Hardware Software Co-Design of Deep Learning Augmented Wireless Channel Estimation

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

M.Tech

Electronics and Communication Engineering

BY
Animesh Sharma

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI
NEW DELHI– 110020

December 19, 2022

Certificate

This is to certify that the thesis titled **"Design and performance analysis of Deep-Learning based channel estimation on System-On-Chip"** submitted by **Animesh Sharma** for the partial fulfillment of the requirements of *Master of Technology in Electronics and Communications Engineering* is a record of the bonafide work carried out by her under my guidance and supervision at Indraprastha Institute of Information Technology, Delhi.
This work has not been submitted anywhere else for the reward of any other degree.

Dr. Sumit J Darak **Date: December 19, 2022**
Associate Professor
Department Of Electronics and Communication
Indraprastha Institute of Information Technology, Delhi

Acknowledgement

I want to take this opportunity to express my sincere gratitude to the people who have supported me during my thesis work. Foremost, I would like to express my sincere gratitude to my advisors, Dr. Sumit J Darak and Syed Asrar Ul Haq, for their invaluable guidance, encouragement, and support throughout my thesis work. Their constructive feedback and regular work discussions helped me accomplish the task with great clarity.
Also, i would like to thank the research group at Algorithms to Architecture lab at IIITD for making the lab environment engaging and fun for everyone.

This work has been established in collaboration and support of IIT Indore's DR-ISHTI CPS foundation as CHANAKYA PG fellowship. I need to duly acknowledge their support throughout the project duration.

Last but not least, I would like to thank my family and friends for supporting me through challenging times and for their guidance in managing my thesis work efficiently.

Abstract

The feasibility study of deep learning (DL) approaches for reliable, flexible, and high throughput wireless physical layer (PHY) has received significant interest from academia and industry. In this direction, 3GPP has set an ambitious goal of introducing standards for intelligent and reconfigurable PHY by 2028. Channel estimation is one of the critical signal processing units of the wireless PHY, and recent works have shown the potential use-case of DL approaches to improve the performance of state-of-the-art statistical channel estimation approaches such as least-square (LS) and linear minimum mean square error (LMMSE). Existing DL-based channel estimation approaches have not yet been realized on system-on-chip (SoC). Our preliminary study shows that their complexity exceeds the complexity of the entire PHY. The high latency of DL is another concern. The work presented in this thesis aims to offer innovative solutions at the algorithm and architecture levels to address these challenges.

The first contribution is efficiently mapping LS, LMMSE and DL-based channel estimation approaches on heterogeneous SoC. Via hardware-software co-design and fixed point analysis, we compare the functional correctness, resource utilization, and execution time of existing architectures for a wide range of signal-to-noise ratios (SNR) and wireless channels. Specifically, we highlight the high complexity and latency of existing DL approaches. The second contribution of the thesis is to design a compute-efficient deep neural network (DNN) augmented LS-based channel estimation (LSDNN) algorithm and its efficient mapping on the SoC. We demonstrate substantial savings in complexity and latency without significant degradation in functional accuracy. Specifically, the proposed LSDNN approach offers 88-90% lower latency and 38-85% lower resources than recent DL-based channel estimation approaches. In addition, it offers 75% lower latency and 90-94% lower resource utilization than the LMMSE. The hardware IPs and demonstration on Zynq SoC offer opportunities for commercialization and a framework for verification of upcoming channel estimation algorithms on SoC.

Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

In wireless networks, the physical layer (PHY) at the transmitter transmits the data over the wireless radio channel, and PHY at the receiver receives and decodes the transmitted data. For reliable communication over a noise-fading channel in the presence of interference and hardware impairments, transmitter PHY performs channel coding and modulation on the data and inserts pilots, i.e., known data, at regular intervals over time and frequency. The receiver PHY performs channel estimation using the pilots and equalization on the received data. This is followed by data demodulation and channel decoding [3] [4]. Over the last two decades, various innovations in the algorithm and architecture have led to substantial improvement in the throughput, reliability, and latency of the wireless PHY [5] [6]. For instance, the throughput of the wireless networks has increased from 9.6 kilobits per second (kbps) in 1G to 10 Gigabit per second (Tbps) in 5G. Similarly, latency has been reduced from 1000 milliseconds (ms) in 1G to 1 ms in 5G. Such evolution of wireless PHY has enabled service operators to bring data-intensive multimedia and ultra-reliable low-latency services into reality [7].

Channel estimation and equalization is one of the most computationally complex tasks at the receiver PHY. With the introduction of new services ranging from data-intensive multimedia and ultra-reliable low latency to vehicular communication, channel estimation has become increasingly complex due to large bandwidth and high throughput requirements resulting in demand for fewer pilots and support for a wide range of wireless channels with different fading and mobility constraints. Along with reliable data reception, channel estimation enables the estimation of channel state information (CSI) at the receiver. The CSI is communicated by the receiver to the transmitter, and the transmitter uses it to select the PHY parameters for subsequent communication. The upcoming services are sensitive to CSI estimation accuracy. Conventionally, statistical least-square (LS) and linear minimum mean square esti-

mation (LMMSE) are widely used for channel estimation. The LS is popular due to its simple design and low latency. However, various studies have shown that the LS performs poorly and may not be suitable for next-generation applications and services. The LMMSE needs prior knowledge of wireless channel parameters and is more computationally complex than LS. It is not optimal for non-linear and non-stationary channels. Furthermore, the fixed-point realization of MMSE is challenging, resulting in a huge area, delay, and power overhead [1].

Recently, machine learning (ML) and deep learning (DL) have been extensively used in various applications. The feasibility study of DL approaches for reliable, flexible, and high throughput wireless physical layer (PHY) has received significant interest from academia and the wireless industry. This is because DL approaches can more accurately optimize non-linear signal degradation due to channel, interference, and hardware impairments than conventional approaches [2] [8]. Furthermore, DL approaches involve simple computations that can be parallelized compared to conventional techniques such as LMMSE which involves large-size matrix inversion. In this direction, 3GPP has set an ambitious goal of introducing standards for intelligent and reconfigurable PHY by 2028.

Channel estimation is one of the critical tasks of the wireless PHY, and recent works have shown the potential use-case of DL approaches to improve the performance of statistical channel estimation approaches. Existing DL-based channel estimation approaches have not yet been realized on system-on-chip (SoC), and our preliminary study shows that their complexity exceeds the complexity of the entire PHY. The high latency of DL is another concern. The work presented in this thesis aims to offer innovative solutions at the algorithmic and architectural levels to address these challenges.

1.2 Objectives and Contributions

The work presented in this thesis aims to design compute-efficient DL-based channel estimation for wireless PHY and efficiently map it on Zynq System-on-Chip (SoC) via hardware-software co-design. The main contributions are summarized below:

1. The first contribution is to study and efficiently map LS, LMMSE and existing DL-based channel estimation approaches on Zynq SoC. We study the functional accuracy, resource utilization, and latency of various architectures obtained via hardware-software co-design and word-length analysis.

2. The second contribution is to design a compute-efficient deep neural network (DNN) augmented LS-based channel estimation (LSDNN) and its efficient mapping on the Zynq SoC. We demonstrate substantial savings in complexity and latency without significant degradation in functional accuracy. Specifically, the proposed LSDNN approach offers 88-90% lower latency and 38-85% lower resources than recent DL-based channel estimation approaches. In addition, it offers 75% lower latency and 90-94% lower resource utilization than the LMMSE.

3. The hardware IPs and demonstration on Zynq SoC offer opportunities for commercialization and framework for verification of upcoming channel estimation algorithms on SoC.

1.3 Literature Survey

Numerous works have shown that traditional frameworks such as Detection-theory [9], Shannon theory [10], and Queuing theory [11]-based wireless physical layer suffer from accuracy loss because of the random nature and variability of wireless channels. Rather than just investigating approximate frameworks, the modern data-centric approaches along with their ability to solve difficult-to-model complications, provide a viable substitute for improving wireless-PHY robustness [12] [13]. In addition, there have been quite a few studies that demonstrate the superior performance of machine and deep learning (DL) based approaches in tackling challenges posed by complex wireless communication networks. The Deep Learning-based techniques allow extracting features from the data itself, removing the requirement of manually extracting features [14]. As a result, a slew of DL-based techniques for several PHY complications like spectrum sensing [15], localization [16], modulation classification [17] [18] direction-of-arrival estimation [19] [20], MIMO high-resolution channel feedback [21] [22]. The current frameworks struggle with scalability challenges due to the high cost of exhaustive searches and repetitive heuristic algorithms involved in large-scale heterogeneous networks [23] [24]. And data-centric techniques can prove to be more efficient in overcoming such scalability challenges [25] [26]. There are many issues like as potential use cases, feasible gain and complexity trade-offs, data-set availability, and standard compatibility implications, that must be addressed to make Deep Learning-aided intelligent and re-configurable wireless PHY a reality.

The deep learning-based PHY can either consist of a whole receiver and transmitter

omitted by a standalone DL architecture or have independent DL blocks for each sub-block or combinations of sub-blocks in PHY [18] [27] [28][[2]]. The main downside of replacing a transmitter and receiver completely with DL is that it won't provide access to intermediate outputs, creating it incompatible with current standards. In this thesis, we follow the second approach replacing each block or group of blocks of PHY with DL architecture without compromising on the compatibility with wireless standards. In this thesis, we focus on the channel estimation task in wireless PHY.

Communication systems based on rigid mathematical channel models struggle in complex scenarios because these models are unable to capture imperfections in practical channels. Thus, there has been a lot of recent study in the literature to find an ideal estimation approach. Since the channel matrix is a 2-D matrix, convolution neural networks (CNNs) based channel estimation was the first choice, and thus one of the earliest DL-based approaches used CNNs to improve the performance of the channel estimation. ChannelNet [1] consists of two consecutive CNN models, a super-resolution network (SRCNN [29]) followed by a denoising network (DnCNN [30]). The received reference pilot symbols are first LS estimated, then interpolated to whole frame dimensions, then passed through SRCNN, followed by DnCNN, to finally get the whole channel estimated. The estimation procedure can be seen in figure 1.1. ChannelNet consists of 23 convolution layers, which incurs huge computational complexity, large memory and high latency.

