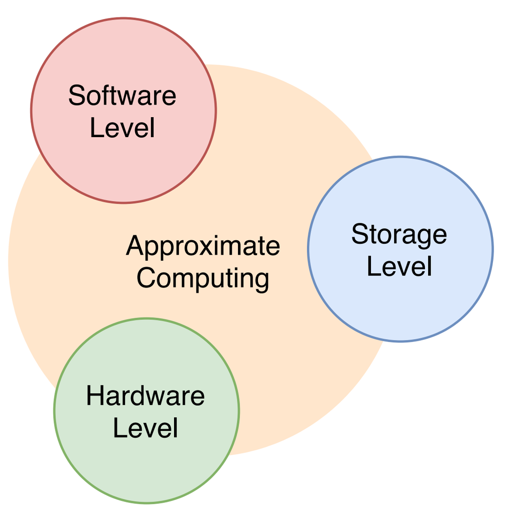
Towards Approximate-Computing AI Applications through Code-Mutation and Genetic Search

XXXIV Cycle - II year presentation

Salvatore Barone
Tutor: Antonino Mazzeo

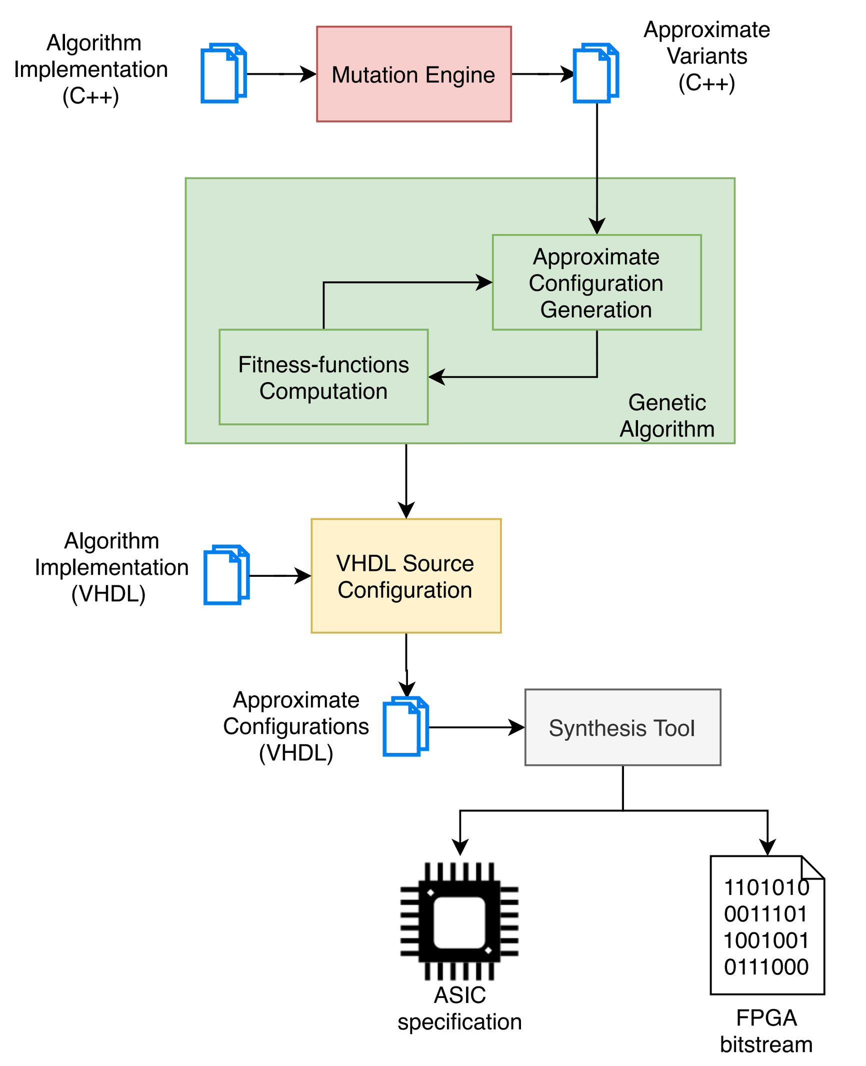
The Era of Approximate Computing

- Rather than the best possible result, Approximate Computing allows to achieve better computational performances by **carefully** relaxing non-critical functional system specifications.
- Interested domains:
  - signal processing (audio, video, image), artificial intelligence, machine learning, data mining, and so forth.



