VOTA: A 2.45TFLOPS/W Heterogeneous Multi-Core Visual Object Tracking Accelerator Based on Correlation Filters

Junkang Zhu[1], Wei Tang[1], Ching-En Lee[1], Haolei Ye[2], Eric McCreath[2], Zhengya Zhang[1]
[1]University of Michigan, Ann Arbor, MI, USA [2]Australian National University, Canberra, Australia

Abstract

VOTA is a domain-specific accelerator for correlation filter (CF)-based visual object tracking (VOT). It encompasses a Winograd convolution core, a FFT core and a vector core in a high-bandwidth starring topology. VOTA's frame-based instructions and execution enable a 537GFLOPS performance and reduce the code size. An instruction-chaining mechanism permits inter-core pipelining to improve the utilization to 84.2%. A 10.2mm^2 28nm FP16 VOTA prototype incorporating a RISC-V host CPU is measured to achieve 2.45TFLOPS/W at 0.72V. Running OPCF, a CF-based VOT enhanced by adaptive boosting and particle filters, the chip achieves 1157FPS on 640×480 input frames at 0.9V and 175MHz, consuming 296mW.

Introduction

VOT finds practical applications in surveillance, transportation, robotics and human-computer interface. Tracking by detection using a deep CNN is computationally expensive. CF is lighter and faster without detection. CF tracks by correlating filters over an input, and the target is located by the maximum response [1], [2] (Fig. 1). CF is commonly realized in the Fourier domain to simplify correlations to element-wise multiplications, and the filter adaption is done in parallel. CF is among the top performers in recent VOT Challenges [3].

An advanced CF tracker operates on features extracted by a light-weight CNN. For each feature, a CF is learned. To track a set of features associated with an object, multi-channel CFs are learned. After correlation, the multi-channel responses are summed to locate an object. To improve tracking robustness, we apply adaptive boosting to adjust the weighting of the channels. To track fast motion, we add particle filtering [4], a Monte Carlo method, to narrow down the likely target regions. Our design is named oriented particle CF (OPCF) (Fig. 2) and it is representative of an advanced CF tracker. The computational kernels of such a CF tracker are diverse, including convolution (conv), FFT/IFFT, and various real and complex vector operations (Fig. 3). To support online training and Fourier-domain processing, FP16 is required. A custom ASIC lacks flexibility as CF algorithm parameters change, and a GPU does not provide the best efficiency.

We present visual object tracking accelerator (VOTA) to support the class of advanced CF trackers. VOTA contains a Winograd conv core (WINO), a FFT core (FFT) and a vector core (VEC), all in FP16, to support the VOT computational kernels. WINO is optimized for small-kernel conv; FFT is specialized for 2D complex FFT/IFFT; and VEC provides flexible vector operations. VOTA's frame-based instructions reduce the OPCF code size by 34×. By instruction-chaining and inter-core pipelining, the hardware utilization reaches 84.2%.

System Design and Star-Ring Topology

VOTA is integrated with a RISC-V CPU, an instruction and a data memory over an AXI interconnect (Fig. 4a). The RISC-V CPU serves as the host to VOTA. The system is completed with an SPI and a JTAG interface. To overcome the bandwidth limitation of a shared global bus that is common in SoCs and avoid wide crossbars' routing congestion, we choose a star-ring topology to allow the three cores in VOTA to pass data seamlessly through four wide-bandwidth 1Mb memory modules (Fig. 4b). Based on common use cases, VEC is placed in the center to handle vector processing, shaping, pre- and post-processing for WINO and FFT. Each memory module supports 2Kb-wide word access with over 44GB/s bandwidth to each core. A simple arbiter allows two cores to share a memory module in three access modes: real/complex vector or random access.

Frame-based ISA and Microarchitecture

VOTA offers a frame-based ISA (Fig. 5a) to simplify programming its three cores and reduce code size (Fig. 5b). A frame instruction (Fig. 5c) operates on a frame of 64×64 real (FP16) or complex (2× FP16) numbers. The three cores implement frame-based processing by breaking a frame into row operations (Fig. 5d). Multiple row executions are in flight in the processing pipelines.

