Power, Performance and Area Modelling Techniques for System on Chips

Abhishek Sengupta
Master of Science in Communications Engineering
Technical University of Munich
Munich,Germany
ge79car.sengupta@tum.de

***Abstract*—Power,Performance and Area (PPA) are the most crucial factors to be considered while developing an System-On-Chip to improve performance, durability and reliability . This paper primarily focuses on an accurate PPA estimation using Machine Learning techniques like Neural Networks, Decision Trees. The predictions are based on the design parameters of the hardware system. The models utilize Transfer Learning for the predictions in order to save resources from training models multiple times to complete relatable tasks.We achieved an accuracy of upto 98% in PPA predictions.**

***Index Terms*—Power, Performance, Area, PPA, Machine Learning, Transfer Learning, System-On-Chip.**

I. Introduction

Power,Performance and Area of a System-On-Chip should be taken into consideration through all the design phases. Contemporary processing systems are embedded with diversified components like CPU's, GPU's and have countless design alternatives like frequency, memory registers, number of cores.These factors need to be considered at the earliest design phase so that these does not affect the final design. Hence, we use Machine Learning models at various stages of the design process to predict the PPA of that stage and the future design stages based on the hardware design at that stage. The ML model can be trained on the PPA measurements that we obtain by running applications on test hardware and capturing the change in the PPA measurements on the variation of hardware parameters like clock frequency, memory capacity like RAM, number of processors. An approximation of the PPA at a preliminary design phase can assist us in optimizing the PPA of the chip. We focus on Transfer Learning in order to use a miniumum amount of data to train the model and hence save on both time and resources.

II. State of the art of PPA Analysis using Machine Learning

PPA Modelling has been a very critical thing in the design of System-On-Chips. Different Machine Learning techniques have been used over time for PPA predictions. PPA predictions have been made at all design phases to develop an overall optimized circuit.

Previously Convolutional Neural Networks(CNN's) have been used in the prediction of the power consumed by a chip. Also, various Machine Learning frameworks have been developed to estimate the performance of an FPGA based on its design. Neural Networks have also been deployed to calculate PPA for hardware optimization.

In this paper, we try to demonstrate the generation of PPA models using Machine Learning algorithms and to deliver a comparitive study of the different models based on their degree of accuracy.

III. Machine Learning Models for PPA Modelling

Here we demonstrate the various Machine Learning algorithms that can be used for PPA Modelling. We focus on using Transfer Learning, that is to re-use trained models for PPA predictions of different System-On-Chips. Later, we go on to do a comparitive study of the different models based on their degree of accuracy.

A. Neural Network

A Neural Network(NN) is one of the most widely used Machine Learning Models. The NN consists of an input layer, hidden layers and an output layer. Each layer contains many neurons which are connected with the neurons of the other layers.Each connection has a weight and on being trained it learns from the data. Each layer has an activation function which is added in order to help the NN learn complex patterns in the data.The activation function helps in adding non-linearity to the data and also helps in restricting the output from a neuron restricted to a certain limit.

A Neural Network can share the learning among various similar hardware problems because learning is in the form of weights. This is where the phenomenon of Transfer Learning comes into play. We have decided to design a base model with the same set of weights and a set of layers which are trained specifically for each unique hardware design problem. A Neural Network with a shared base model and a single

trainable hidden layer for different hardware designs is shown in Fig 1.

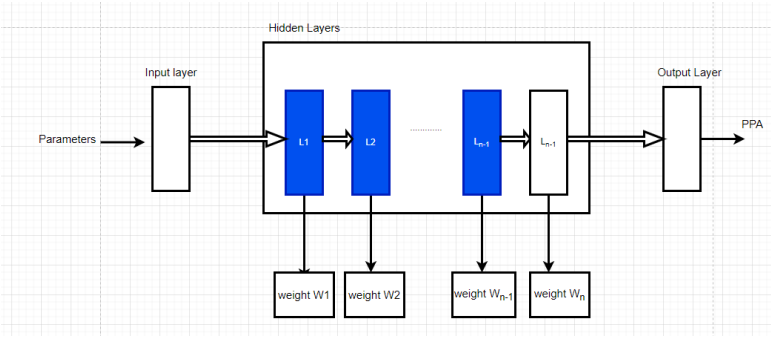

Fig. 1. Neural Network with the shared layers depicted in green

In this way, we can limit the quantity of data required for a new PPA prediction.This phenomenon of Transfer Learning where we use a base model is based on the fact that the internal structure of the hardware may be similar to some extent.

B. Decision Trees

Decision Tree is an algorithm that contains conditional control statements. It has a tree like structure that branches out from a root leaf node and forms many leaf nodes where decisions are made. In PPA, a Decision Tree can make predictions based on conditional statements like if the number of cores in a processor is greater than a certain amount, or the instructions executed in a circuit in unit time is greater than a threshold. Decision Trees are a very reliable model of prediction based on internal calculations of Entropy and Information Gain.