- The literature proved that Approximate Computing is effective due to the **inherent application resiliency**
  - a property for an algorithm to return acceptable outcomes despite some of its inner computations being inaccurate;
  - IA applications, such as classifiers and Neural Networks, are excellent examples of resilient algorithms!
  - The required area overhead makes the design of a hardware accelerator unfeasible.
- There is no generic and application-independent approach;
- The solution space grows very quickly;
- Error metrics definition is critical;
- **Accuracy and gains are conflicting objective**.
- Multi-objective Optimization is NP-hard

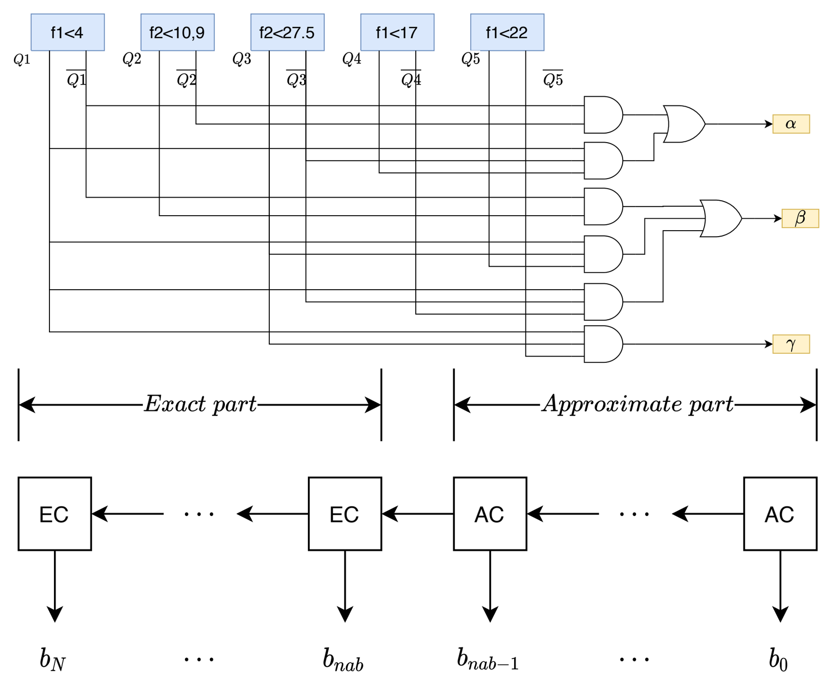
A MOP-based flexible approach to the Design of Approximate Hardware