WINO adopts the Winograd algorithm [5] to minimize the computational complexity of 3×3 conv that is popular in state-of-the-art CNNs. The Winograd algorithm realizes a 3×3 kernel by a 4×4 input conv using only 16 multiplications, a 2.25× reduction over the conventional conv. The Winograd conv is computed by transforming weights and activations, followed by pointwise multiplications and inverse transform (Fig. 6a). A 3×3 kernel is transformed to a 4×4 weight matrix W in precomputation. A WINO unit computes a 3×3 kernel by a 4×4 input tile conv to produce a 2×2 output in four pipeline stages (Fig. 6b): 1) transform a 4×4 input tile to F, 2) pointwise multiplication of $W \odot F$, and 3) inverse transform in two stages. WINO contains 32 units to process 32 overlapping 4×4 input tiles at a time, constituting 4 input rows in a frame (with padding) (Fig. 6c). In every cycle, two input rows are fetched, and two output rows are produced. Input registers provide input reuse between units and across cycles, and transformed weights are precomputed and broadcasted to the units to provide weight reuse (Fig. 6d). 1×1 conv can be realized by bypassing stages and 5×5 or larger conv can be done by overlapping 3×3 conv.

FFT performs pipelined 64-point complex FFT with transpose buffering (Fig. 7a). 2D 64×64 patch FFT required for CF can be done by two passes through the FFT core with transpose (Fig. 7b). IFFT is computed by reversing the FFT core's pipeline. VEC contains 64 VEC units, each made of a pipelined FP16 floating point unit that performs real and complex add, subtract, multiply, square and divide.

Instruction-Chaining and Inter-Core Pipelining

VOTA allows the cores to operate in parallel independently or in a pipeline. When the cores are pipelined, barrier instruction (BARR) needs to be inserted between instructions with data dependencies, but it results in a low utilization waiting for a core completing a frame instruction. To improve utilization and latency, we allow instructions with data dependencies to be packed and chained in a fine-grained pipeline (Fig. 8a). The instruction-chaining controller utilizes scoreboards to track resource availability to prevent hazards (Fig. 8b). When an instruction is dispatched, its data dependency is checked from the memory status table, and its instruction dependency is checked from the function unit status table. After an upstream instruction produces the first output row, the instruction status table is updated, and the controller releases the downstream instruction without waiting for the entire frame to be ready. The fine-grained inter-core pipelining improves the hardware utilization to 84.2% for the OPCF workload (Fig. 8c) and shortens the processing latency by 63%.

Chip Measurement Results

A 12.96mm^2 28nm chip is fabricated with a 10.2mm^2 core area (Fig. 9). The chip is measured to achieve 537GFLOPS at 0.9V and 175MHz in room temperature, dissipating 296mW (Fig. 10). The chip runs OPCF on 640×480 inputs at 1157FPS with a 0.86ms latency. At 0.72V, the chip achieves 2.45TFLOPS/W (Fig. 11). VOTA is the first programmable domain-specific accelerator for VOT to support advanced CF trackers. No direct comparison is available, but VOTA demonstrates competitive FP16 compute density and power efficiency over recent DNN and vision processors (Table I).

Acknowledgements

The authors would like to thank Ford-UM Research Alliance for funding this work and the TSMC University Shuttle Program for chip fabrication support.

References

[1] D. S. Bolme, *CVPR* 2010.
[2] J. F. Henriques, *TPAMI*, 2014.
[3] M. Kristan, *ICCV* 2017.
[4] P. Del Moral, *J. Royal Stat. Soc.*, 2006.
[5] A. Levin, *CVPR* 2016.
[6] P. N. Whatmough, *VLSI*, 2019.
[7] J. Oh, *VLSI* 2020.
[8] J. Lee, *ISSCC* 2019.
[9] A. Suleiman, *VLSI* 2018.

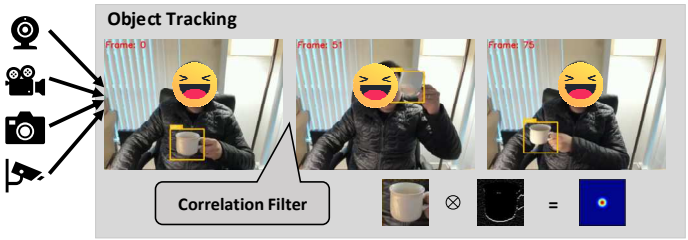

Fig. 1. Visual object tracking (VOT) based on correlation filter.

