This device can work in LoCoMo-Net and MyoNet modes to help the amputated individual. As the proposed architecture is reconfigurable, individuals having both hindlimb (upper limb) and forelimb (lower limb) amputation need not wear two separate devices.

REFERENCES

[1] *Limb Loss Rehabilitation, Prosthetics and Orthotics | Johns Hopkins Physical Medicine and Rehabilitation (hopkins-medicine.org)*. Accessed: May 16, 2020. [Online]. Available: https://www.hopkinsmedicine.org/physical_medicine_rehabilitation/services/programs/orthotics-amputee-rehab.html

[2] K. Park and S. Lee, "Movement intention decoding based on deep learning for multiuser myoelectric interfaces," in *Proc. 4th Int. Winter Conf. Brain-Comput. Interface (BCI)*, Feb. 2016, pp. 1–2.

[3] S. Soman, Jayadeva, S. Arjunan, and D. K. Kumar, "Improved sEMG signal classification using the twin SVM," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 4507–4512.

[4] U. Côté-Allard, C. L. Fall, A. Campeau-Lecours, C. Gosselin, F. Laviolette, and B. Gosselin, "Transfer learning for sEMG hand gestures recognition using convolutional neural networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 1663–1668.

[5] M. Atzori, M. Cognolato, and H. Müller, "Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands," *Frontiers Neurorobot.*, vol. 10, p. 9, Sep. 2016.

[6] X. Zhai, B. Jelfs, R. H. M. Chan, and C. Tin, "Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network," *Frontiers Neurosci.*, vol. 11, p. 379, Jul. 2017.

[7] W. Zhang, M. Tomizuka, and J. Bae, "Time series prediction of knee joint movement and its application to a network-based rehabilitation system," in *Proc. Amer. Control Conf.*, Jun. 2014, pp. 4810–4815.

[8] R. Kianifar, A. Lee, S. Raina, and D. Kulic, "Automated assessment of dynamic knee valgus and risk of knee injury during the single leg squat," *IEEE J. Transl. Eng. Health Med.*, vol. 5, pp. 1–13, 2017.

[9] Y. Huang, Z. He, Y. Liu, R. Yang, X. Zhang, G. Cheng, J. Yi, J. P. Ferreira, and T. Liu, "Real-time intended knee joint motion prediction by deep-recurrent neural networks," *IEEE Sensors J.*, vol. 19, no. 23, pp. 11503–11509, Dec. 2019, doi: 10.1109/JSEN.2019.2933603.

[10] Q. Liu, L. Ma, Q. Ai, K. Chen, and W. Meng, "Knee joint angle prediction based on muscle synergy theory and generalized regression neural network," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2018, pp. 28–32.

[11] Y. Chen, J. Hu, F. Zhang, P. Li, and Z. Hou, "EMG-based estimation of knee joint angle under functional electrical stimulation using an artificial neural network," in *Proc. 32nd Chin. Control Conf.*, Jul. 2013, pp. 4661–4665.

[12] A. Gautam, M. Panwar, A. Wankhede, S. P. Arjunan, G. R. Naik, A. Acharyya, and D. K. Kumar, "Locomo-Net: A low-complex deep learning framework for sEMG-based hand movement recognition for prosthetic control," *IEEE J. Transl. Eng. Health Med.*, vol. 8, pp. 1–12, 2020.

[13] A. Gautam, M. Panwar, D. Biswas, and A. Acharyya, "MyoNet: A transfer-learning-based LRCN for lower limb movement recognition and knee joint angle prediction for remote monitoring of rehabilitation progress from sEMG," *IEEE J. Transl. Eng. Health Med.*, vol. 8, pp. 1–10, 2020.

[14] A. Nimbekar, Y. V. S. Dinesh, A. Gautam, V. Hunsigida, A. R. Nali, and A. Acharyya, "VLSI architecture design methodology for deep learning based upper limb and lower limb movement classification for rehabilitation application," in *Proc. IEEE 13th Latin Amer. Symp. Circuits Syst. (LASCAS)*, Mar. 2022, pp. 1–4.

[15] J. Wu, "Introduction to convolutional neural networks," Nat. Key Lab Novel Softw. Technol., Nanjing Univ., Nanjing, China, Tech. Rep., 2017, p. 495, vol. 5, no. 23. [Online]. Available: https://api.semanticscholar.org/CorpusID:36074296?utm_source=wikipedia

[16] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*.

[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[19] D. Biswas, L. Everson, M. Liu, M. Panwar, B. Verhoef, S. Patki, C. H. Kim, A. Acharyya, C. Van Hoof, M. Konijnenburg, and N. Van Helleputte, "CorNET: Deep learning framework for PPG-based heart rate estimation and biometric identification in ambulant environment," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 2, pp. 282–291, Apr. 2019.

[20] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[21] I. V. O. Sutskever and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[22] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proc. 8th Workshop Syntax, Semantics Struct. Stat. Transl. (SSST)*, 2014, pp. 103–111.

[23] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.

[24] P. R. Schaumont, "The nature of hardware and software," in *A Practical Introduction to Hardware/Software Codesign*, 2nd ed. New York, NY, USA: Springer, 2013, pp. 3–30.

[25] J. Teich, "Hardware/software codesign: The past, the present, and predicting the future," *Proc. IEEE*, vol. 100, pp. 1411–1430, May 2012, doi: 10.1109/JPROC.2011.2182009.

[26] M. Santarini, "Zynq-7000 EPP sets stage for new era of innovations," *Xcell J.*, vol. 75, pp. 8–13, May 2011.