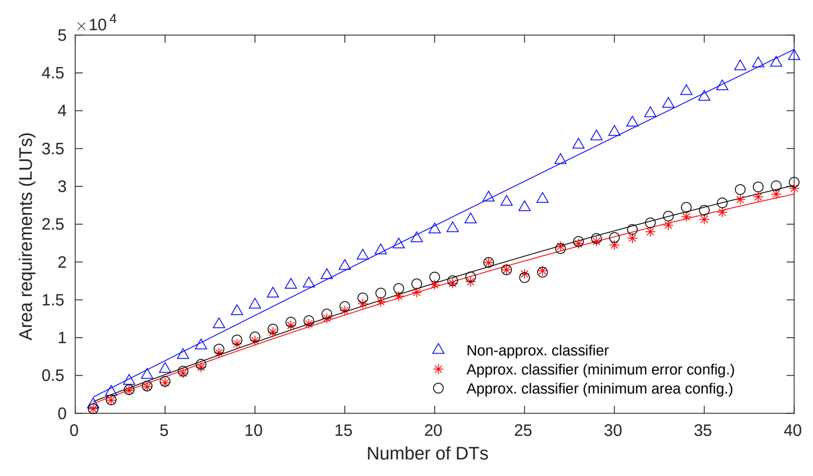


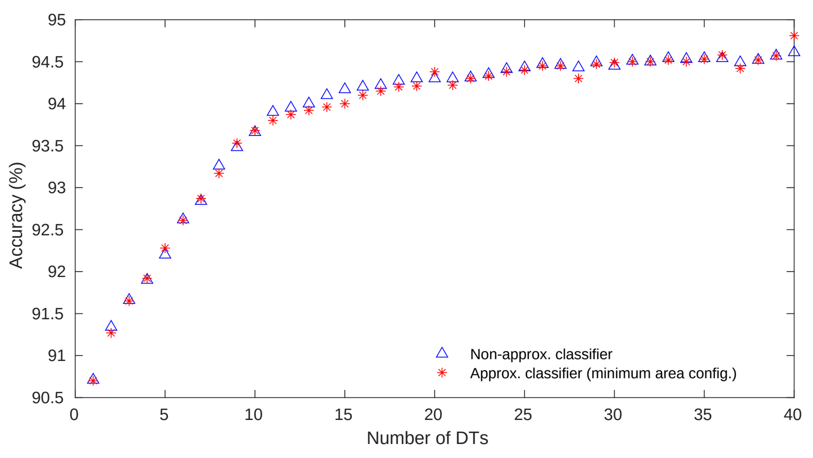
- To speed-up simulations, we consider C/C++ implementation of the algorithm to be approximate.
  - Approximate variants generation is performed using the Clang-Chimera tool, which is an Clang/LLVM-based C/C++ source-to-source mutation engine part of the IIDEAA framework.
  - The MOP resolution is performed by using the ParadisEO framework, a template-based evolutionary computation library.
- Approximate-configurations provided at the end of the DSE are employed to configure VHDL sources and perform synthesis.
- The methodology does not depend on a particular domain:
  - It does not take into account the training process;
  - Approximation is introduced on trained models.
  - Fitness-functions for MOP still have to be defined case-by-case

Approximate DT-based MCSs

- Exploit hardware-provided parallelism.
- 50 different model, with classes and trees ranging from [x,y] and [1,40], respectively
- Precision scaling to reduce FPGA resource requirements;
- Error metric: classification accuracy;
- Gain estimation: amount of neglected bits.

