EEE 598 "Neuromorphic Computing Hardware Design" Spring 2022

Final Project Assignment

Your final project end goal is to design an energy-efficient custom hardware using 7nm PDK for MNIST handwritten digit recognition task. The final metric will involve both your hardware's energy consumption (per one MNIST image) and test accuracy for the MNIST dataset.

Towards the goal of designing energy-efficient hardware, you can start from a publicly available algorithm for MNIST MLP/CNN, but you might also want to consider tailoring a software algorithm to make it more hardware friendly, such as lowering the precision or pruning/compressing the neural network. Low-precision or compressed networks could involve some amount of accuracy degradation, so still you need to find the optimal design point of lower the energy consumption while minimizing the accuracy loss.

The information and contents provided below could serve as references for your project design and implementation, but you don't necessarily need to follow any exact content or methodology. These are for your references and guidelines.

Software for artificial neural network (ANN)

A couple representative deep learning tutorial in Python / Matlab are provided below.

https://pytorch.org/tutorials/

https://classroom.udacity.com/courses/ud187

https://www.mathworks.com/products/deep-learning.html

As you also went through in HW2, there are many other publicly available codes, webpages and videos, if you search on the internet for deep learning, deep neural networks, convolutional neural networks, etc.

Software for spiking neural network (SNN)

For SNN algorithm, you can either (1) convert a trained ANN to a SNN, or (2) train a SNN from scratch using back-propagation (learning rules from ANN), or (3) train a SNN from scratch using STDP or bio-inspired learning rules.

For (1), a representative paper would be https://ieeexplore.ieee.org/abstract/document/8351295, and the codes for this paper can be found at https://github.com/NeuromorphicProcessorProject/snn_toolbox.

For (2), a representative paper would be https://ieeexplore.ieee.org/document/8325230, and the codes for this paper can be found at https://github.com/ShihuiYin/BASNN.

For (3), this scheme is not strongly recommended, mainly because reported accuracies of SNNs trained with STDP-based learning have not been as high as those of (2) or ANNs.

Hardware ANN papers

http://ieeexplore.ieee.org/document/7801877/

http://ieeexplore.ieee.org/document/7738524/

http://ieeexplore.ieee.org/document/7870354/

http://dl.acm.org/citation.cfm?id=3001163

http://ieeexplore.ieee.org/document/7560203/

http://ieeexplore.ieee.org/document/7870351/

https://arxiv.org/pdf/1711.00215.pdf

https://ieeexplore.ieee.org/document/8310262

https://ieeexplore.ieee.org/document/9246543

Hardware SNN papers (digital CMOS)

http://ieeexplore.ieee.org/document/7231323/

http://ieeexplore.ieee.org/document/7015626/

https://ieeexplore.ieee.org/document/8325230

Project topic decision guidelines

Your project objective is to design a neural network hardware for MNIST dataset. You can use either ANN or SNN, but your job is to work on the classification hardware (not training). For classification, it would mean that you can obtain an already trained network that is publicly available, or use a network that is trained in CPU/GPU, or apply some hardware-friendly training techniques yourselves, and then you will implement the classification neural network in custom hardware.

There are several ways to think about towards lower the energy consumption of your neural network hardware while minimally degrading the test/classification accuracy. In a high-level, they will include smaller neural networks, efficient dataflow architecture, low precision hardware, and compression. While considering these, you don’t want to degrade the test accuracy noticeably, since both energy and test accuracy (error) are used for the final metric that your hardware design will be evaluated on.

Sample project topics

For further technical discussion with any of the following topics, make an appointment to setup a meeting time by sending an email to jaesun.seo@asu.edu.

New circuit/architecture techniques that can make classification or learning more efficient?

- Memory, logic, architecture, dataflow, on-chip communication, power, etc.

Sparsity/compression and accuracy vs. hardware analysis

- Both ANNs and SNNs can benefit from sparsity (weights are sparse such that many of them are zero, spike events are sparse, etc.). For a given application, how much sparsity or compression can the neural network tolerate without losing substantial accuracy? How much power can be benefited from such sparsity?

- Sample papers on neural network compression include:
  https://arxiv.org/pdf/1608.03665.pdf
  https://arxiv.org/pdf/1510.00149.pdf
  https://dl.acm.org/citation.cfm?id=3080215

Binary / low-precision neural network design, tradeoff analysis

- There have various 'binary' neural networks that have been recently proposed, such as only making the weights binary (BinaryConnect: https://arxiv.org/abs/1511.00363) or making both the neuron activations and weights binary (BNN: https://arxiv.org/abs/1602.02830, XNOR-Net: https://arxiv.org/abs/1603.05279). Inside all these three papers, there are links to the publicly available github codes.

- Besides binarization, various low-precision (e.g. 2-bit, 4-bit) neural network algorithm/hardware designs have been proposed, including https://arxiv.org/abs/1606.06160, https://www.jmlr.org/papers/volume18/16-456/16-456.pdf, https://ieeexplore.ieee.org/document/8335699, etc.

- The quantized low precision model often requires the float-to-integer conversion before the hardware deployment:

  https://arxiv.org/pdf/1712.05877.pdf,

  The integer-only inference module is also available in Pytorch (https://pytorch.org/docs/stable/quantization.html) and Tensorflow (https://www.tensorflow.org/model_optimization/guide/quantization/training)

Information related to final project

The given 7nm asap7 standard cell library doesn't have flip-flops with asynchronous reset. Please write your Verilog RTL to only use flip-flops with synchronous reset.