[27] *The Softmax Function, Neural Net Outputs as Probabilities, and Ensemble Classifiers*. Accessed: May 16, 2020. [Online]. Available: https://towardsdatascience.com/the-softmax-function-neural-net-outputs-as-probabilities-andensemble-classifiers-9bd94d75932

[28] AI & Deep Learning Solutions to Optimize Your Data Pipelines | NetApp. *Artificial Intelligence Solutions: Let Your Data Flow*. Accessed: Jul. 15, 2020. [Online]. Available: https://www.netapp.com/artificial-intelligence/

[29] Y.-H. Chen, T. Krishna, J. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Jan. 2016, pp. 262–263.

[30] S. Chakradhar, M. Sankaradas, V. Jakkula, and S. Cadambi, "A dynamically configurable coprocessor for convolutional neural networks," in *Proc. 37th Annu. Int. Symp. Comput. Archit.*, Jun. 2010, pp. 247–257.

[31] H. Nakahara, T. Fujii, and S. Sato, "A fully connected layer elimination for a binarizec convolutional neural network on an FPGA," in *Proc. 27th Int. Conf. Field Program. Log. Appl. (FPL)*, Sep. 2017, pp. 1–4.

[32] J. Zhang and J. Li, "Improving the performance of OpenCL-based FPGA accelerator for convolutional neural network," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, Feb. 2017, pp. 25–34.

[33] D. T. Nguyen, T. N. Nguyen, H. Kim, and H. Lee, "A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 8, pp. 1861–1873, Aug. 2019.

[34] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 50, no. 7, pp. 848–854, Jul. 2003.

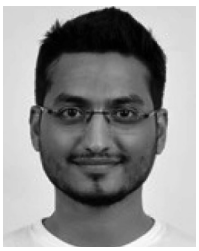
[35] K. Guo, L. Sui, J. Qiu, S. Yao, S. Han, Y. Wang, and H. Yang, "Angel-eye: A complete design flow for mapping CNN onto customized hardware," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2016, pp. 24–29.

[36] M. Panwar, A. Gautam, D. Biswas, and A. Acharyya, "PP-Net: A deep learning framework for PPG-based blood pressure and heart rate estimation," *IEEE Sensors J.*, vol. 20, no. 17, pp. 10000–10011, Sep. 2020.

[37] ug479_7Series_DSP48E1.pdf • Viewer • AMD Adaptive Computing Documentation Portal (xilinx.com). *7 Series DSP48E1 Slice*. Accessed: Jun. 2, 2023. [Online]. Available: https://docs.xilinx.com/v/u/en-US/ug479_7Series_DSP48E1

ANAGHA NIMBEKAR received the B.Tech. and M.Tech. degrees, in 2017 and 2019, respectively. She is currently pursuing the Ph.D. degree in microelectronics and VLSI with the Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Hyderabad, India. Her research interests include reconfigurable VLSI architecture designing, FPGA prototyping, and SoC design.

Y. V. SAI DINESH received the B.Tech. and M.Tech. degrees, in 2021 and 2022, respectively. He is currently pursuing the Ph.D. degree in microelectronics and VLSI with the Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Hyderabad, India. His research interests include SoC design and high speed interconnects for AI acceleration.



ARVIND GAUTAM received the Ph.D. degree in microelectronics and VLSI from the Indian Institute of Technology Hyderabad, India, in 2019. After Ph.D. degree, he has joined TSMC, Taiwan, as a Principal Engineer, from 2019 to 2022. In December 2022, he joined Synopsys, Munich, Germany, as an Application Engineer. He has authored several peer-reviewed journals and conferences. His research interests include low-power VLSI design, STA methodology for advanced sub-nanometer nodes, and 3DIC technologies.

VIDHUMOULI HUNSIGIDA received the bachelor's degree (Hons.) in electronics and communications from Osmania University. He got the Electronic Fellowship from DRDO EFC-10. He has been working in industry in various research domains, since 1995. Currently, he is the Director of Software Development with AMD. He has four patents.

APPA RAO NALI received the M.Tech. degree from IIT Kharagpur. He is currently the Director of Software Development Tools with AMD. He is having 22 years of experience in software development. He worked on PCB tools development, software development tools for AMD devices, and embedded driver development. His expertise is at system and multi-tier architecture design. He has two patents.

AMIT ACHARYYA (Member, IEEE) received the Ph.D. degree from the School of Electronics and Computer Science, University of Southampton, U.K., in 2011. He was a Scientist-B with Defence Research and Development Organization, from 2005 to 2007, a Research Fellow with the University of Southampton, in 2011, and an Assistant Professor (on contract position) with the Indian Institute of Guwahati, from 2011 to 2012. He is currently an Associate Professor with the Indian Institute of Technology (IIT) Hyderabad, Hyderabad, India. He has authored more than 80 international refereed journals and more than 90 international peer reviewed conferences and contributed towards six book chapters. His research interests include VLSI systems for resource-constrained applications, low power design techniques, machine learning hardware design, edge computing, healthcare technology, and chip-design targeting remote health monitoring, including cardiovascular diseases, diabetes, autism spectrum disorder, neurological disorder, orthopedically handicapped patients, accelerating cancer diagnostic procedures through hardware software co-design, signal processing algorithm and VLSI architectures, digital arithmetic, and hardware security. He is also handling several projects of Government of India, including Science and Engineering Board (SERB), Department of Science and Technology (DST), Ministry of Electronics and Information Technology (MEITY), and Defence Research and Development Organization (DRDO) apart from working in the private industry sponsored projects, including Xilinx Inc., USA.