ScaleHLS: Achieving Scalable High-Level Synthesis through MLIR

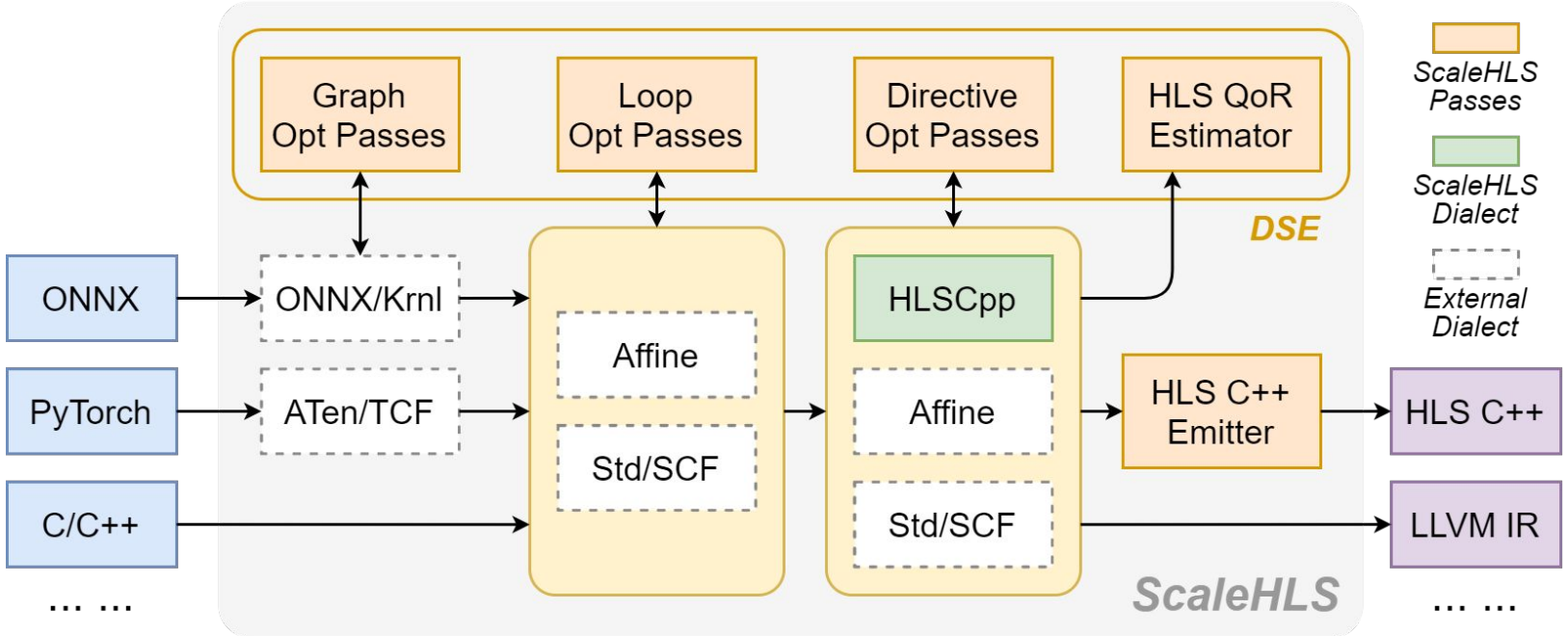
Hanchen Ye[1], Cong Hao[2], Hyunmin Jeong[1], Jack Huang[1], Deming Chen[1]

[1]University of Illinois at Urbana-Champaign, [2]Georgia Institute of Technology

Motivations

- Large HLS designs (with a large number of submodules/loops and complicated interconnections) are difficult to effectively optimize;
- These designs are desired to be optimized at multiple abstraction levels:
  - Graph level: node fusion/insertion, IP/template integration, etc;
  - Loop level: loop tiling, loop fusion, loop permutation, local buffer insertion, etc;
  - Directives level: loop pipelining, loop unrolling, array partition, etc;
- Existing approaches are limited:
  - Non-comprehensive design space (can only represent and optimize one or two abstraction levels);
  - Design space exploration (DSE) algorithm not scalable;
- ScaleHLS: handle large HLS designs through a multi-level representation and optimization based on MLIR.

