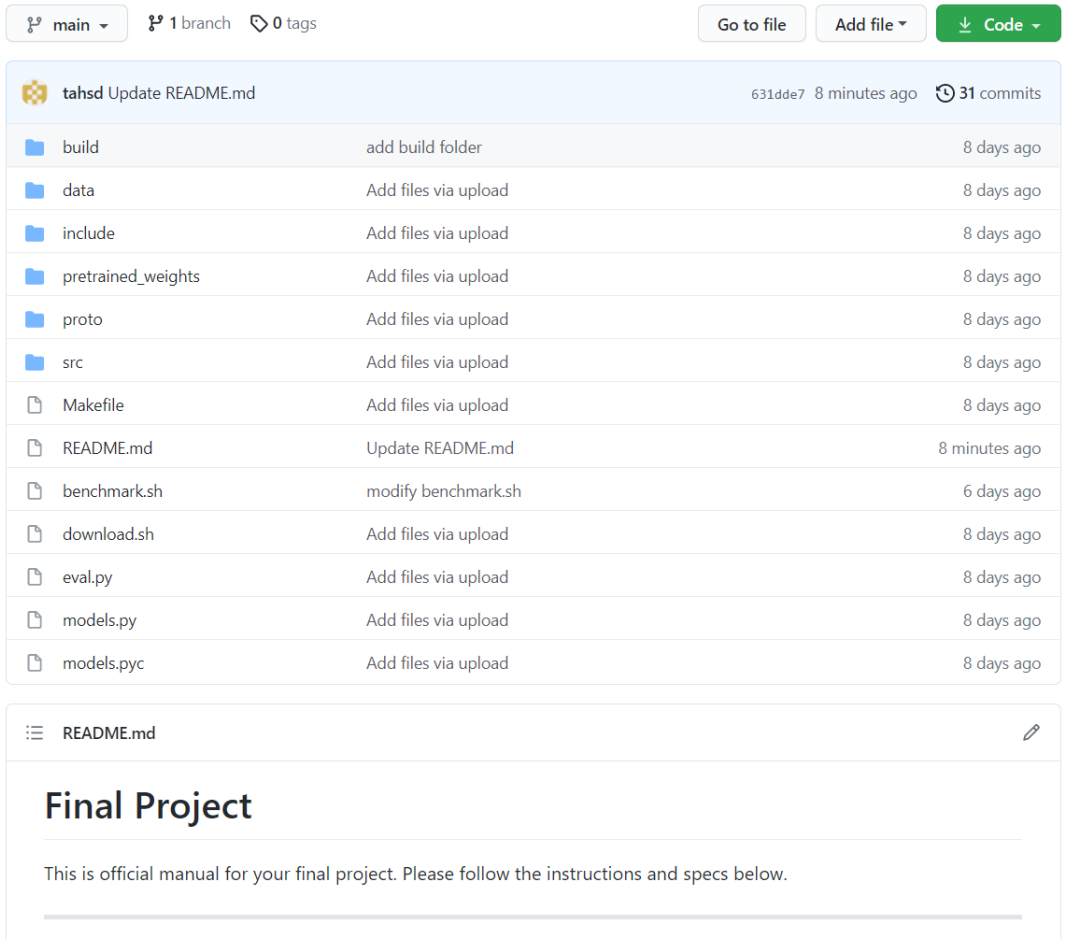
Final Project

- V0 & Optimization

Computing Memory Architecture Lab.

Github Repository



https://github.com/tahsd/hsd21_project

Final Term Project Notification

- **Term Project V0 & Optimization**
  - Term Project V0
    - Implement 8x8 Matrix-Matrix multiplication accelerator
  - Term Project Optimized Version (Optional)
    - Optimize your work