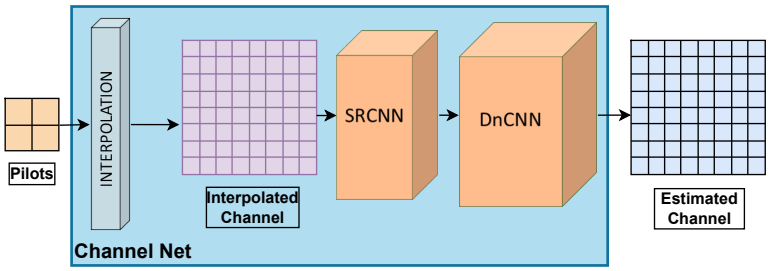

Figure 1.1: ChannelNet architecture in [1] for channel estimation.

The drawback of ChannelNet were addressed by ReEsNet [8], which introduces residual learning neural network instead of two large CNN blocks of ChannelNet. This helps in reducing complexity and even improves performance. The IReSNet [2] replaces the transposed convolution layer in ResNet with a bi-linear interpolation layer, to make the model lighter and make it more compatible with flexible pilot patterns. Still, the complexity is huge compared to conventional channel estimation approaches making it difficult to realize in real networks. The work presented in this thesis aims to address these challenges.

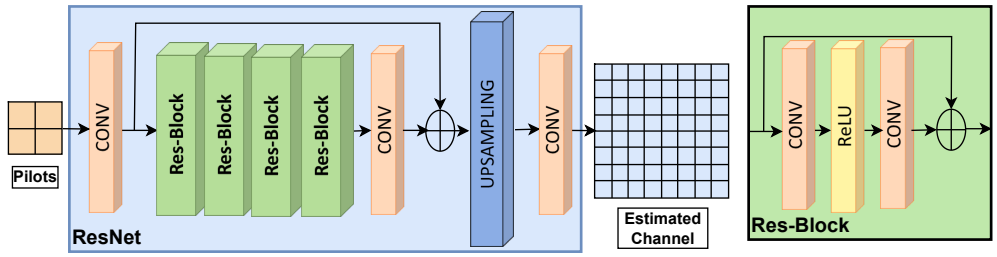

Figure 1.2: ReEsNET architecture in [2] for channel estimation.

1.4 Thesis Organisation

The thesis is organized as follows - Chapter 2 gives an overview of the end-to-end system model of the simulated environment used for the evaluation of different channel estimation schemes. Chapter 3 gives a detailed description of two conventional approaches (LS and MMSE) and two Dl-based approaches (IReSNet and LS-DNN). Chapter 4 shows the in-depth results and analysis for all the discussed channel estimation approaches and compares and compares them based on accuracy, latency and resource utilization. Chapter 5 concludes the work and summarises the results.

1.5 Notation

The notation used throughout the work is shown below in Table 1.1.

Table 1.1: Notations

Symbols	Representation
Vectors	Boldface lower case
Matrices	Boldface upper case
Transmitted Signal	$\mathbf{X}$
Received signal	$\mathbf{y}$
Channel Matrix	$\hat{\mathbf{H}}$
AWGN noise	$\hat{\hat{\mathbf{Z}}}$
LS estimated value	$\mathbf{H_{LS}}$
MMSE estimated value	$\mathbf{H_{MMSE}}$
Auto-correlation matrix	$\mathbf{R}_{H_P H_P}$
Cross-correlation matrix	$\mathbf{R}_{H H_P}$
Average power of Noise	$\sigma^2_{\mathbf{N}}$
Average power of Transmitted signal	$\sigma^2_{\mathbf{N}}$

Chapter 2: System Model Overview

In this chapter, we discuss the system architecture considered throughout this thesis work, which is an orthogonal frequency-division multiplexing (OFDM) system based on pilots symbols in the data frame. We consider a downlink scenario in a single-input and single-output (SISO) system simulated for a wide range of SNR from -5 dB to 25 dB, where the receiver can have a random mobile velocity of up to 50 kmph (Doppler=97 Hz). All the channel estimation approaches we are discussing in this thesis are pilot-based estimation techniques evaluated under 3GPP (3rd-generation-partnership-project) channel environments.

2.1 Wireless Channel

The distinct feature in mobile wireless channels is it's variation of the channel strength over time and frequency. This variation in channel strength is called fading. There are broadly two types of fadings: 1) Large-scale fading, that occurs because of path loss of signals and shadowing by large obstacles such as buildings and hills. This fading is generally frequency independent and is the matter of consideration for cell-site planning, and 2) Small-scale fading, that happens due to the interference of multiple signal paths between transmitter and receiver. This fading happens at the spatial scale of the order of the carrier wavelength, and is frequency dependent. The effects of small scale fading are mitigated at the receiver using an equalizer which uses channel state information in the form of channel frequency response, also known as channel estimates.

2.2 OFDM PHY

The main idea of OFDM is to use multi-carrier modulators with overlapping spectra and densely spaced sub-carriers. Although the spectrum of subcarriers is overlapping, the time domain components should be chosen to be mutually orthogonal. The

needed orthogonality is achieved using the Fast Fourier Transform (FFT). The key benefit of using an OFDM-system lies in its ability to handle transmitted data over fading channels. The maximum delay-spread is one of the most crucial parameters for describing fading channels, and when OFDM transmitters divide the input bit stream into several parallel bit-streams, the symbol length increases, and the relative delay spread decreases. As a result, OFDM systems can accommodate fading channels better.

2.3 OFDM-based transmitter and receiver PHY

The OFDM transmitter and receiver block diagram can be seen in figure 2.1. The data is first modulated using a QPSK modulator block, followed by Pilot insertion, which inserts known reference symbols into the data frame at predefined pilot locations. The modified frame is then passed into the IFFT block and CP addition block, after which the frame is passed to the final OFDM transmitter block, which produces the transmitted signal for the wireless channel. At Rx, the frame goes through CP removal and FFT blocks, followed by pilot extraction. This pilot-extracted data matrix is the input of the channel estimation block.

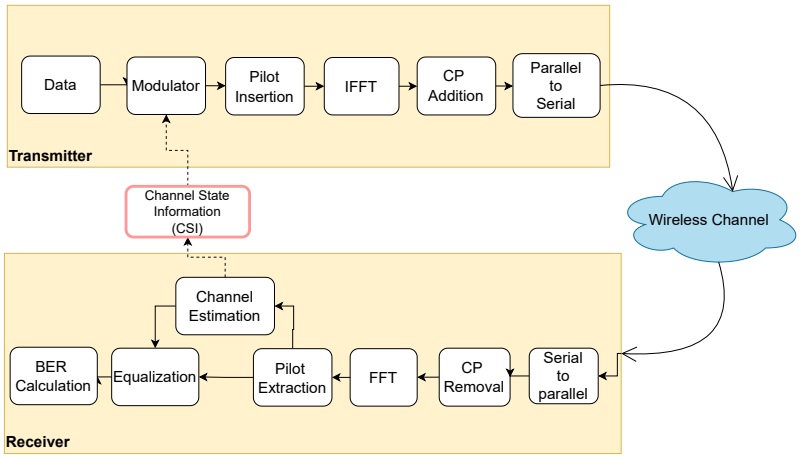

Figure 2.1: OFDM Tx Rx block

2.4 Frame Structure

The OFDM frame is generally a 2-dimensional matrix representing different OFDM symbols for every sub-carrier. Pilot symbols (known reference signals) are used in the data frame to track the modifications carried out by the channel effects on the

transmitted data. In literature, there has been a lot of work on different types of pilot arrangement used on the data-frame for respective pilot-based channel estimation. The different types of pilot arrangements include, Block type channel estimation, in which the pilots are periodically inserted into all sub-carriers of OFDM symbols, comb type channel estimation, where pilots are uniformly inserted in each transmitted OFDM sysmbol but with certain sub-carriers separated from each other within a specific period of time, or decision directed channel estimation, which includes preamble pilot symbols along with pilots in the data-frame. Owing to its high overhead, decision directed channel estimation is suitable for fast fading channel models and the block type for slow fading channel models. The frame structure used in this thesis is represented in figure 2.2

The data frame consists of 72 sub-carriers and 14 OFDM symbols. In this data frame, the 1st and 8th OFDM symbol, are defined as pilot symbols and for every pilot symbol, every third sub-carrier is again a pilot signal and all other sub-carriers of pilot symbols are null sub-carriers. Thus, we get 24 x 2 pilot values from this data-frame. These pilot values are the key inputs for pilot-based channel estimation.

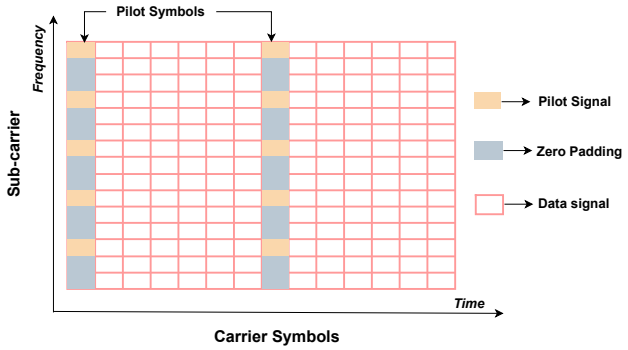

Figure 2.2: Frame Structure

2.5 Channel Estimation overview

In OFDM systems, the input-output relationship of the k_{th} symbol and the i_{th} sub-carrier is represented as:

$$\mathbf{Y}_{i,k} = \mathbf{H}_{i,k}\mathbf{X}_{i,k} + \mathbf{Z}_{i,k}; \tag{2.1}$$

where, $Y_{i,k}$ corresponds to received signal, $X_{i,k}$ and $Z_{i,k}$ corresponds to transmitted data and additive white gaussian noise, respectively. $H_{i,k}$ represents channel matrix at i_{th} symbol and k_{th} sub-carrier.