Kernel	Feature Extraction	CF Training	CF Inference	Adaptive Boosting	Particle Filter
Convolution	✓				
FFT/inverse FFT		✓	✓		✓
Real Vector Addition	✓	✓	✓	✓	✓
Real Vector Multiplication		✓	✓	✓	✓
Real Vector Division		✓			
Complex Vector Addition		✓	✓		✓
Complex Vector Multiplication		✓	✓		✓
Complex Number Conjugate			✓		✓
Matrix Transpose		✓	✓		✓

Fig. 3. The computational kernels of a CF tracker.

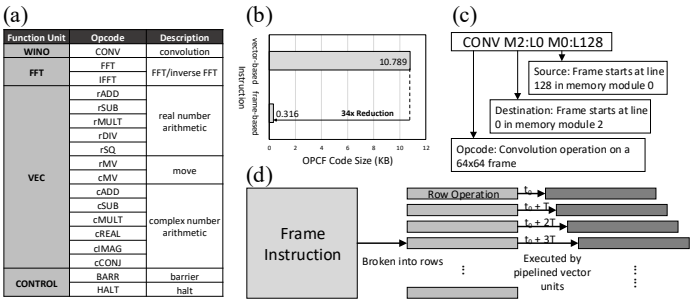

Fig. 5. (a) VOTA's frame-based ISA, (b) reduction in code size by frame-based ISA over vector-based ISA, (c) format of a frame instruction, (d) frame-based processing.

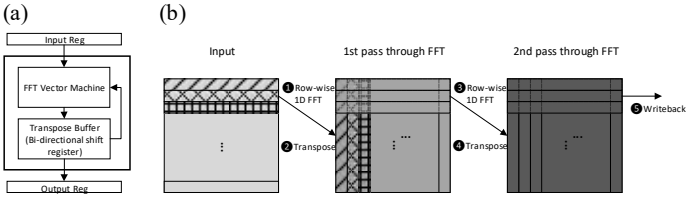

Fig. 7. (a) FFT core with a 64-point complex FFT machine and a transpose buffer, (b) illustration of the execution of a 2D FFT.

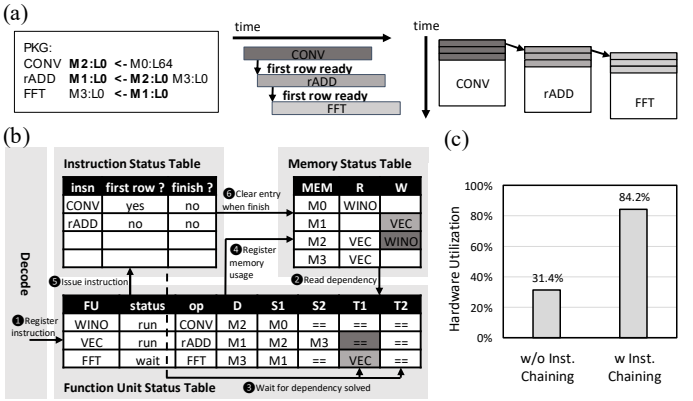

Fig. 8. (a) Illustration of the format and execution of an instruction package, (b) diagram of the instruction-chaining controller, (c) comparison of the utilization with and without instruction-chaining.

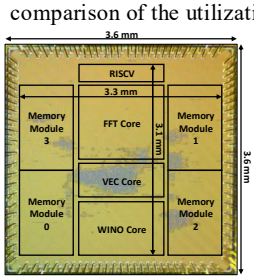

Fig. 9. Chip photograph.

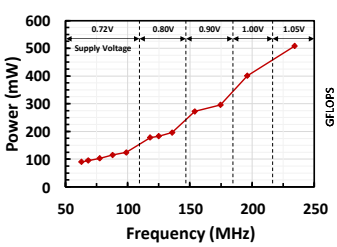

Fig. 10. Power measurements.