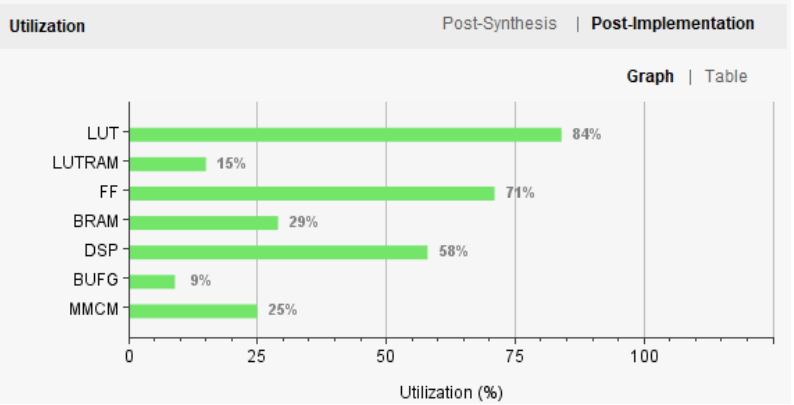
- **Optimization methods**
  - We suggest three ways to optimize your work : DMA / Quantization / Zero Ski pping
  - DMA
    - It boots data transfer speed between DRAM and BRAM.
  - Quantization
    - Quantize both your activation and weights to 8-bit integer.
  - Zero-Skipping
    - Avoid multiplication with zeroes to save computation time

Optimization (1) Quantization

- Modify your SW/HW codes for INT8 quantization.

- Upgrade your IP
  - Switch 32b DSP with 8b integer multiply-adder
    - Recall lab3: multadd IP
  - Dataset and CNN model will be given in INT8 format

- Report the results
  - Performance gain
    - Compare computation time between fp32 and INT8 ver.
  - Utilization
    - After implementation



Optimization (2) Zero-skip

- Modify your SW/HW codes for zero-skipping.
  - Reduce zero data operations
    - Zero data transmission is redundant
    - Zero data calculation is redundant

- Report the results
  - Performance gain
    - Compare computation time between and after each project
  - Utilization
    - If you modify hardware

Optimization (3) DMA

- Try to build your system with DMA
  - Modify the SW codes and HW block design

- Report the results
  - Performance gain
    - Compare computation time between and after each project

Rules for optimization

- You can try more than one optimization,

but do not apply multiple optimizations at the same time.

- If you have done quantization & DMA, for example, submit thre e verions of your work.
  - V0 (Baseline)
  - V0 + quantization
  - V0 + DMA
  - You should include the same V0 for all the versions.

- Each version of your work will be evaluated independently.

Schedule

- Important schedule
  - 5/31 : Final exam
  - 6/2 : Interim submission
  - 6/19 : Final submission

Scoring Criteria

- Implementation
  - Baseline
  - Once you succeed any implementation of those optimizing methods, you will get additional score regardless its performance.
- Inference time
  - Baseline
  - You will be scored on the speed gain by using each optimizing method.
- Accuracy
  - Baseline
  - Accuracy should be robust enough to quantization and zero-skipping.
- Report

Report

- Explain SW/HW System that you implemented.
- It is sufficient to include only project-specific contents.
- You must include the results and analysis of your work
  - inference time, total_image, accuracy, etc.
- If you implement any optimized version, compare the results between V0 and the optimized one and include your analysis in the report.
- Your submission should reproduce the same result as in the report.
- Either in Korean or in English
- # of pages does not matter
- **PDF only!!**

Final Term Project Notification

- Requirements
  - Result
    - Attach all the HW/SW project folders and the bitstream file for the V0 version.
    - If you have implemented optimized ones, attach them additionally. (Refer to Slide 7)
    - Attach a video that can show the results
      - Below is a description of the scenario in which you take a video
        - 1. Keep the camera away and make sure your host computer and ZedBoard are all visible
        - 2. Turn on your ZedBoard
        - 3. Zoom that camera into the host monitor & start minicom
        - 4. Run benchmark.sh
    - Bitstream file's name: "<Team_number>_zynq_<Version>.bit" ex) 7_zynq_V0.bit / 7_zynq_dma.bit
    - Software code folder name: "<Team_number>_SW_<Version>" ex) 7_SW_V0 / 7_SW_dma,
  - Report
  - **Result + Report to one .zip file**

- Upload (.zip) file on ETL
  - Submit one (.zip) file
    - [Project]name1_name2.zip
    - Due: 6/19(SAT) 23:59
    - **No Late Submission**