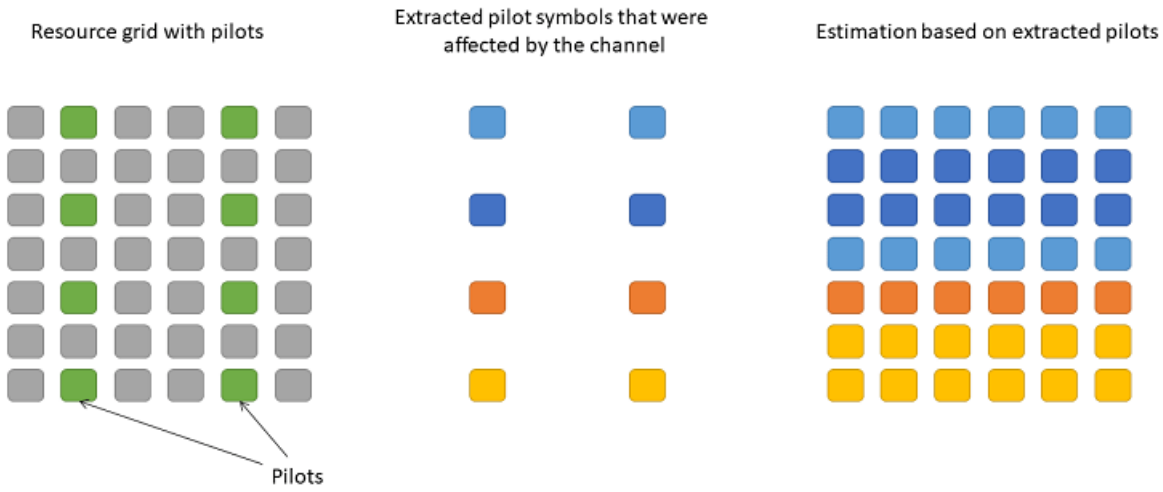

Figure 2.3: General Channel Estimation approach

The typical method of channel estimation is by adding known reference subcarriers, called pilot subcarriers, at predefined locations in the data frame and estimating the channel at these locations using the pilots. Channel at data subcarriers is estimated by interpolating the channel at pilot locations to entire frame. Such a channel estimation approach utilizing the pilot reference symbols is called pilot-based channel estimation, as shown in Figure 2.3.

2.6 Simulation Environment

The entire PHY of OFDM system i.e. OFDM transmitter, receiver and the wireless fading channel is simulated using MATLAB. All the channel estimation algorithms and approaches have been carried out on MATLAB for simulation results, whereas the hardware performance results have been carried on AMD Xilinx's Zynq series hetrogeneous SoC Zc-706, consisting of both a processing system (ARM Cortex A9) as well as an FPGA fabric on board.

The system is simulated over a wide range of SNRs (-5 dB to 25 dB), different Doppler shifts and 3GPP's multi-path fading channels: Extended Pedestrian A model (EPA), and Extended Typical Urban model (ETU). These channels differ in their delay profiles. The EPA channel model is a low delay spread environment, and while ETU is high delay spread environment as shown in 2.1

The prime focus in this work is on observing the performance (MSE as well as BER) of all the discussed estimation approaches evaluated on different channel conditions. For every SNR, 2000 frames were processed and evaluated to produce the corre-

Channel Model Delay profiles	EPA	Excess tap delay (ns)	0	30	70	90	110	190	410
		Relative power (dB)	0	–1.0	–2.0	–3.0	–8.0	–17.2	–20.8
	ETU	Excess tap delay (ns)	0	50	120	200	230	500	1600
		Relative power (dB)	–1.0	–1.0	–1.0	0	0	0	–3.0

Table 2.1: Channel Model delay profiles

sponding readings. A summary of all the system parameters and there values is represented in the table 2.2.

Parameter	**Particuar**
Modulation Type	QPSK
Guard interval type	Cyclic Prefix (CP)
Noise model	AWGN
Pilot Subcarriers	24
Pilot Symbols	2
Number of deployed subcarriers	72
CP Length	16
Bandwidth	1.08 MHz
Carrier frequency	2.1 GHz
Subcarrier Spacing	15 KHz
Number of frame per slot	1
Number of OFDM symbols per slot	14

Table 2.2: BASEBAND PARAMETERS

Chapter 3: Channel Estimation Approaches

In this chapter, we provide a detailed discussion of all channel estimation approaches explored throughout this project: the algorithm, its positives, its negatives, and lastly, the proposed hardware architecture for them.

3.1 Conventional Channel Estimation

The conventional Channel estimation approaches are generally mathematically modeled and try to exploit the statistical relationship between the pilot symbols to produce an improved channel estimate and channel state information (CSI).

But the problem with them is that either they are not able to perform satisfactorily well in the low SNR conditions or require to know the accurate channel characteristics beforehand, which is simply not a realistic alternative.

Let us now discuss each of the two conventional approaches in detail: Least Square (LS) estimation and the Minimum mean square Error (MMSE) estimation.

3.1.1 LS Estimation

The least square estimator just minimizes the squared difference between the received symbol and the golden channel response. The Least square estimate of the wireless channel at any pilot position p is given by-

$$\mathbf{H}_{LS} = min(||\mathbf{y}_p - \mathbf{H}_p\mathbf{X}_p||^2) \tag{3.1}$$

Where, y_p is the received channel response at pilot position p,H_p is channel matrix value at position p, and X_p corresponds to the transmitted signal at pilot position p.

Further, optimizing 3.1, we get the following result:

$$\mathbf{H}_{LS_p} = \mathbf{Y}_p / \mathbf{X}_p \tag{3.2}$$

Where, y_p is the received channel response at pilot position p and X_p corresponds to the transmitted signal at pilot position p.

Now, from 3.2, it is evident that the Least Square estimation does not depend on any prior information regarding the channel statistics and noise.
And since the pilot values are generally a constant value, thus LS estimator just involves a simple complex division of a constant complex value with the received complex signal value at pilot positions.

But this estimation would only produce LS estimated results at pilot positions. To get the estimation of the points other than pilot locations, we need to interpolate the LS estimated values at pilot positions to eventually get LS estimations spread over the whole channel matrix.

3.1.2 Hardware Architecture

Least Square channel estimation involves division operation of a received complex symbol and a predefined complex long training symbol. A complex division operation involves six real operations and can be implemented as shown in figure.

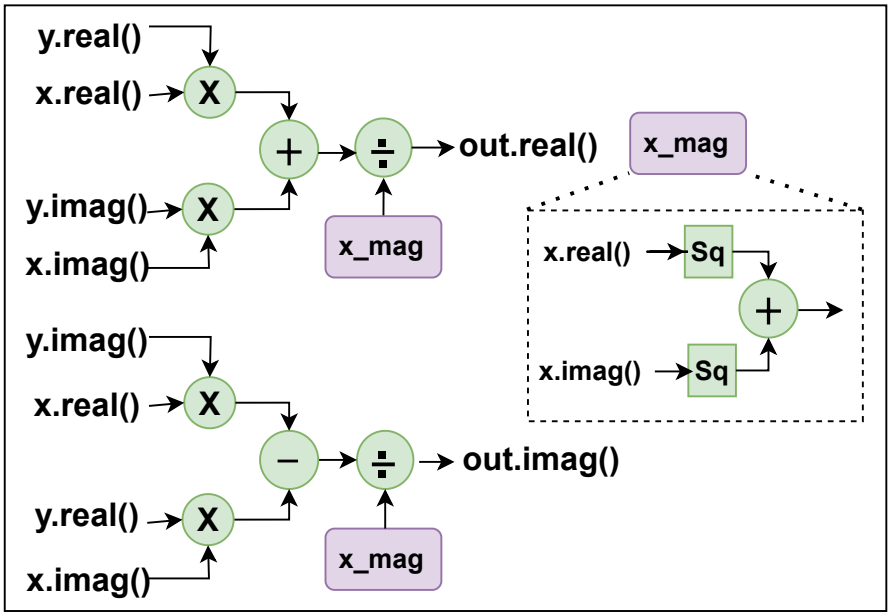

Figure 3.1: LS Hardware architecture

LS estimation is the most simplest estimation approaches of all.Owing to it's simpler design, it is expected to be light on hardware utilization as well as be quicker in terms of latency.

But in difficult situations i.e. low SNR conditions, it fails to produce sufficiently satisfactory results. And this is a massive disadvantage when it comes to critical communication applications, because this approach would simply fail systems during low SNR conditions.

3.1.3 LMMSE Channel Estimation

The Linear minimum mean square error (LMMSE) estimator is an improved version of LS estimator that uses LS estimates along with second order channel statistics, in order to produce improved channel estimation results.

By minimizing the Euclidean distance between H and H_{LS}, linear MMSE channel estimate at the pilot symbol can be concluded as :

$$\mathbf{H}_{MMSE} = \mathbf{R}_{HH_P} \times (\mathbf{R}_{H_PH_P} + I \times \frac{\sigma_N^2}{\sigma_X^2})^{-1} \times \mathbf{H_{LS}}; \tag{3.3}$$

Where H is the channel gain matrix at the pilot symbol, H_p denotes the real measured channel gain matrix for the pilot sub-carriers and $\frac{\sigma_N^2}{\sigma_X^2}$ denotes the numerical reciprocal of the signal-to-noise ratio (SNR) and the H_{LS} denotes to the LS estimate of the corresponding pilot position. The scalars σ_X^2, σ_N^2 denote the average power of the transmitted signal and the AWGN noise respectively.R_{HH_p} and $R_{H_p}H_p$ are the cross-correlation matrices of H, H_p and the auto-correlation matrix of H_P.

This LMMSE estimate is only able to estimate the values of the pilot symbols. Thus, Bi-linear interpolation is used to interpolate and get an estimate of the remaining frame values.

As evident from 3.3, LMMSE estimation involves a lot of complex matrix operations in comparison to just an LS estimator. The usual flow of MMSE computation includes first a matrix addition of diagonal matrice consisting of constant $\frac{1}{SNR}$ value and auto-correlation matrix, followed by a matrix inverse operation and then two vector multiplication operations of resultant matrix with cross-correlation matrice and LS estimated pilot symbol vector.

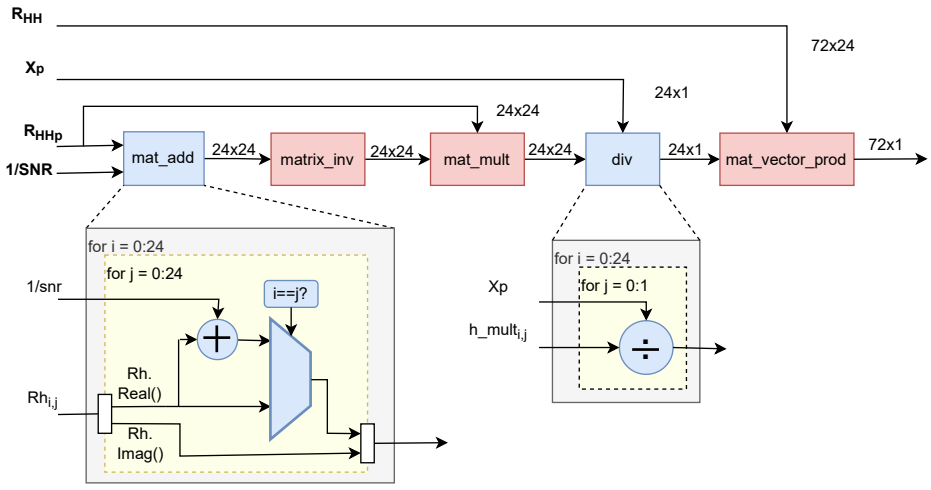

Figure 3.2: LMMSE algorithm flow

3.1.4 Hardware Architecture:

The LMMSE channel estimation requires prior knowledge of the channel correlation matrix (Rh) and SNR in addition to LTS symbols. The LMMSE architecture is shown in Fig. 8 and is based on Eq. 4. In the beginning, Rh is separated into real and imaginary matrices, and the term (1/SNR) is added with each diagonal element of the real matrix of Rh. Then, the inverse of the Rh matrix is performed. This is followed by various matrix multiplication and addition operations.