Unfortunately there are no memory compilers available for this 7nm CMOS process, so you will use latches of flip-flops for memory elements or storage of weights/synapses and neurons.

From behavioral Verilog RTL, you will generate synthesized netlist, and then proceed to Innovus automatic place-and-route to generate the layout. Then

(1) Synthesis and Automatic Place & Route (APR)

- Refer to class lecture and the instruction/tutorial document that is provided for synthesis using Cadence RTL compiler and automatic place and route (APR) using Cadence Innovus.
- You should employ a power/ground ring for M2 (horizontal) / M3 (vertical) that surrounds the core region in Innovus, and connect M1 VDD/VSS in the standard cells to the ring.
    - Examples of the power rings/stripes can be found at /afs/asu.edu/class/e/e/e/eee525b/asap7_library/scripts/example.apr.tcl
- For clock tree synthesis in Innovus, follow the steps in the tutorial, and you should try to optimize further.

(2) Power Measurement

- Follow the instructions in “Innovus_power_analysis.pdf” document uploaded at Canvas and refer to the user manual to measure the average power consumption that corresponds to your testbench operation using the saved *.vcd file.

The energy consumption you report should be based on the (1) latency measurement from post-layout netlist simulation (at the clock frequency that doesn’t give you any timing violation) and (2) power measurement by Innovus power analysis using the post-layout simulation with the *.vcd file.

Essentially, energy = latency × power.

Final project metric

All the submissions will be evaluated and sorted using the following formula:

$$Normalization\ Value = \begin{cases} 1.002, & students\ in\ team = 2 \\ 1.0, & students\ in\ team = 3 \\ 0.998, & students\ in\ team = 4 \end{cases}$$

$$Modified\ error = 1 - \left(\frac{Accuracy * Normalization\ value}{100}\right)^2$$

$$Quality\ Metric = Energy^{0.7} * Modified\ error^{1.5} * 10^3$$

Accuracy should be reported as percentage (e.g. 98.55%), Energy required for one classification of one image should be reported in µJ. You would want to minimize the quality metric above. Points for optimization will be given proportional to the rank of the submission. That is, the project with rank *i* (=0, 1, 2, …) will get the following points for optimization.

$$Points(i) = \frac{Number\ of\ submissions - i}{Number\ of\ submissions} \times 15$$

Final project timeline

From now on till the end of the semesters, please visit office hours or setup appointments with instructor to finalize project ideas, discuss the progress of the project, etc. (send email to jaesun.seo@asu.edu for separate appointments)

For all deadlines below, late submission penalty will be applied as follows.

- Submission late by <=30 minutes: no penalty
- Submission late by >30 minutes and <= 2 hours: 10% (out of 100%) deducted
- Submission late by >2 hours and <= 6 hours: 20% (out of 100%) deducted
- Submission late by >6 hours and <= 24 hours: 40% (out of 100%) points deducted
- Submission late by >24 hours: no points

3/23 (Wed) 5:00pm: Submission of first draft of project idea due

- Document should include: motivation, baseline, new ideas to explore (if any), what you want to implement, ideal results you want to obtain
- Send one email per team to jaesun.seo@asu.edu (copy all team members in the email)

3/30 (Wed) 5:00pm: Submission of final draft of project idea due

- Document should include: motivation, baseline, proposed scheme, what exactly you will be implementing, expected results, application

- Send one email per team to jaesun.seo@asu.edu (copy all team members in the email)

4/20 (Wed) 5:00pm: Submission of intermediate project report due

- Use IEEE template: http://www.ieee.org/conferences_events/conferences/publishing/templates.html
- The intermediate paper should include: (1) outline of the paper with sections and subsections, (2) bullet points that will be written in each section and subsection of the paper, (3) ideas on what kind of figures or tables that will be included.

4/26 (Tue) during class: final project demonstration (1)

- Less requirement for teams that present later on this date
- Teams should report post-synthesis results (frequency, throughput, power), and should report an almost-complete progress (intermediate timing/area results, screenshots, etc.) for Innovus (APR). However, teams don't need to report post-APR results.

4/28 (Thu) during class: final project demonstration (2)

- More requirement for teams that present later on this date
- Teams should report post-APR results (frequency, throughput, area).

5/4 (Wed) 5:00pm: Submission of final project report due

- Use IEEE template: http://www.ieee.org/conferences_events/conferences/publishing/templates.html
- Should be exactly 4 pages (no more, no less)

Final project grading criteria

As was announced in the first lecture, the final project is responsible for 50% of the total grade of this course. Information on further breakdown of the 50% is shown below:

− Intermediate report: 8%

− Final presentation: 20%

− Final report: 22%

For the intermediate report (8%), the following things will be evaluated.

− Description of motivation and project scope (3%)

− Overall outline structure (placeholders of plots, etc. are fine), novelty in design, extent in design progress (5%)

For the final presentation (20%), the following things will be evaluated.

− Presentation of motivation and project scope (5%)

− Presentation of novelty in SW or HW design (5%) : anything that you newly considered

− Clarity and organization of PPT slides (5%)

− Depth and quality of reported results (5%)

For the final report (22%), the following things will be evaluated.

− Description of motivation and project scope (4%)

− Description of novelty in SW or HW design (4%) : anything that you newly considered

− Formatting, grammar, writing quality of report (3.5%)

− Clarity, organization and conciseness of report (4%)

− Depth and quality of reported results (6.5%)

Good luck!