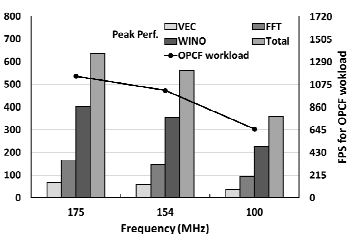

Fig. 11. Performance measurements.

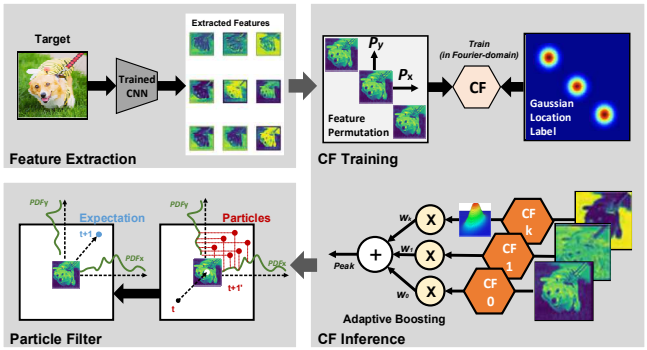

Fig. 2. Oriented Particle Correlation Filter (OPCF) – a CF-based VOT enhanced by adaptive boosting and particle filters.

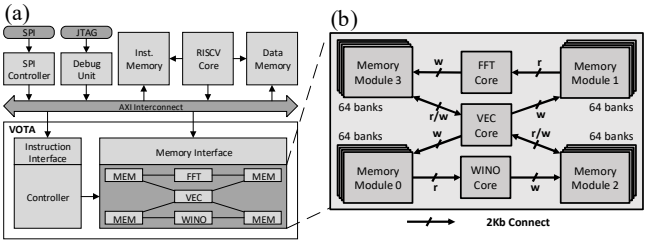

Fig. 4. (a) Diagram of the VOT SoC integration, (b) the high bandwidth star-ring topology.

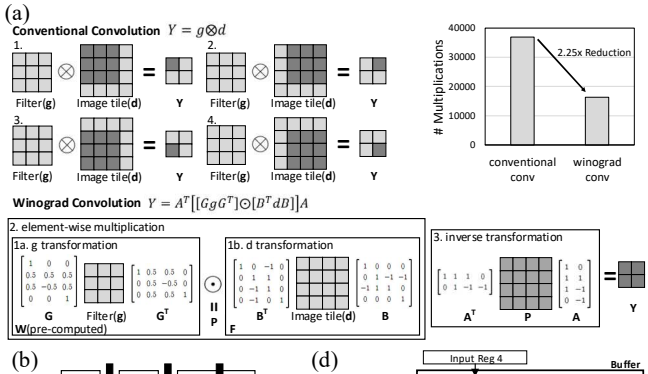

Fig. 6. (a) Comparison between the conventional and Winograd convolution, (b) pipeline stages of a WINO unit, (c) mapping of input tiles to WINO units, (d) the dataflow in WINO and buffer states of input registers.

Table I: Comparison with Prior Works

	This Work	Whatmough[6]	Oh[7]	Lee[8]	Suleiman[9]
Application	Visual Object Tracking	DNN, DSP, Security	DNN Training	DNN Training	Visual-inertial Odometry
Technology (nm)	28	16	14	65	65
Area (mm^2)	10.23	25	9.844	16	16.07
SRAM	576KB	9MB	2MB	372KB	854KB
Frequency (MHz)	60 - 230	353.8 - 732.48	1000 - 1500	50 - 200	62.5**/83.3***
Power (mW)	90 - 508	-	-	43.1 - 367	27
Throughput	193 - 720 GFLOPS (FP16)	10 - 100 GOPS*	2000 - 3000 GFLOPS (FP16)	>300 GFLOPS (FP16)	10.5 – 59.1 GOPS**/ 1 - 5.7 GFLOPS***
Power Efficiency	2.45 TFLOPS/W (FP16)	1.04 TOPS/W	1.4 TFLOPS/W (FP16)	1.74 TFLOPS/W (FP16)	-

*Derived from a figure **Fixed-point in VFE ***Floating-point in BE