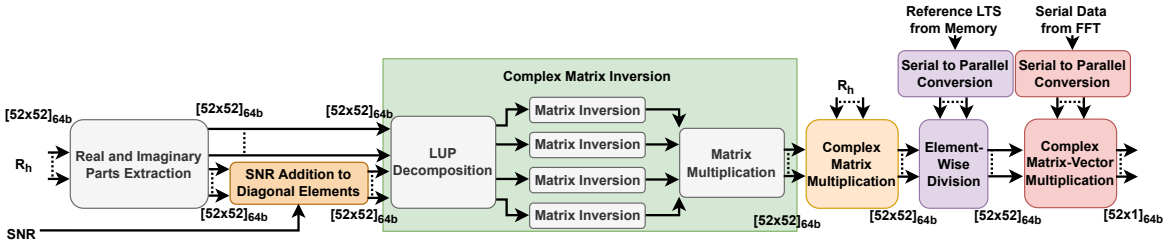

Figure 3.3: LMMSE hardware architecture

We have modified Xilinx's existing matrix multiplication and matrix inversion reference examples to support the complex number arithmetic since the baseband wireless signal is represented using complex samples. The well-known lower-upper (LU) decomposition method is selected for matrix inversion. We parallelize individual operations like element-wise division and Matrix Multiplication on the FPGA. Every element in the matrix is parallelly processed to compute division, and every row column dot product in matrix multiplication is performed in parallel to speed up the computation. In the end, multiple instances of these IPs are integrated to get the desired LMMSE functionality, as shown in Fig. 8.

3.2 Deep Learning based Channel Estimation

The mathematically modelled traditional channel estimation techniques generally fail to produce sufficient accuracy in low SNR conditions, as they are not able to figure out the impact of random noise in the channel. Meanwhile data driven deep learning based channel estimation approaches prove to a better candidate as they are able to learn the imperfections in the channel in a much efficient manner due to proper neural network training.Thus, producing much improved channel estimate results compared to the conventional approaches.

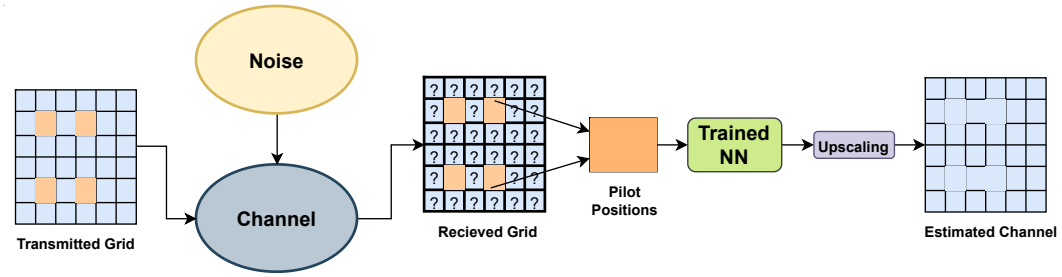

Figure 3.4: DL-based Estimation

The foremost benefit of moving to deep learning based approaches is that they can provide us better performance as well as an opportunity to accelerate the computations via leveraging the parallel processing abilities of FPGA systems. This is where hardware software co-design comes in to help optimize mapping of an algorithm on a System-on-chip (SoC).

3.2.1 CNN-based Channel Estimation

The channel matrix is a 2-dimensional matrix representing corresponding symbols on x-axis and sub-carriers on the y-axis. And since, Convolution Neural Networks are widely known to process 2-D data better, using CNNs for this application was an ideal choice. And as discussed in section 1.3, there has been a lot of work on the type and size of CNNs suited for channel estimation application.

So, we pick the most recent work [2], where the authors propose residual learning based Neural network called Interpolated ResNet (IReSNet), which outperforms all the previous versions of the CNN-based Channel Estimation approaches. Thus, in this project, we will be exploring this architecture to evaluate and compare the results against other prominent estimation techniques.

3.2.2 IReSNet

IReSNet is an improved version of ResNet [31] neural model developed for pilot based OFDM channel estimation using residual learning which proved to be highly efficient in terms of accuracy as well as total learn-able parameters compared to previous CNN based estimations.

The key modification in IReSNet compared to previous ReEsNet implementation is the omission of up-sampled convolution (also known as Transposed convolution layer) with Bi-linear interpolation layer, which not only helps in cutting down resources, but also leads to an improved performance in almost all SNR conditions.

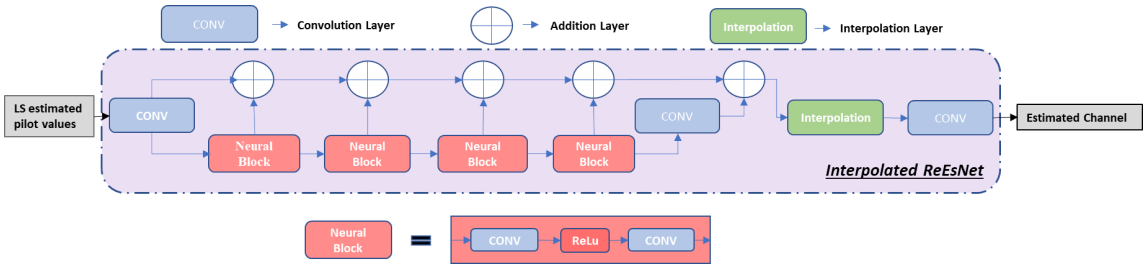

Figure 3.5: IReSNet

As depicted in 3.5, the IReSNet network consists of total 26 layers compromising of a number of convolution layers, addition layers and an interpolation layer. The network involves 1 convolution layer at front, followed by 4 successive Residual blocks consisting of CONV+ReLU+CONV arrangement, and the result is again passed through convolution layer before passing it into Bi-linear interpolation layer, which interpolates the pilot estimated values to whole channel matrix. Then lastly, a single convolution layer is used at the end after the interpolated channel estimate.

The LS estimated pilot values are given as an input to the IReSNet network and a whole estimated channel matrix is obtained at the output. The obtained results produce the best results compared to other approaches in literature and simulations, but the network consists of 10 convolution layers, 5 addition layers and 1 interpolation layer. Hence, the performance would definitely be having trade-offs with latency and resources required.

Since, the IReSNet network is fairly deep and memory intensive. So, there is always a scope of the model being over-parameterized. So, we also try to explore different flavours of IReSNet design in this project by reducing the number of residual Neural Blocks in the architecture.We explore and analyse these observations in much detail moving ahead in the Results chapter 4.

3.2.3 Hardware Architecture:

Convolution operations are the most computationally intensive part of the CNN. Being highly capable of parallelism, these are ideal candidates for FPGA based acceleration on the target heterogeneous SoC. In our proposed design, we build one generic convolution layer module that executes layer-wise CNN operations and exploits the parallelism opportunities in MAC operations of convolution operation using hardware optimizations like array partitioning and pipe-lining. And the Addition layers and interpolation layer are kept on SW (ARM cortex a9) in order to get benefit from the heterogeneous architecture of the target SoC.

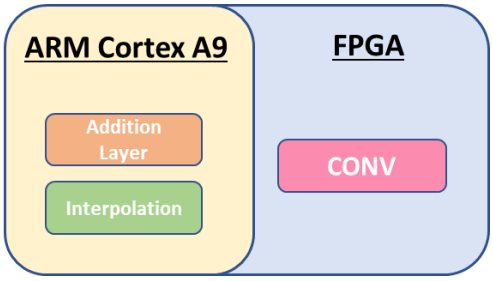

Figure 3.6: HW/SW co-design of IReSNet

3.2.4 DNN-based Channel Estimation

DNN-based channel estimation involves the use of fully connected layers in order to learn the imperfections of channel and better estimate the channel matrix. The functionality of a single neuron involves MAC operation among inputs as seen in the figure below:

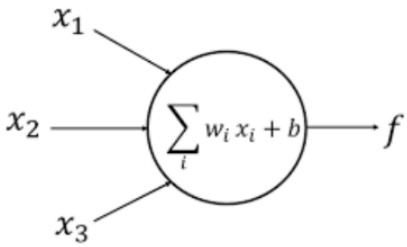

Figure 3.7: Working of a Neuron in DNN

In fully connected layers, the neuron of a layer is connected to the all the neurons of

previous and next layers having separate weights as well as biases for each connection.

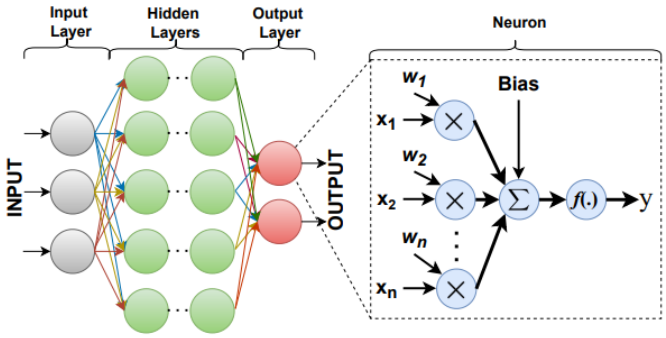

Figure 3.8: Fully connected Layer

Fully connected layers also present a great scope of parallelism due to the possibility of parallel MAC operation computation of each neuron in every layer. But, since here each of the connections has it's own weights and biases, the learn-able parameters are higher than a comparable Convolution layer, where the learn-able data is just the small kernel matrix. thus, we can say Fully connected layers are generally more memory intensive compared to the CNNs.

3.2.5 LS DNN Channel Estimation

A fully connected neural network (FCNN) will be augmented to least square channel estimation to improve the performance. As shown in the block diagram below, first pilots will be extracted and the channel at pilot positions will be estimated using LS. Then the LS estimated channel will be passed through an FCNN to get the improved performance. To get the channel at data locations, the estimated channel will be interpolated using Bi-Linear interpolation layer.

