Project description 2024

Start Assignment

- Due Dec 14 by 11:59pm
- Points 100
- Submitting a text entry box or a file upload
- Available Oct 24 at 5pm - Dec 18 at 11:59pm

Project: Weight- and output-stationary reconfigurable 2D systolic array-based AI accelerator and mapping on Cyclone IV GX

Part1. Train VGG16 with quantization-aware training (10%)

- Train for 4-bit input activation and 4-bit weight to achieve >90% accuracy.

- But, this time, reduce a certain convolution layer's input channel numbers to be 8 and output channel numbers to be 8.

- Also, remove the batch normalization layer after the squeezed convolution.

e.g., replace "conv -> relu -> batchnorm" with "conv -> relu"

- This layer will be mapped on your 8x8 2D systolic array. Thus, reducing to 8 channels helps your layer's mapping in an array nicely without tiling.

- This time, compute your "psum_recovered" such as HW5 including ReLU and compare with your prehooked input for the next layer (instead of your computed psum_ref).

- [hint] It is recommended not to reduce the input channel of Conv layer at too early layer position because the early layer's feature map size (nij) is large incurring long verification cycles.

(recommended location: around 27-th layer, e.g., features[27] for VGGNet)

- Measure of success: accuracy >90% with 8 input/output channels + error < 10^-3 for psum_recorvered for VGGNet.

Part2. Complete RTL core design connecting the following blocks: (5%)

- 2D array with mac units

- Scratchpad memories for 1. activation & weight (input) SRAM, and 2. psum SRAM (for psum you might need multiple banks)

- L0 and output FIFO (Note you do not use IFIFO in this project because the weight will be given from west to east via L0)

- Special function processor (for accumulation and ReLU)

- On the other hand, a corelet.v includes all the other blocks (e.g., L0, 2d PE array, ofifo) other than SRAMs.

As only corelet.v will be applied on the FPGA board, the above hierarchy helps Part5.

- Measure of success: Completion of your entire core design, and no compilation error after all the connection

Part3. Test bench generation to run following stages: (20%)

- Please use the testbench template (core_tb.v in the "project" directory in git)

- Your testbench has accessibility to the ports of your core.v (So, your testbench is a sort of controller)

- Complete the following stages: (Note you need to verify only 1 layer, which is 8x8 channels, not all the layers)

1) Input SRAM loading for weight and activation (e.g., from DRAM, which is emulated by your testbench)

2) Kernel data loading to PE register (via L0)

3) L0 data loading

4) Execution with PEs

5) Psum movement to psum SRAM (via OFIFO)

6) Accumulation in SFU and store back to psum SRAM

7) ReLU in SFU and store back to psum SRAM

8) Generating text stimulus vector (input.txt, weight.txt) and expected output (output.txt) text files for the squeezed layer as you did in HW7.

9) Apply the stimulus text file to your testbench (core_tb.v) to run all the stages described in Part3.

10) Verify your results are the same as the expected output text file.

Measure of success:

- generation of your stimulus and expected output files, and zero verification error compared output.txt.

- TA will test your design with their own input.txt, weight.txt, and output.txt and it must pass.

Part4. Mapping on FPGA (Cyclone IV GX EP4CGX150DF31I7AD) (10%)

(More details will be given in upcoming classes)

- Installation guideline is given in Pages / Course resources tab.

- Map your corelet.v (NOT core.v) on FPGA via Quartus Prime 19.1.

- Complete synthesis and placement/route.

- Measure your frequency at the slow corner.

- Measure your power with a 20% input activity factor.

- Note this is not frequency competition. This is just for students to go through the entire step.

- This is only required for Vanilla version only. Feel free to extend this part to +alpha if you can show some improvement with this.

Measure of success: reporting the final frequency + power numbers + and specs in TOPs/W, TOPs/mm^2 and TOPS/s.

Part5. Weight-stationary and output stationary reconfigurable PE (20%)

- You are supposed to design a reconfigurable PE, which supports both weight and output stationary, and modify all the corresponding implementations (array, core, and so on) in verilog / testbench / verification.

- Your PE should have some muxes to re-route the data flow given 1-bit control signal while the input / weight / output registers are shared across two different modes.

- Your testbench is supposed to pass the functional test with the 1st convolution layer's input, weight, and activations.

- Note you cannot put all the output channels in your small 2-D array so please map only the first 8 output channel and only 1st eight nij (coordinate) of output feature map.

Measure of success:

- zero verification error of rtl results compared to the estimated results from pytorch sim. (Does not require FPGA mapping)

- TA will test your design with their own input.txt, weight.txt, and output.txt and it must pass.