Entropy is an Information Theory metric by which the amount of uncertainty in a group of observations is measured Given a dataset with N classes, the Entropy can be measured as :

$$E = -\sum_{i=1}^{n} p_i log_2 p_i$$

where p_i is the probability of randomly selecting an example in class i.

Information Gain is a parameter that helps in determining the quality of splitting. The Information Gain can be calculated as :

$$InformationGain = Entropy_{parent} - Entropy_{children}$$

The parent node can be defined as the node where splitting occurs and the children nodes are formed after splitting at the parent node. The less the entropy of the child, the more the Information Gain. Hence, to make accurate decisions we choose the branch with the highest Information Gain.

C. Gradient Boost Regressor

Gradient Boosting works on the principle that the best possible next model when combined with the ensemble of the previous models, can lead to a better prediction accuracy. It leads to the minimization of the overall prediction errors.For implementing the Gradient Boosting Algorithm, we need a loss function which is to be optimized, a weak learning model to make predictions and an additive model to add to the weak learning model in order to minimize the loss function. Hence, the Gradient Boost Regressor, iteratively combines the weak learners like Decision Trees into a strong learner.

IV. Model Parameters and Model Generation

A. Parameters used for PPA Modelling

The PPA Modelling of a System-On-Chip depends on the architectural design of the chip. Hence to calculate the power, we take into account the Register Transfer Level(RTL) Implementation, RTL synthesis parameters such as clock period, enable signals, circuit delay. Based on these parameters at a particular design stage, we can predict the PPA at that particular stage and in the future design stages.

B. PPA Model Generation Framework

A Framework to generate the PPA model is shown in Fig 2. In the first stage, we take up the design data from the RTL. Then a sample space is generated using a Latin Hyper-Cube(LHC). Once the samples are generated, we run the RTL Synthesis tool to obtain the PPA for all the combinations of parameters from the sample space. The parameters of the extracted PPA can be used in the training set to train the model and the final model is obtained.

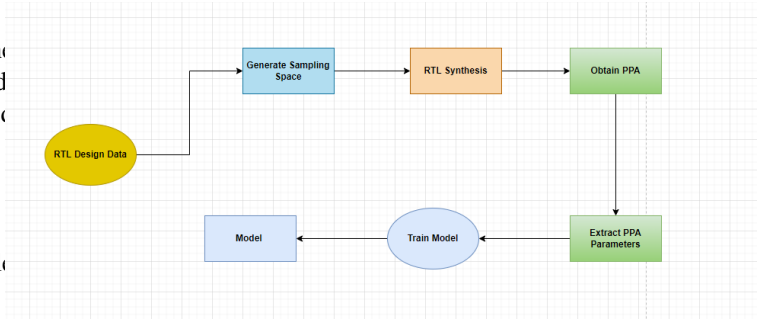

Fig. 2. PPA Model Generation Framework

C. PPA Model Testing Framework

A Framework to test the generated PPA model is being shown in Fig 3. This model helps in determining whether the extracted PPA parameters of the test hardware are accurate enough for the hardware design to move to the next stage. Firstly, we take up the design data and perform the RTL Synthesis. Then we obtain the PPA and extract its parameters. If the accuracy target is met, we proceed to the next stage of the design process, and if it is not met the RTL design data needs to be changed and the process needs to start afresh.

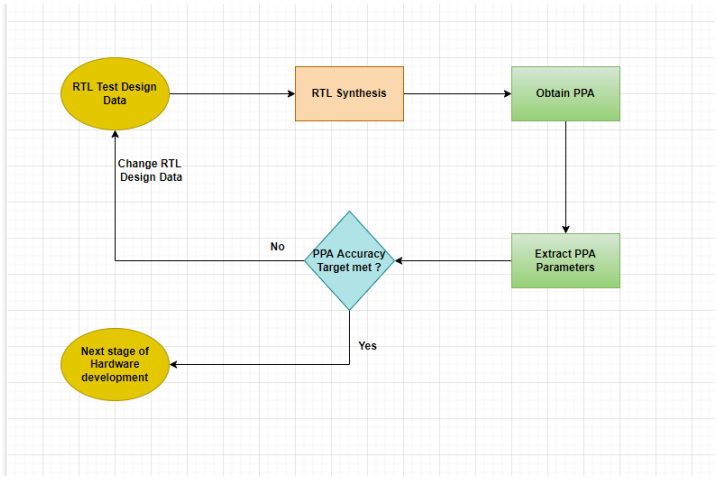

Fig. 3. PPA Model Testing Framework

V. Dataset to create and evaluate models

The RTL Design data contains parameters like number of combinatorial and sequential blocks, registers, finite state machine's and clock signal measurements. The RTL synthesis is done for numerous combinations of these parameters and the PPA measurements are extracted for each of the synthesized RTL's. These PPA measurements serve as the training data in relation to that particular RTL design from where it was extracted. We use this data to train our model and later make PPA predictions based on a given RTL synthesis.

VI. Experiments and Results