Layer	Weights	Bias
Input Layer	96x48	48
Hidden Layer-1	48x2016	2016
Output Layer	-	-

Table 3.1: Layer description of LS-DNN network

The learn-able parameter in LS-DNN are 101.3k, which is approximately 10x times more than what IReSNet required (9.3k).

3.2.6 Hardware Architecture

The hardware architecture of the DNN involves array partitioning the weights of each layer to be able to execute concurrent MAC operations of each layer with inputs vector and biases. A descriptive view of the hardware architecture is shown in the figure below:

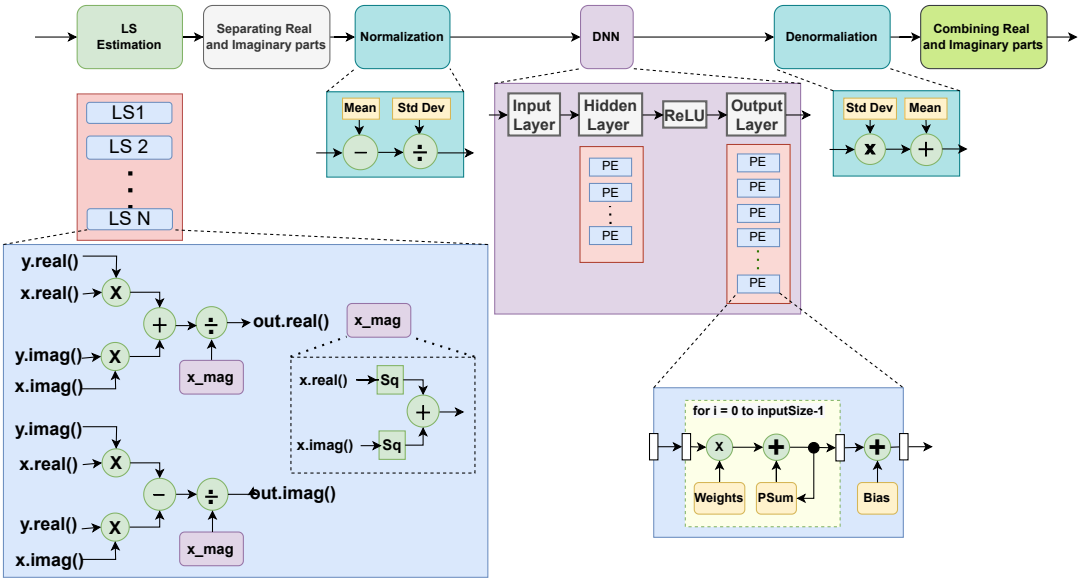

Figure 3.9: LS DNN architecture

Chapter 4: Performance Analysis and Complexity Comparison

In this chapter, we compare the functional accuracy of various channel estimation approaches for a wide range of SNRs, wireless channels, and word length. We use the mean square error (MSE) of channel estimation output and bit-error-rate (BER) of the end-to-end transceiver as performance metrics. Then, we compare the resource utilization and execution time of various architectures obtained via hardware-software co-design and word-length optimization. All the architectures are implemented on AMD Xilinx's ZC-706, a state-of-the-art Xilinx's Zynq series heterogeneous SoC platform consisting of a hard processor (ARM Cortex A9) and FPGA fabric (programming logic). Here, we consider four channel estimation approaches: 1) LS, 2) LMMSE, 3) IReSNet, and 4) LSDNN. In the case of IReSNet, we consider three different architectures for a block size of 2, 3, and 4. They are referred to as IReSNet_2, IReSNet_3, and IReSNet_4, respectively. Similarly, for LSDNN, we consider two architectures: 1) Compute-efficient architecture, LSDNN_CE, and 2) Low execution time, i.e. low latency architecture, LSDNN_LL. We do not consider ChannelNet [1] and ReEsNet [8] since it is already shown that the IReSNet [2] has lower complexity than them.

4.1 Functional Accuracy Verification

In this section, we compare the functional accuracy of various architectures using floating-point arithmetic. Later, we optimize the word length of each architecture such that it does not lead to significant degradation in functional accuracy. We consider two well-known multipath fading channels Extended Pedestrian A model (EPA), and Extended Typical Urban model (ETU). We consider the SNR range from -5 dB to 25 dB.

4.1.1 Double Precision Floating Point (DPFP) Word Length

In Fig. 4.1, we compare the MSE of all architectures for a wide range of SNRs in the presence of EPA and channels. As expected, MSE decreases with the increase in SNR. In both channels, LMMSE offers lower MSE than LS. It can be observed that DL-based approaches outperform LS and LMMSE and they do not need prior knowledge of the channel as in LMMSE. The performance of IReSNet degrades with the decrease in block size. The MSE performance of the LSDNN is slightly worse than that of the IReSNet. Note that each of the DL algorithms is trained independently for the respective channel and prior knowledge of channel type is needed. Please refer to Chapter 5 for more details.

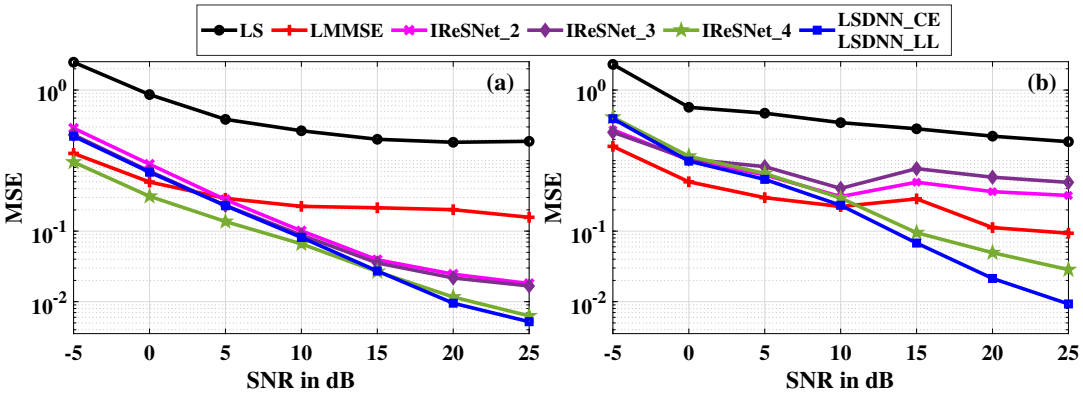

Figure 4.1: Comparison of MSE for different channel estimation approaches over a wide range of SNRs for (a) EPA Channel, and b) ETU Channel.

Next, we compare the BER of end-to-end transceiver systems for a wide range of SNRs in the presence of EPA and channels. It is well known that BER is a preferred metric to analyze the performance of wireless PHY compared to MSE. It can be observed that the BER of the LMMSE is better than that of the LS. Furthermore, DL approaches offer lower BER than LS and LMMSE, especially at high SNR. Though LSDNN incurs higher MSE at low SNR than IReSNet, the BER of both approaches is nearly identical. Thus, we can conclude that the overall channel estimation performance of DL approaches is better than LS and LMMSE. Also, IReSNet and LSDNN offer identical BER performance for both types of channels.

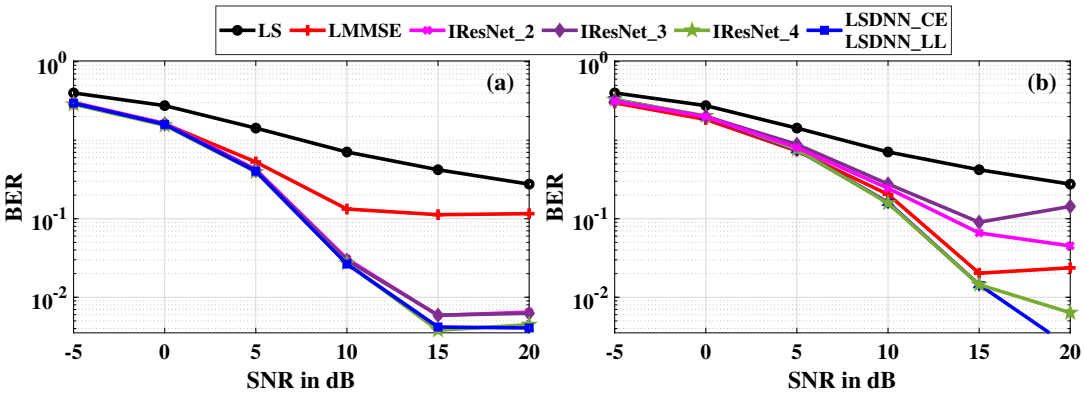

Figure 4.2: Comparison of BER for different channel estimation approaches over a wide range of SNR for (a) EPA Channel, and b) ETU Channel.

4.1.2 Fixed Point Word Length

Since architecture with SPFP WL suffers from high resource consumption, latency, and power consumption, fixed-point WL is preferred. Fixed point WL is represented as $< W, I >$ where W, I is the number of total and integer bits, respectively. Thus, $(W - I)$ represents the total number of fractional bits. The fixed-point WL selection involves identifying a number of bits to represent integer and fractional parts. Since each algorithm has different arithmetic operations and hence, the dynamic range of intermediate outputs, WL needs to be selected independently for each architecture.

Fixed-Point WL Selection for LS:
For LS, we compare the average MSE for different WL with respect to that of SPFP WL. As shown in Table 4.1, we fixed the number of fractional bits to a high value of 12 and vary the total number of integer bits. It can be observed that the MSE degrades significantly for I lower than 4.

Table 4.1: Selection of I for LS

Word Length	$(W - I)$	**Average MSE**
SPFP	-	0.29778
<15,3>	12	0.52125
<16,4>	12	0.29778
<17,5>	12	0.29778

Next, for fixed $I = 4$, we find out the minimum possible W. In Fig. 4.3, we compare the effect of (W) on the MSE performance of LS for fixed $I = 4$ and it can be observed that the MSE does not improve for any $W \geq 8$. Thus, we have selected the WL of <8,4> for the LS architecture.

