End-to-End Demo

Kalhan Koul

Stanford University

Docker Demo – Gaussian Blur

- Now let's tie everything we have learned together!
- Use a filter and convolution to blur an image!

Original image

Gaussian Blur filter applied

- Use a small CGRA with small tile of an image for speed purposes

Docker Demo – Gaussian Blur

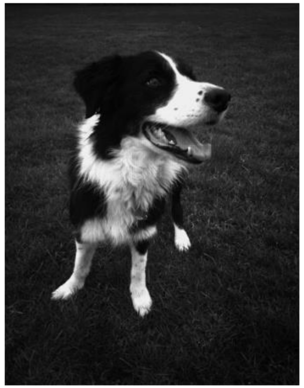
- --> cd /aha/
- Generate hardware for a 4x16 CGRA
- --> aha garnet --width 4 --height 16 --verilog --use_sim_sram --glb_tile_mem_size 128
- --> aha map apps/gaussian
- --> aha pnr apps/gaussian --width 4 --height 16
- --> aha test apps/gaussian (requires **vcs**)

Design Files

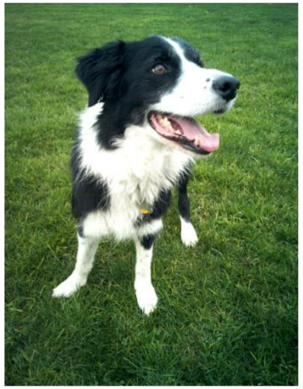
- /aha/Halide-to-Hardware/apps/hardware_benchmarks/apps/gaussian/bin
- CoreIR: design_top.json
- Placement: design.place
- Routing: design.route
- Estimated Freq: design.freq
- GLB configuration: design_meta.json
- Bitstream: gaussian.bs

Something To Try Later – Camera Pipeline

- Processes raw image (sensor data) into RGB image using hot-pixel suppression, demosaicing, color correction, gamma correction, and contrast



Input Image
Raw Image

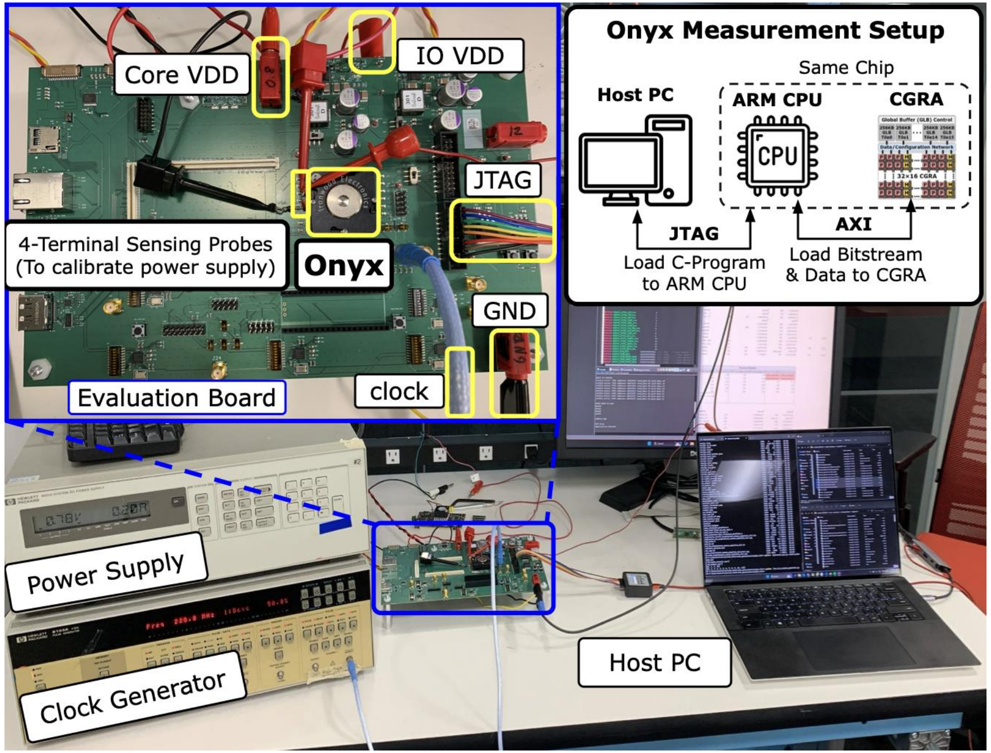


Output Image
Raw Image

Something To Try Later – Camera Pipeline

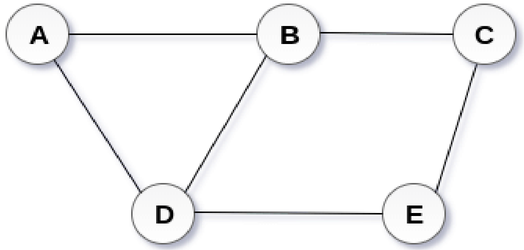
- Remove previously cached RTL
- --> rm garnet/garnet.v
- --> aha garnet --width 32 --height 16 --verilog --use_sim_sram --glb_tile_mem_size 128 (~20 mins)
- --> aha map apps/camera_pipeline_2x2
- --> aha pnr apps/camera_pipeline_2x2 --width 32 --height 16
- --> aha test apps/camera_pipeline_2x2 (~20 mins)

Sparse Application on Chip - Triangle Counter



Sparse Application on Chip - Triangle Counter

- Counts number of triangles in an undirected graph using the tensor expression: *number of triangles =* $\frac{A^3}{6}$
- For large social network graphs, this application is very sparse



Input
Undirected Graph

1

Number of Triangles
Number