OpenVX Overlay using Partial Reconfiguration

Hossein Omidian Guy Lemieux
Department of ECE, University of British Columbia, Vancouver, Canada { lemieux | hosseino } @ ece.ubc.ca

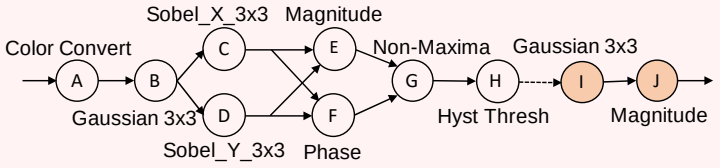
Oveview

- OpenVX (by Khronos Group, same that makes OpenCL)
  - C library and run-time system for Computer Vision
  - Portable, targets many hardware accelerators
  - Think: cache + pipeline friendly version of OpenCV
- OpenVX application is dynamic
  - Arbitrary graph of pipelined “kernels” or nodes
  - Application not known in advance, cannot use HLS
- Overlay for OpenVX (Processor + SVP + CVIs)
  - C language API run on ARM or MicroBlaze
  - Kernels accelerated on soft vector processor (SVP)
  - Many kernels synthesized in advance using Vivado HLS into a Custom Vector Instruction (CVI)
  - Multiple CVIs selected, merged and configured into one Partially Reconfigurable Region (PRR)
- Single-kernel performance example (Magnitude)
  - Baseline before acceleration with CVIs

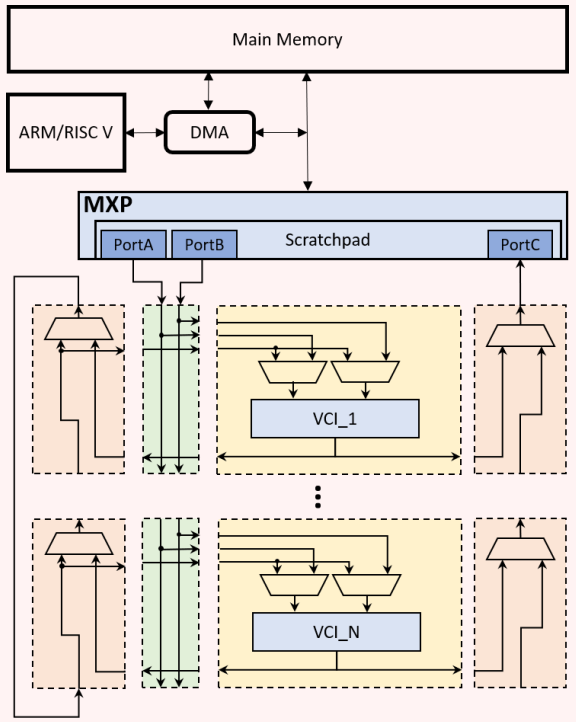
Running Platform	Throughput (MegaPixel/Sec)	Speedup vs ARM
ARM Cortex-A9 (667MHz)	10.31	1.0
VectorBlox SVP-V4 (100MHz / 12,989 LUTs)	65.54	6.3
VectorBlox SVP-V8 (100MHz / 22,517 LUTs)	128.92	12.5
Custom hardware (100MHz / 3,000 LUTs)	1176.04	114