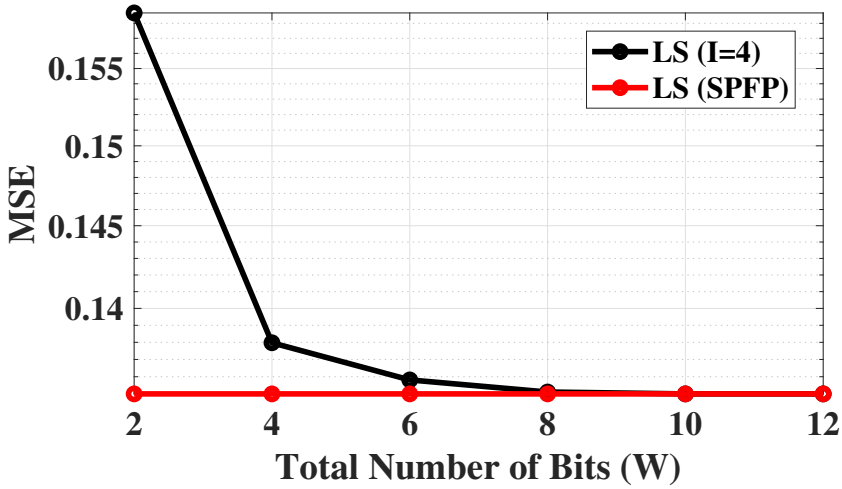

Figure 4.3: Effect of (W) on the MSE performance of LS for fixed $I = 4$.

In Fig. 4.4, we compare the MSE of LS with SPFP WL, and two fixed point WL of <8,4> and <8,3>. It can be observed that even single-bit reduction results in a significant increase in MSE thereby validating the selection of <8,4> WL for LS.

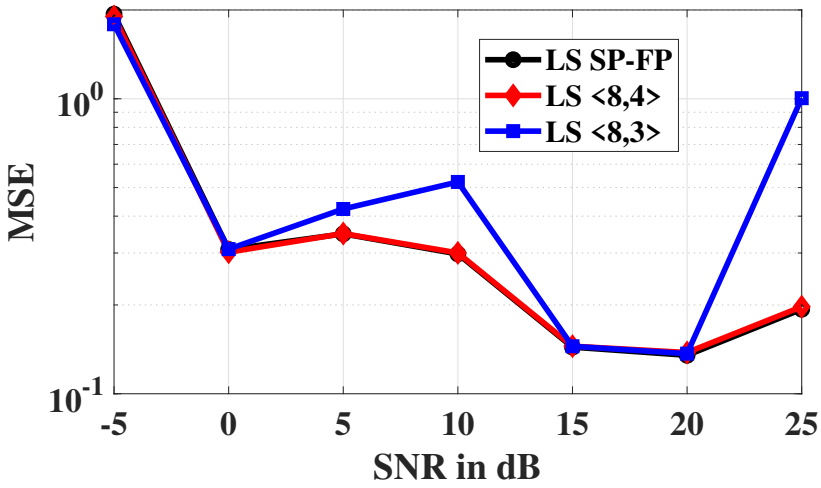

Figure 4.4: Comparison of the MSE for different WL architectures of LS over a wide range of SNR.

In Fig. 4.5, we compare the resource utilization of LS architecture for different WL. It can be observed that architecture with WL of <8,4> offers significant savings in resources over LS SPFP architecture without compromising functional accuracy. Numerically, architecture with WL of <8,4> offers 66%, and 33% savings in flip-flops (FF) and look-up-table (LUT) in FPGA. In addition, it eliminates the need for embedded digital signal processing (DSP) units in FPGA.

Fixed-Point WL Selection for LMMSE:

Since LMMSE involves matrix inversion, the dynamic range of intermediate outputs

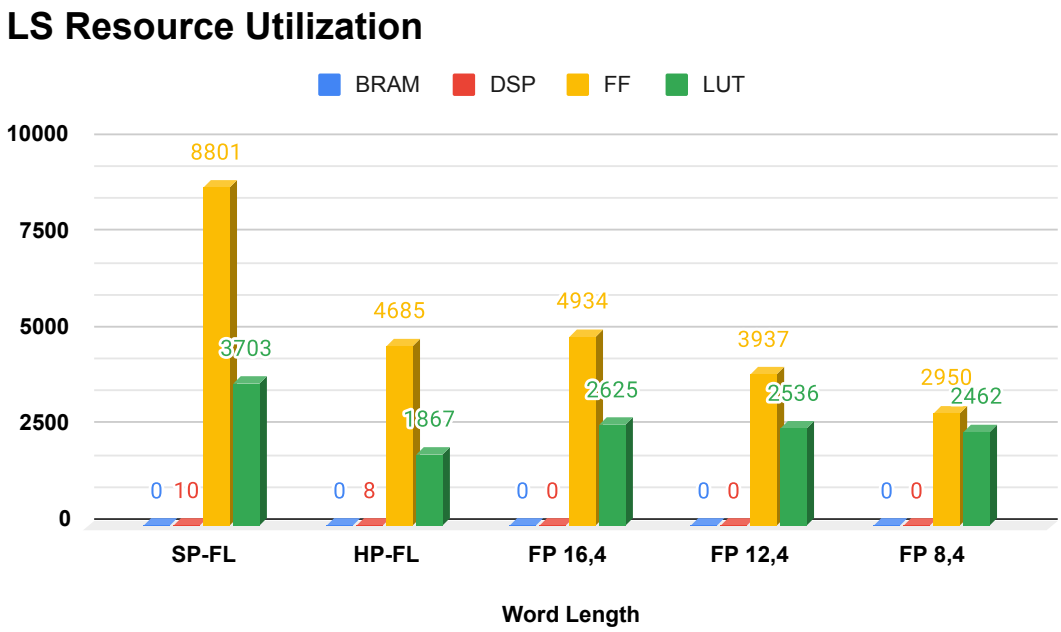

Figure 4.5: Comparison of the FPGA resource utilization for different WL architectures of LS.

is high. Hence, we could not get desirable accuracy with the fixed-point implementation of the LMMSE.

Fixed-Point WL Selection for IReSNet:
For IReSNet, we compare the average MSE for different WL with respect to that of SPFP WL. As shown in Table 4.2, we fixed the number of fractional bits to a high value of 12 and vary the total number of integer bits. It can be observed that the MSE degrades significantly for I lower than 4.

Table 4.2: Selection of I for IReSNet

Word Length <W,I>	$W-I$	**MSE**
SPFP	-	0.099368
15,3	12	0.228793
16,4	12	0.099639
17,5	12	0.099891

Next, for fixed $I = 4$, we find out the minimum possible W. In Fig. 4.6, we compare the effect of (W) on the MSE performance of LS for fixed $I = 4$ and it can be observed that the MSE does not improve for any $W \geq 16$. Thus, we have selected the WL of <16,4> for the IReSNet architecture.

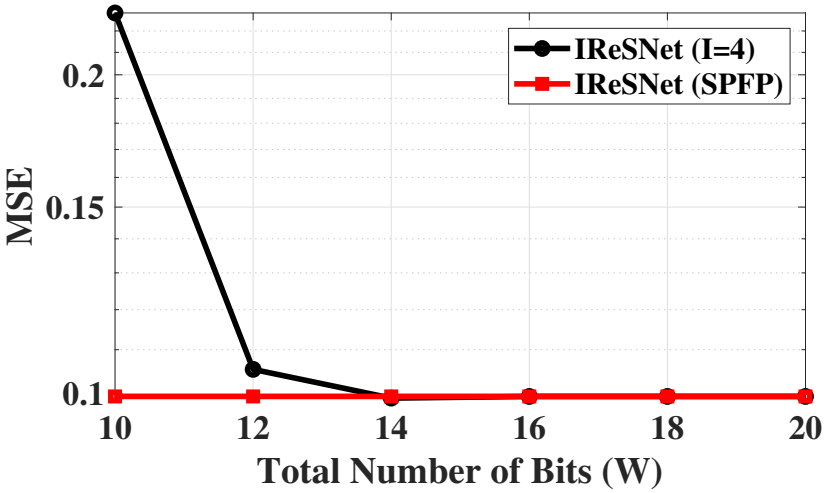

Figure 4.6: Effect of (W) on the MSE performance of IReSNet for fixed $W - I = 4$.

In Fig. 4.7, we compare the MSE of IReSNet with SPFP WL, half-precision floating point (HFFP) WL, and three fixed point WL of <16,4> <12,5> and <8,5>. It can be observed that WL of <16,4> offers similar performance as that of SPFP WL.

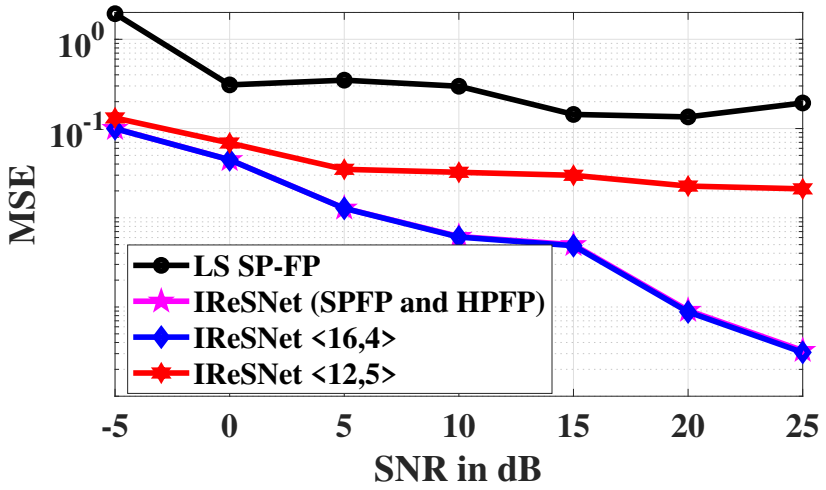

Figure 4.7: Comparison of the MSE for different WL architectures of IReSNet over a wide range of SNR.

In Fig. 4.8, we compare the resource utilization of IReSNet architecture for different WL. It can be observed that architecture with WL of <16,4> offers significant savings in resources over IReSNet SPFP architecture without compromising functional accuracy. Numerically, architecture with WL of <16,4> offers 50%, 85%, 60%, and 49% savings in block RAM (BRAM), DSP, FF, and LUT in FPGA.

