Class Project –Design a Hardware Accelerator

Project Description

Technology scaling is no longer providing the energy efficiency gain as in the past due to dramatically increasing leakage currents and the end of supply voltage scaling. On the other hand, the design and fabrication costs of integrated circuits are skyrocketing as a result of the ever-growing difficulty in verification and the increasing complexity of CMOS lithography. As a result, future applications call for the underlying hardware drivers to have both energy efficiency and flexibility to address cooling and battery life issues, as well as to maintain a sustainable market growth.

The objective of our class project is to design a hardware accelerator for a specific or a domain of algorithms that will outperform software counterparts running on general-purpose computing platforms (e.g. CPU, GPGPU) by orders of magnitude in terms of energy/area efficiency or throughput (or most likely both). As a result, such a hardware accelerator will significantly improve target applications or enable new capabilities that are not possible by using software solutions.

The hardware design should be conducted by using high-level synthesis tools, e.g. the Synopsys SMC or Vivado SysGen based Simulink flows, that we introduce in class. You should focus the design effort on modeling and exploration on the algorithm and architectural levels, which have the biggest impact on implementation results. You should best utilize the Simulink environment and the CAD tool flow to minimize your effort on design verification and circuit-level optimizations. Doing part of the design (e.g. control FSM) in HDL codes (can be imported into Simulink models) or fine tuning the generated HDL codes is allowed as long as it helps you achieve your design targets. **However, doing the entire design using HDL is NOT allowed.**

You can explore hardware architectures for communication, multimedia, and biomedical signal processing, machine learning, and other domain of algorithms or applications. The topics can be derived from your own research or open literature. Again, **focus your design effort on algorithm modeling (with hardware-awareness) and architectural exploration.**

Project Schedule

Phase 1: Team Sign-up | Due: Feb. 23rd (Tue) 2017, 11:59pm

Team up with your classmates and sign up the same group on Blackboard -> Group. **The default group size is 3 students.** If your group size is greater than 3, you need to justify why additional people are needed, and proportionally more design work will be expected.

Phase 2: Proposal | Due: Mar. 16th (Thu) 2017, 11:59pm

Format: 1 page (excluding references), Times New Roman, 12 point, single spacing.

You are encouraged to submit your proposal and start to work on your project early.

The proposal should include:

1. Project title, group number on Blackboard, names of group members
2. Problem description and prior work (cite references)
   - What is the algorithm to be implemented?
   - What are the target applications?
   - Why is a hardware accelerator needed for the target applications? What are the limitations of software solutions? Or existing hardware solutions if any?
3. Proposed work and expected deliverables (be realistic - you have about 7 weeks!)
   - What is your implementation target? FPGA, ASIC, or both?
   - What are your design specifications? e.g. target throughput, power, energy efficiency, etc.
   - **Note that your implementation targets and design specifications should be driven by the target application. You may want to optimize the design towards a single or multiple metrics depending on your application requirements.**
   - If successful, what difference will your design make to the target applications?
4. References

Special Request of High-End FPGAs

If you plan to work on the FPGA mapping of large algorithms/systems in your project, the following high-end FPGAs in our research lab can be offered upon request for research purposes.

- Xilinx Virtex-7 FPGA VC707 Evaluation Kit
- Altera DE4 Development and Education Board
- DE5-Net FPGA Development Kit (OpenCL support)

To request the high-end FPGAs, add a "**Special Request of High-End FPGAs"** section at the end of the proposal (not subject to the page limit) to express your interest. Explain why a particular high-end FPGA could benefit the proposed work.

Phase 3: Intermediate Report | Due: Apr. 13rd (Thu) 2017, 11:59pm

Format: 2 pages, 2-column IEEE format

Time stamp your progress by documenting any initial results. By this time, you should have the 1st version of your design working properly. This could be your reference point for the architecture exploration later on so that you could see the trajectory of how your

design gets improved over time with different techniques (that you propose) applied.

The intermediate report is optional, and it will not be graded. It is just to help you check the milestones and prepare for the final report.

Phase 4: Final Presentation | Week 15-16 in class

Format: 11 min (9 min presentation + 2 min Q&A will be timed and enforced), max 15 slides in total (including title slide)

- Presentation schedule will be arranged randomly and announced on Apr. 6th.
- **Recommended organization:**
  - 1 slide: project title, team number, name of team members
  - 3-4 slides: Introduction.
    Explain what you are implementing, why a HW accelerator is needed, and what is your design target or specification?
  - 5-7 slides: Technical Approach
    What is the architecture you come up with, what techniques have you applied to optimize your design towards the final target?
  - 3-5 slides: Results
    Show the quantitative assessment of your design, how does it compare to existing (software) solutions, and what do the results mean/imply to the target application?
- The presentation will NOT be graded based on the level of completion of the project by the presentation date. Rather, it will be graded based on the clarity and compellingness of
  1) Motivation: why a HW accelerator is needed?
  2) Technical approach: how is the design optimized towards the objective?
  3) Preliminary Results: quantitative assessment of the HW accelerator showing the impact.
- Submit your slides on Blackboard by 11:59pm the day before your presentation date.
- File name: Group-#.ppt(x)