Part6. +alpha (20% + 5% bonus)

- Add anything else for example:

1) any techniques that you learn from the course, or

2) technique from your own idea, or

3) thorough verification, e.g., for multiple layers or tiled layers, or

4) mapping other networks, e.g., NLP or ResNet, or

5) scalable design, e.g., multi-core for tiled layer processing, or

6) others.

- Since the verification for this part is subjective to your enhancements and ideas, do your own verification as per your changes and present your results

NOTE: If your enhanced RTL can be mapped to FPGA, you can report how much TOPS/watt, TOPS/area and TOPS/s improved over vanilla version can be reported

Part7. Poster and Report (15%)

- Poster days are Dec 3 and Dec 5. Please come to the Henry Booker room in the 2nd floor of Jacobs Hall.

- There is no strict format for poster presentation. Please see the example poster : **Poster example (https://canvas.ucsd.edu/courses/58798/files/13391922)**

- On poster day, elevator speech 3min + 30s Q&A (stop watch will be given + hard stop) per team

- Note if you couldn't finish a certain part, e.g., functional verification, or +alpha by the poster day, you can still present and finish by the report deadline. In this case, only 70% of the score is given to the technique.

- **Note you are supposed to send your zip file of pytorch code and RTL by the end of poster day 11:59 pm to prove your progress.**

(This file does not need to be highly cleaned, but to prove that your progress by the poster day)

- On **Dec 5th**, once the poster session ends, course evaluation & quiz will happen in the class (bring your hand-written note and laptop).

- Focus on your unique strength (+alpha part)

- Skip general intro / general motivation / do not explain common parts

- Explain idea concisely and prove the efficacy

- Summary table to show:

1) Frequency, power, accuracy, and other specs (TOPS/w, GOPs/s, # of gates)

2) Verification result

3) Benefit of your idea

- The report needs to be documented with proper **explanations for each part of the project**. Keep the explanations precise.

- Only a member in each team needs to submit the report (instead of 5 identical reports).

- The report should have what is done, what results are observed, and what is your inference for **each part of the project.**

- The report page min and max limit is 2 and 5 pages, respectively. So add your inference results accordingly.

NOTE:

- Report needs to be submitted while you do not need to submit the poster.
- The description in this page could be updated for better clarity later.
- FAQ will be maintained below:

The final deliverables are as follows:

- **One PDF file** containing your final report.
  - The report should clearly **address each step** you took during the design process and **explain your innovative techniques** for the alpha part (**attach any necessary figures/screenshots of your codes**). To ensure receiving full credit for each step, **include any necessary data to measure your success in achieving the required goals**. There is no format requirement for the final report, but the **maximum page limit is 5**.
- **One zip file** includes **one folder for your verilog codes** and **one folder for your notebook files**. You **do not need to upload your VGG model**.

The Deliverables are **DUE** on **Dec 14th at 11:59 PM**. The last day to submit the deliverables is **Dec 18th** with the Late Submission **penalty of 20%** being applied as each day passes.

FAQ:

1. Some techniques that I am implementing are hard to show the benefit by Quartus Prime. How can I quantify?

- Indeed, some of your techniques cannot be measured through the tools given in this course. In such a case, please quantify in a reasonable way, e.g., calculate your benefit theoretically or through any other experiment to prove it. Or, search for a related paper to estimate the benefit in a similar situation.

2. Do I need to prepare both posters and separate slides for the presentation?

- No, only a poster is needed. You are supposed to explain with your poster figures.

3. What is a corelet?

- Corelet includes all the blocks except core, e.g., ofifo, L0, PE array. For part4, only corelet (not core) is required to be implemented.

4. Given the target function and +alpha part, may I edit the ports of core?

- Yes, the core and tb are just a template to help students. Feel free to edit.

5. Memory size can be modified?

- yes.

6. How to use Quartus?

- Install guideline is on Pages/Course resources tab.

7. May I create my own dual port memory ?

- Sure.

8. As we are processing 64 nij indices, our L0 and OFIFO should have 64 depth ?

- No, while it pops out, it can receive the new contents at the same time. so, your depth should not be that high.

9. What is the final output of the hardware simulation and where it should be stored ?

- summation of 9 vectors should pass the ReLU. The output of ReLU is the final result. In the hardware, it should be finally stored in psum mem.

10. Any suggestion on poster size ?

- 48 inch X 36 inch preferred. I also suggest that the file be the size. On powerpoint, you can change the size by going to Design->Slide size-> custom slide size. Otherwise, the printing center had some difficulty in printing.