ScaleHLS Framework

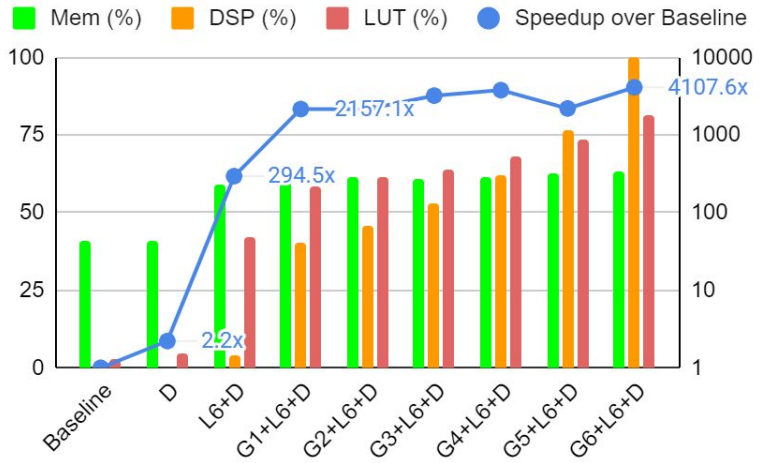


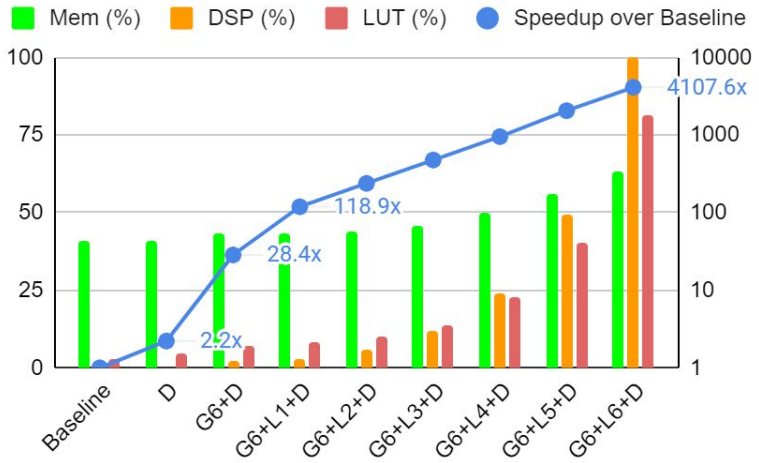
DSE Results of Computation Kernels

Kernel	Prob. Size	Speedup	LP	RVB	Perm. Map	Tiling Sizes	Pipeline II	Array Partition
BICG	4096	41.7×	No	No	[1, 0]	[16, 8]	43	A:[8, 16], s:[16], q:[8], p:[16], r:[8]
GEMM	4096	768.1×	Yes	No	[1, 2, 0]	[8, 1, 16]	3	C:[1, 16], A:[1, 8], B:[8, 16]
GESUMMV	4096	199.1×	Yes	No	[1, 0]	[8, 16]	9	A:[16, 8], B:[16, 8], tmp:[16], x:[8], y:[16]
SYR2K	4096	384.0×	Yes	Yes	[1, 2, 0]	[8, 4, 4]	8	C:[4, 4], A:[4, 8], B:[4, 8]
SYRK	4096	384.1×	Yes	Yes	[1, 2, 0]	[64, 1, 1]	3	C:[1, 1], A:[1, 64]
TRMM	4096	590.9×	Yes	Yes	[1, 2, 0]	[4, 4, 32]	13	A:[4, 4], B:[4, 32]

- Speedup is with respect to the baseline designs only optimized by LLVM optimizations of Vivado HLS.
- *LP* and *RVB* denote *Loop Perfection* and *Remove Variable Bound*, respectively.
- In the *Loop Order Optimization*, the i-th loop in the loop nest is permuted to location $PermMap[i]$, where locations are from the outermost loop to inner.

Ablation Study of MobileNet-v2





- Speedup is with respect to the baseline designs only optimized by LLVM optimizations of Vivado HLS.
- D, $L\{n\}$, and $G\{n\}$ denote directive, loop, and graph level optimizations, respectively. Larger n indicates stronger optimizations are applied.
- The directive, loop, and graph optimizations contribute around 2.2x, 133.9x, and 12.9x speedups.
- **Open-source code and full-length paper will be available in April!**