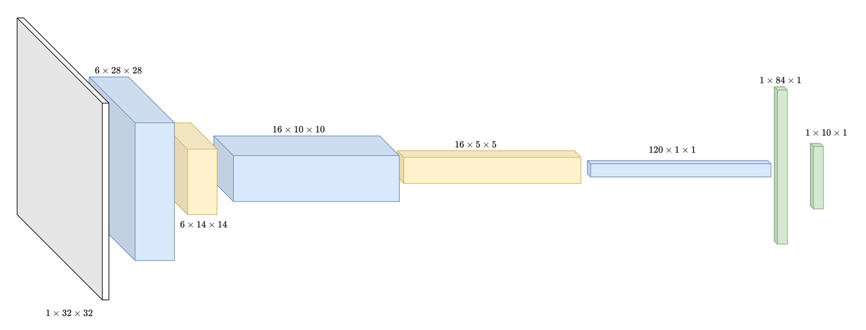


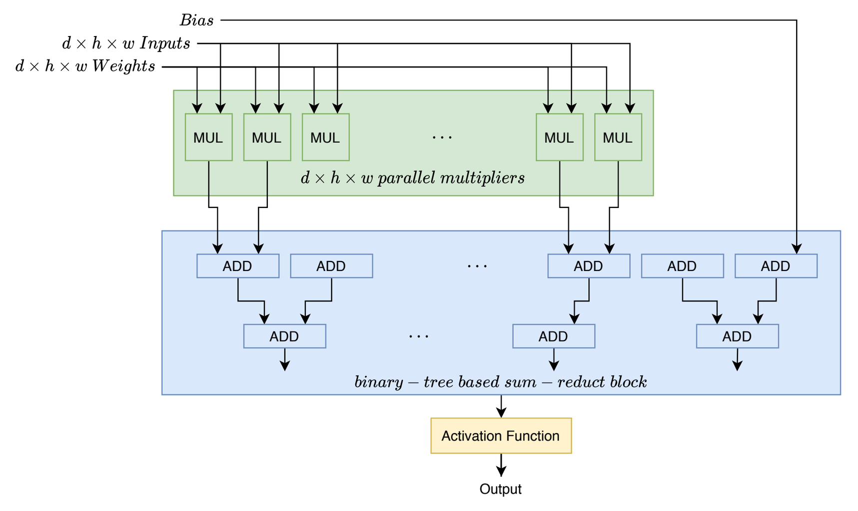


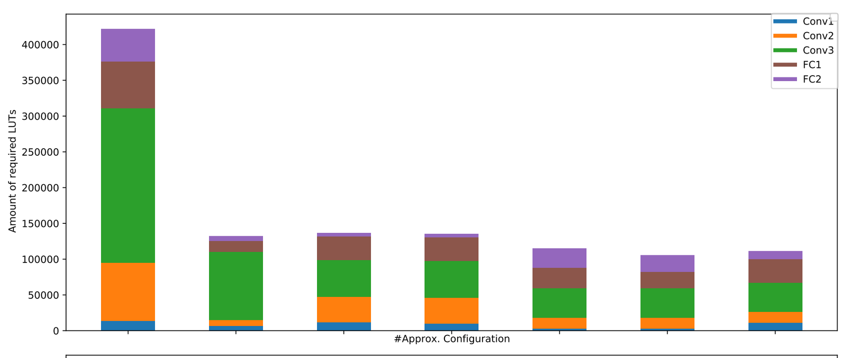


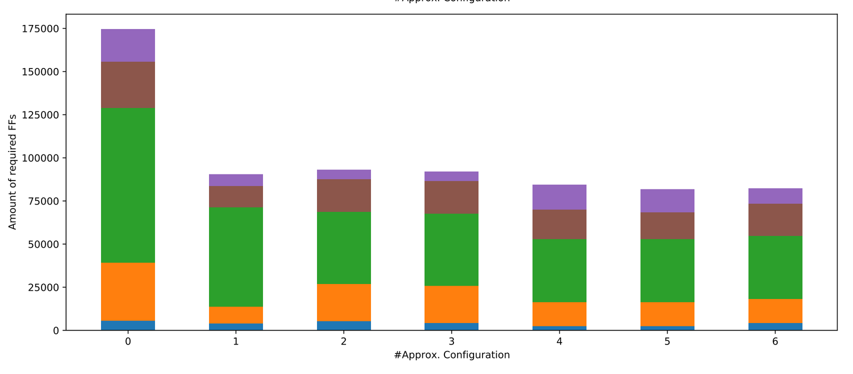
Approximate Neural Networks

- Preliminary study on LeNet5
- 5 different model of the same network (double, float, clustered float, int16, int8);
- Imprecise arithmetic (lsb truncation), to reduce FPGA resource requirements;
- Error metric: classification accuracy
- Gain estimation: weighted sum of neglected bits.









Future Developments

- Experimental results show a significant reduction in area requirements, for both the minimum error and minimum area configuration.
  - Since the classification is very resistant to error, those configurations are very similar both in terms of area requirements and classification error.
- The LeNet5 model is quite simple when compared to modern CNNs;
  - Result show hardware implementation of the whole network is still infeasible on a single FPGA;
  - Single neuron hardware accelerator is feasible on mid-range FPGA.
- Different approximate computing techniques:
  - Loop-perforation;
  - Inexact aritmetics (instead of mere truncation);
- Different CNN/RNN models;
  - SqueezeNet
  - MobineNet
  - ResNet.

Contacts

DIETI – UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE DELL'INFORMAZIONE

salvatore.barone@unina.it
salvator.barone@gmail.com
salvator.barone

Join the conversation here!

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Ph.D itee INFORMATION TECHNOLOGY ELECTRICAL ENGINEERING