Fixed-Point WL Selection for LSDNN:

For LSDNN, we compare the average MSE for different WL with respect to that of

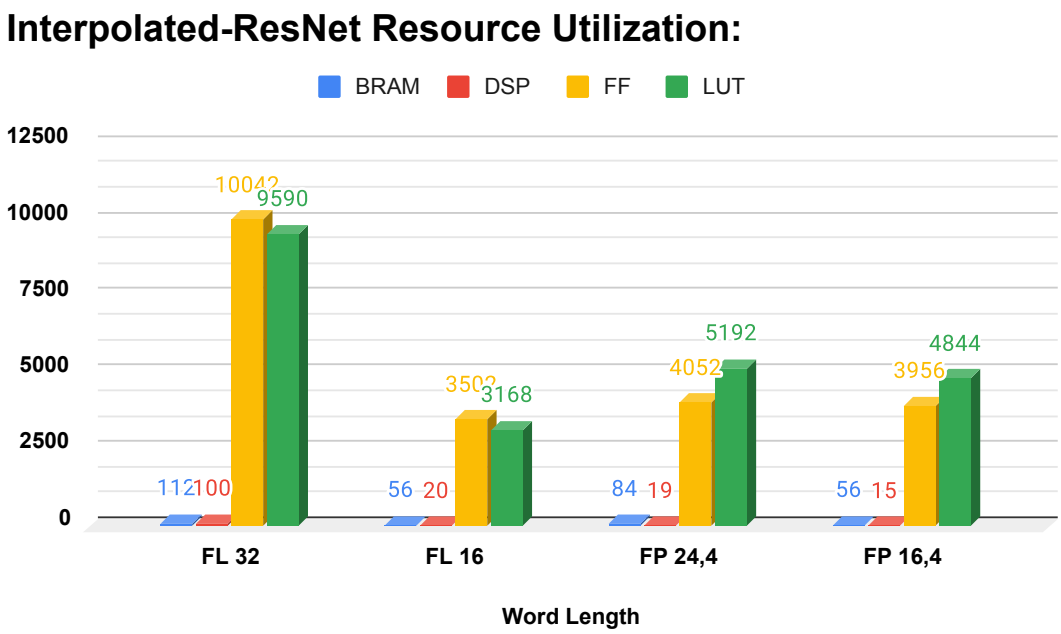

Figure 4.8: Comparison of the FPGA resource utilization for different WL architectures of IReSNet.

SPFP WL. As shown in Table 4.3, we fixed the number of fractional bits to a high value of 18 and vary the total number of integer bits. It can be observed that the MSE degrades significantly for I lower than 8.

Table 4.3: Selection of I for LSDNN

Word Length <W,I>	$W - I$	**MSE**
sw	-	0.125617
25,7	18	0.249926
26,8	18	0.125617
27,9	18	0.125617

Next, for fixed $I = 8$, we find out the minimum possible W. In Fig. 4.9, we compare the effect of (W) on the MSE performance of LS for fixed $I = 8$ and it can be observed that the MSE does not improve for any $W \geq 26$. Thus, we have selected the WL of <26,8> for the LSDNN architecture.

In Fig. 4.10, we compare the MSE of LSDNN with SPFP WL, and two fixed point WL of <26,8> and <26,7>. It can be observed that even single-bit reduction results in a significant increase in MSE, thereby validating the selection of <26,8> WL for LSDNN.

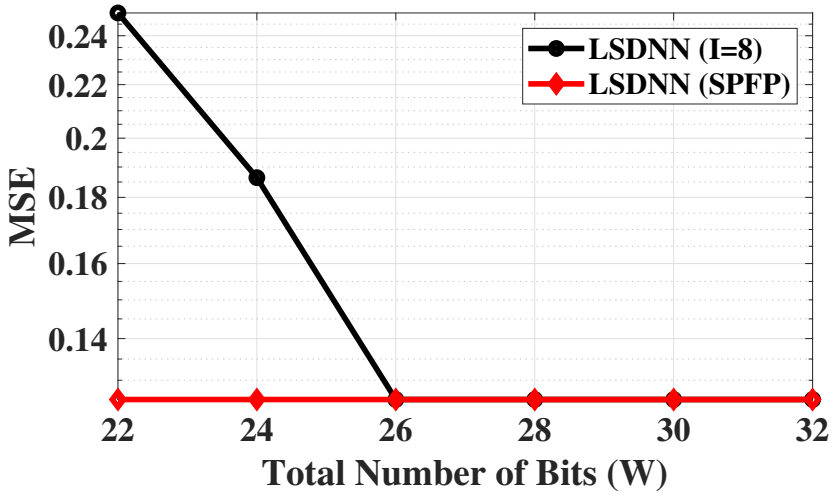

Figure 4.9: Effect of (W) on the MSE performance of LSDNN for fixed $W - I = 4$.

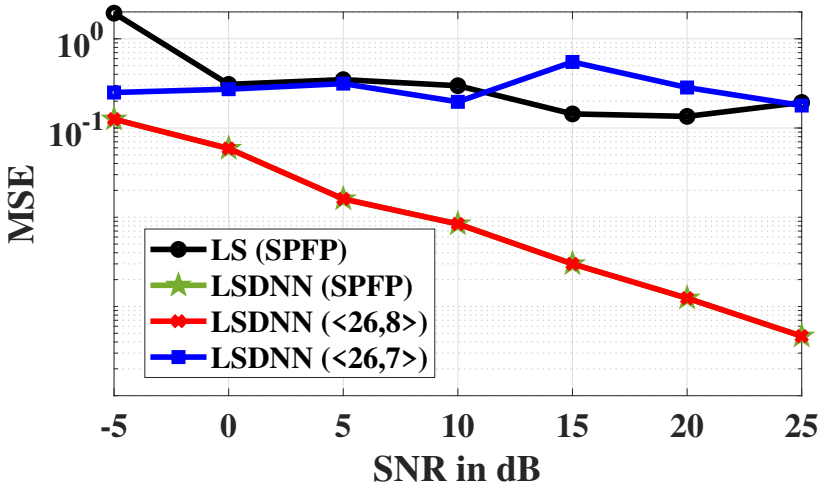

Figure 4.10: Comparison of the MSE for different WL architectures of LSDNN over a wide range of SNR.

In Fig. 4.11, we compare the resource utilization of LSDNN architecture for different WL. It can be observed that architecture with WL of <26,8> offers significant savings in resources over IReSNet SPFP architecture without compromising functional accuracy. Numerically, architecture with WL of <26,8> offers 19%, 20%, 30%, and 20% savings in block RAM (BRAM), DSP, FF, and LUT in FPGA.

4.2 Hardware Software Co-design

In this section, we discuss various configurations of each architecture obtained by hardware-software co-design (HSCD) on Zynq SoC. Specifically, each configuration differs from others depending on which part of the architecture is realized in the pro-

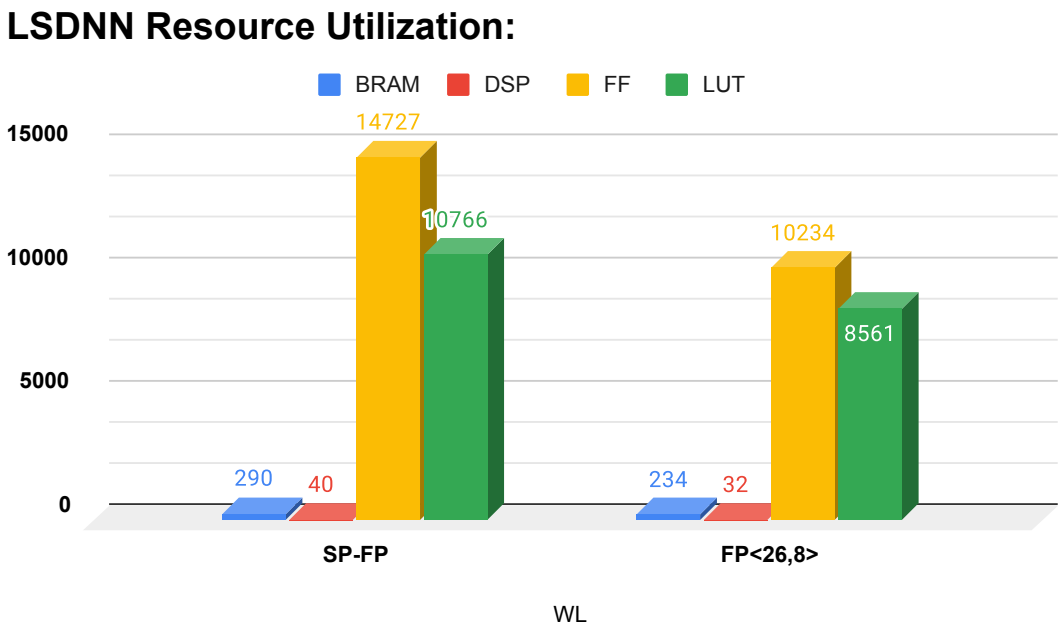

Figure 4.11: Comparison of the FPGA resource utilization for different WL architectures of LSDNN.

cessing system (PS) i.e. ARM processor, and programmable logic (PL), i.e., FPGA. In the end, the architecture which offers lower latency and fewer FPGA resources is selected.

4.2.1 HSCD: LS

The LS is relatively simple to implement and consists mainly of LS computation and interpolation (INTP) to the channel estimate over the entire OFDM frame. We considered four configurations obtained by moving LS and INTP operations between PS and PL. From table 4.4, it can be observed that the realization of LS-based channel estimation is that PS offers the lowest latency; hence, FPGA-based acceleration is not needed.

Table 4.4: HW-SW Co-design for LS

S. No.	PS	PL	Execution Time	BRAM	DSP	FF	LUT
1	LS + INTP	NA	0.070028	-	-	-	-
2	LS	INTP	0.098543	0	8	2045	3016
3	INTP	LS	0.071297	0	10	8801	3703
4	NA	LS + INTP	0.070528	0	12	10524	5605

4.2.2 HSCD: LMMSE

In the case of LMMSE, we have realized it completely in PL due to computationally complex floating-point matrix operations.

4.2.3 HSCD: IReSNet

The IReSNet architecture consists of three computational blocks: B1) Convolution layer (CONV), B2) Addition layer (ADD) and B3) Interpolation layer (INTP). As shown in Table 4.5, the realization of CONV in PL offers a significant improvement in latency.

Table 4.5: HW-SW Co-design for IReSNet

S. No.	PS	PL	Latency	BRAM	DSP	FF	LUT
1	B1+B2+B3	-	40.25	-	-	-	-
2	B1+B2	B3	40.45	0	8	2045	3016
3	B1+B3	B2	36.86	0	10	1024	1153
4	B2+B3	B1	18.26	112	100	10042	9590
5	B3	B1+B2	18.99	112	116	12009	12417
6	B2	B1+B3	19.24	112	108	12352	12648
7	-	B1+B2+B3	19.26	112	126	14620	15244

4.2.4 HSCD: LSDNN

The LSDNN architecture consists of LS and two DNN layers, L1 and L2. Based on previous results, LS is realized in PS. From Table 4.6, the architecture with both layers in FPGA offers the lowest latency.