Example OpenVX Application Graph

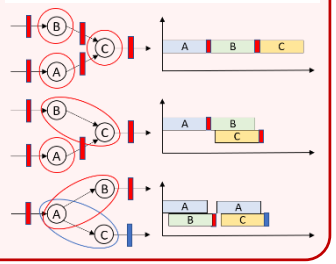
Canny-blur Application



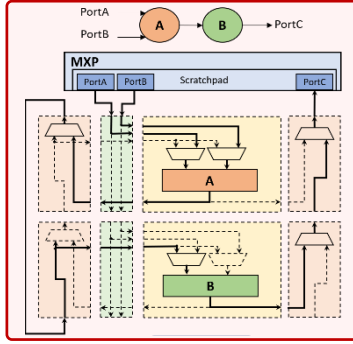
System Architecture



CV kernel	ARM	SVP	HLS
Color Conversion	✓	✓	✓
Channel Extract	✓	✓	✓
Gaussian filter	✓	✓	✓
Sobel filter	✓	✓	✓
Phase	✓	✓	✓
Magnitude	✓	✓	✓
Non-maxima Suppression	✓	✓	✓
Thresholding	✓	✓	✓
Median Filter	✓	✓	✓
Box Filter	✓	✓	✓
Channel Combine	✓	✓	✓
Absolute Difference	✗	✓	✓
Accumulate	✗	✓	✓
Accumulate Squared	✗	✓	✓
Accumulate Weighted	✗	✓	✓
Arithmetic Addition	✗	✓	✓
Arithmetic Subtraction	✗	✓	✓
Bitwise AND	✗	✓	✓
Bitwise OR	✗	✓	✓
Bitwise XOR	✗	✓	✓
Bitwise NOT	✗	✓	✓
Convert Bit Depth	✗	✓	✓
Dilate Image	✗	✓	✓
Erode Image	✗	✓	✓
LBP	✗	✓	✓
Pixelwise Multiplication	✗	✓	✓
Magnitude-cordic	✗	✗	✓
Sqrt-cordic	✗	✗	✓
Arctan-cordic	✗	✗	✓



Bypassing (Pipelining)

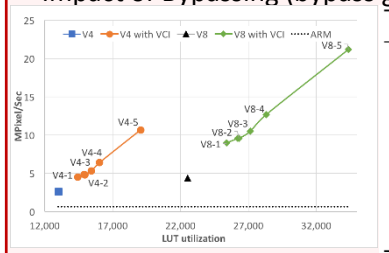


Bypassing

- Adjacent CVIs often put into PRR
- Overlay network can steer data between adjacent CVIs through “configurable AXI-S network”
- Removes writeback to scratchpad
- Increases parallelism

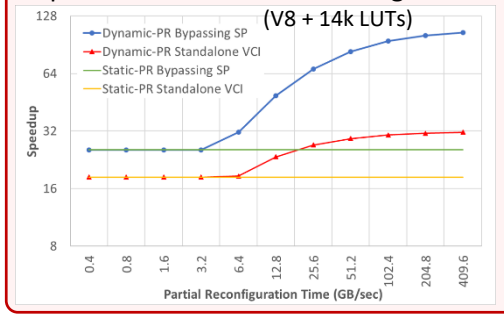
Performance (on Canny-blur)

Impact of Bypassing (bypass groups shown in parentheses)

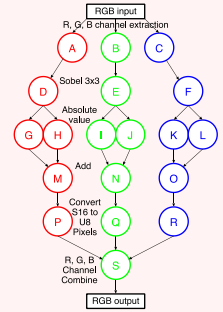


Implementation	Kernels implemented as VCIs
V4-1	(AB)
V4-2	(AB), D
V4-3	(ABC), (ABD)
V4-4	(ABC), (ABD), H, I
V4-5	(ABC), (ABD), H, (IJ)
V8-1	(AB)
V8-2	(AB), C
V8-3	(ABC), (ABD)
V8-4	(ABC), (ABD), H, I
V8-5	(ABC), (ABD), H, (IJ)

Impact of Faster Partial Reconfiguration (V8 + 14k LUTs)



RGB Edge Detect Demo



Conclusions/Future Work

- OpenVX can be implemented as a pure software API
- Basic acceleration using soft vector processor
- Further acceleration using custom vector instructions
  - Bypassing improves pipeline parallelism
  - Faster PR allows more CVIs when hardware limited
- OpenVX Overlay can be part of a larger CNN / ML Overlay

Acknowledgements/References

- NSERC provided funding
- Xilinx, CMC Microsystems provided CAD tools
- VectorBlox provided soft vector processor