Phase 5: Final Report | Due May 2nd (Tue) 2017, 11:59pm

Format: 4-5 pages (excluding references), 2-column, IEEE format

- File name: Group-#.pdf

This should be a well-balanced 5-page report:

- 1-1.5 pages | introduction and design specification
- 2-2.5 pages | algorithm/architecture design & optimization
- 0.5-1 page | implementation results

- 0.5 page | conclusion
- No limit | references

In the implementation result section, make sure to show how your design compares to existing (software) solutions, and what the results mean/imply to the target application.

Presentation Tips

- Have a story first. Presentation = story telling.
- Make your slides simple. **Less is more.** Delete everything that is not mentioned or not related to the main point to avoid distraction or confusion.
- Try to utilize figures and videos to explain concepts, not text. Human brain is wired to understand movies and graphics.
- Texts should not cover >10% of the real estate on a slide. Three text bullets per slide is MAXIMUM. Utilize the title as your first text bullet to convey key information.
- Only use short text bullet to reinforce your key point/result/number etc. Do NOT put long lines and read your slides!
- Speak in simple language (e.g. Steve Jobs vs. Bill Gates)
- "Speaking in jargon carries penalties in a society that values speech free from esoteric, incomprehensible bullshit. Speaking over people's heads may cost you a job or prevent you from advancing as far as your capabilities might take you otherwise.", from Gallo, Carmine (2009-09-11). The Presentation Secrets of Steve Jobs: How to Be Insanely Great in Front of Any Audience (Kindle Locations 1939-1941). McGraw-Hill. Kindle Edition.
- You can calculate the clarity (low score = high clarity) of your talk online at http://www.UsingEnglish.com
- Speak in short modules to maintain interest.
- End by calling to action. Conclusion is NOT summary. A conclusion should tell what one should learn or take away from this talk. A summary is simply what have been presented. Note the difference.

Example Projects from Spring 2016

Throughput-Optimized OpenCL-based Heterogeneous Architecture for Large-Scale Machine Learning Algorithms
Power, Area and Throughput Characterization of Triangle Traversal Algorithm
High Performance CRC calculation hardware accelerator for networking applications
HARDWARE ACCELERATION FOR SEQUENCE ALIGNMENT ALGORITHMS – GENOMIC SEQUENCE MATCHING
Canny Edge Detection Algorithm Implementation
Hardware Accelerator for Image Compression using Wavelet Transform
HARDWARE ACCELERATOR FOR RADIX - 2 FFT IMPLEMENTATION
Towards an Hardware Accelerator for Deep neural networks
FPGA Implementation of Encryption and Decryption Algorithm based on AES
Smith-Waterman Accelerator for Genetic Sequence Alignment
Low power, High Throughput FFT Accelerator for Mobile Wireless Applications
Hardware Implementation of Low Power Lightweight Crypto Algorithm for Bioelectronic Implants
Stencil Engine for Video Streaming Canny Edge Detection
Bank Vault Security System.
FPGA based Hardware Accelerator for Options Pricing
Hardware Design of 2D Convolution Filter and Comparison with its Software Counterpart
FPGA based hardware accelerator for Huffman Encoding
Hardware implementation of Recurrent Neural Network on FPGA
Performance Optimization of Discrete Wavelets Transform Based Image Watermarking Using Genetic Algorithms
Hardware accelerated parallel string matching
Hardware accelerated Implementation of Gesture Recognition using DNN
FPGA BASED HARDWARE ACCELERATOR FOR FINANCIAL APPLICATIONS

IMPLEMENTATION OF A HARDWARE ACCELERATED PID CONTROLLER
Hardware Accelerator for Point Multiplication of Elliptic Curve Cryptography engine
Implementation of Feature Extraction Technique Based On Short Time Fourier Transform
Convolution and Stochastic pooling algorithm
2D Discrete Cosine Transform (DCT) using Matrix Data types
Hardware implementation of clutter filter for portable medical ultrasound
The FPGA Implementation of 128-bits AES Algorithm Based on Parallel Operation
Time to digital converter for time of flight application
IMPLEMENTATION OF 32 AND 64 BIT FLOATING POINT ALU
IMAGE SEGMENTATION USING WATERSHED ALGORITHM
Implementation of Discrete Wavelet transform on FPGA targeting Iris recognition
FPGA based Edge detection in Video
MapReduce Acceleration on FPGA
FPGA Hardware Acceleration for the Ising Model using a Monte Carlo Simulations