Table 4.6: HW-SW Co-design for LSDNN

S. No.	PS	PL	Execution Time	BRAM	DSP	FF	LUT
1	L1+L2	-	0.76345	-	-	-	-
2	L2	L1	0.723881	32	160	19638	15051
3	L1	L2	0.114685	264	32	12735	9773
4	-	L1+L2	0.08123	290	40	14735	10766

4.3 Comparison of Fixed-Point Architectures

This section compares the functional accuracy, resource utilization, and latency of various architectures obtained via HSCD and word-length analysis. As shown in

Fig. 4.12, DL-based channel estimation architectures offers better performance than LS and LMMSE at the lowest possible WL.

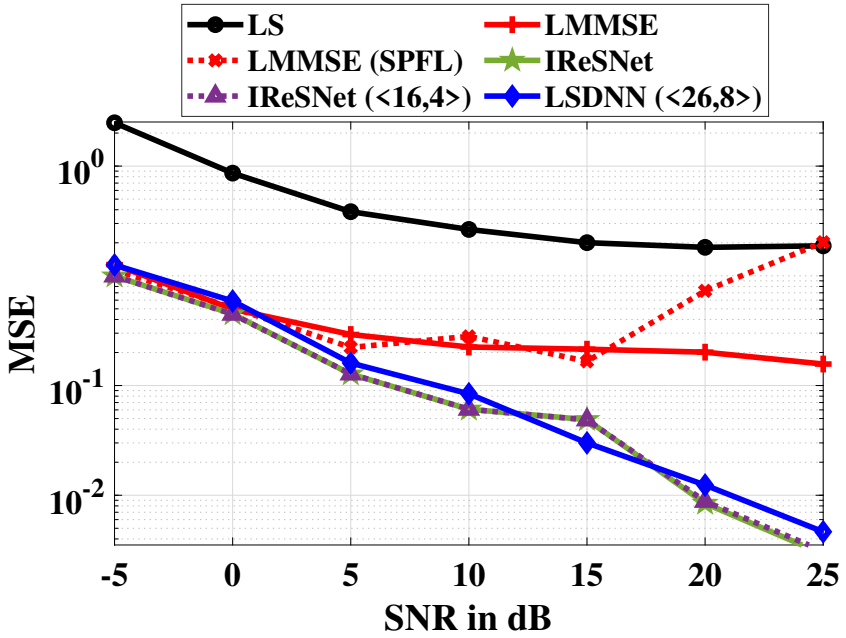

Figure 4.12: MSE comparison of various fixed-point architectures.

In Table 4.7, we compare various fixed-point architectures' resource utilization and latency. For IReSNet, we consider three variations depending on the block size. All of them have identical functional accuracy, but they differ in latency.

Similarly, we consider two architectures for LSDNN. In LSDNN_CE, resource utilization is reduced with a slight increase in latency. In LSDNN_LL, latency is reduced with a slight increase in resource utilization. It can be observed that the latency of LSDNN_CE and LSDNN_LL is 88% and 98% lower than that of IReSNet. At the same time, LSDNN_CE offers is 85%, 60%, 35%, and 38% savings in BRAM, DSP, LUTs and FFs, respectively, over IReSNet. Similarly, LSDNN_CE offers 75% lower latency along with 93%, 94%, 90% and 94% savings in BRAM, DSP, LUTs and FFs, respectively.

Table 4.7: Comparison of Various Fixed-point Architectures

S.No.	Architecture	Word Length	Latency	BRAMs	DSPs	LUTs	FFs
1	LS	SW	0.070028	-	-	-	-
		SP-FL	0.070787	0	10	8801	3703
		FP 8_4	0.070558	0	0	2950	2462
2	LMMSE	SPFL	2.81965	114	101	25416	52564
3	IReSNet_2	SW	39.82998	112	100	10042	9590
		FP 16_4	6.50355	56	15	3956	4844
4	IReSNet_3	SP-FL	39.275744	112	100	10042	9590
		FP 16_4	6.373281	56	15	3956	4844
5	IReSNet_4	SW	40.25315	-	-	-	-
		SP-FL	18.36422	112	100	10042	9590
		FP 16_4	6.63591	56	15	3956	4844
6	LSDNN_LL	SW	0.76345	-	-	-	-
		SP-FL	0.08784	290	40	14727	10766
		FP 26_8	0.08632	234	32	10234	8561
7	LSDNN_CE	SP-FL	0.7962	8	6	2939	3543
		FP 26_8	0.714885	8	6	2554	3021

Chapter 5: Conclusion and Future Scope

In this thesis, we studied the feasibility of deep learning (DL) based channel estimation for the wireless physical (PHY) layer on system-on-chip (SoC). We have realized the existing statistical and DL-based channel estimation approaches on SoC via hardware-software co-design and word-length analysis. We show that the existing DL approaches offer improved mean square error and lower bit-error rate than the least square (LS) and linear minimum mean square error (LMMSE) approaches. We observed that DL-based approaches are relatively easy to implement and optimize on FPGA than LMMSE due to simple arithmetic operations. However, they have very high complexity and latency. We designed LS augmented deep neural network (LS-DNN) algorithm, which has significantly lower complexity and latency than existing DL approaches for a given MSE and BER performance. Via in-depth experimental results for a wide range of SNR and wireless channels and complexity analysis, we demonstrated the superiority of the proposed LSDNN approach over existing works.

One of the main drawbacks of the DL-based approach is the dependence on the training dataset. We observed that the performance of LMMSE and DL-based channel estimation degrades significantly in unknown channel environments. Even in a known channel environment, LMMSE and DL-based approaches are sensitive to channel parameters such as Doppler frequency. Still, they perform significantly better than LS approach. The degradation in performance due to unknown channel conditions can be addressed in two ways". The first approach is to design generalized DL architecture, which works well in multiple channel conditions. However, such generalized architectures are computationally complex. The second approach addresses this problem at the hardware level via reconfigurable architecture. We can configure the hardware with appropriate DL architecture based on the current channel conditions. This will enable the use of small DL models resulting in lower complexity and latency. However, we need an additional intelligence layer to identify the current channel conditions and corresponding DL models. The design of such intelligent and reconfigurable channel estimation is part of future work.

One more limitation of existing SoC-based approaches is that the training of DL models is done offline. In the future, we plan to develop a real-time framework for training on SoC so that optimal models can be trained on-the-fly for a given channel condition.

References

[1] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 2019.

[2] D. Luan and J. Thompson, "Low complexity channel estimation with neural network solutions," 2022.

[3] Y. Ji, J. Yu, Y. Yao, K. Yu, H. Chen, and S. Zheng, "Securing wireless communications from the perspective of physical layer: A survey," *Internet of Things*, vol. 19, p. 100524, 2022.

[4] J. M. Hamamreh, H. M. Furqan, and H. Arslan, "Classifications and applications of physical layer security techniques for confidentiality: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1773–1828, 2019.

[5] R. Vatti and A. Gaikwad, "Throughput improvement of wireless networks using collision control approach," vol. 6, pp. 15–23, 01 2016.

[6] S. Chattopadhyay, "Improving performance of high throughput wireless access networks - an experience in learning," in *23rd International Conference on Distributed Computing and Networking*, ICDCN 2022, (New York, NY, USA), p. 229–231, Association for Computing Machinery, 2022.

[7] M. Z. Asghar, S. A. Memon, and J. Hämäläinen, "Evolution of wireless communication to 6g: Potential applications and research directions," *Sustainability*, vol. 14, no. 10, 2022.

[8] L. Li, H. Chen, H.-H. Chang, and L. Liu, "Deep residual learning meets ofdm channel estimation," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 615–618, 2020.

[9] S. Kay, *Fundamentals Of Statistical Processing, Volume 2: Detection Theory*. Prentice-Hall signal processing series, Pearson Education, 2009.

[10] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[11] M. K. Karray and M. Jovanovic, "A queueing theoretic approach to the dimensioning of wireless cellular networks serving variable-bit-rate calls," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2713–2723, 2013.

[12] S. J. Darak and M. K. Hanawal, "Multi-player multi-armed bandits for stable allocation in heterogeneous ad-hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2350–2363, 2019.

[13] N. Singh, S. V. S. Santosh, and S. J. Darak, "Toward intelligent reconfigurable wireless physical layer (phy)," *IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 226–240, 2021.

[14] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 93–99, 2019.

[15] J. Gao, X. Yi, C. Zhong, X. Chen, and Z. Zhang, "Deep learning for spectrum sensing," *IEEE Wireless Communications Letters*, vol. 8, no. 6, pp. 1727–1730, 2019.

[16] X. Sun, C. Wu, X. Gao, and G. Y. Li, "Fingerprint-based localization for massive mimo-ofdm system with deep convolutional neural networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10846–10857, 2019.

[17] R. Zhou, F. Liu, and C. Gravelle, "Deep learning for modulation recognition: A survey with a demonstration," *IEEE Access*, vol. 8, pp. 67366–67376, 2020.

[18] M. Wang, A. Wang, Y. Zhang, and J. Chai, "Research on the performance of an end-to-end intelligent receiver with reduced transmitter data," *Applied Sciences*, vol. 12, no. 22, 2022.

[19] O. J. Famoriji, O. Y. Ogundepo, and X. Qi, "An intelligent deep learning-based direction-of-arrival estimation scheme using spherical antenna array with unknown mutual coupling," *IEEE Access*, vol. 8, pp. 179259–179271, 2020.

[20] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and doa estimation based massive mimo system," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8549–8560, 2018.

[21] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based csi feedback approach for time-varying massive mimo channels," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 416–419, 2019.

[22] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.

[23] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[24] K. Lin, C. Li, J. J. P. C. Rodrigues, P. Pace, and G. Fortino, "Data-driven joint resource allocation in large-scale heterogeneous wireless networks," *IEEE Network*, vol. 34, no. 3, pp. 163–169, 2020.

[25] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1277–1290, 2019.

[26] X. Zhang, H. Zhao, J. Xiong, L. Zhou, and J. Wei, "Scalable power control/beamforming in heterogeneous wireless networks with graph neural networks," 04 2021.

[27] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[28] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 2019.

[29] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[30] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[31] L. Li, H. Chen, H.-H. Chang, and L. Liu, "Deep residual learning meets ofdm channel estimation," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 615–618, 2020.