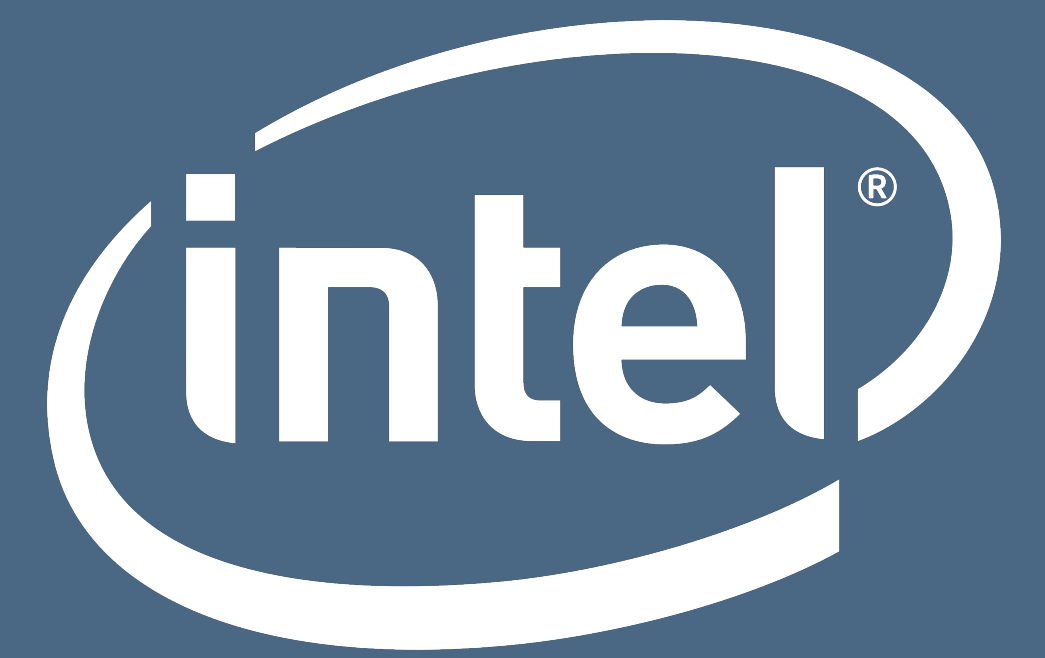


# [SLP] SuperNode-SLP: Vectorizing Chains of Add/Subs

Vasileios Porpodas<sup>1</sup>, Rodrigo C. O. Rocha<sup>2</sup>, Evgueni Brevnov<sup>1</sup>,  
Luís F. W. Góes<sup>3</sup>, and Timothy Mattson<sup>1</sup>

Intel Corporation<sup>1</sup>, University of Edinburgh<sup>2</sup>, PUC Minas<sup>3</sup>

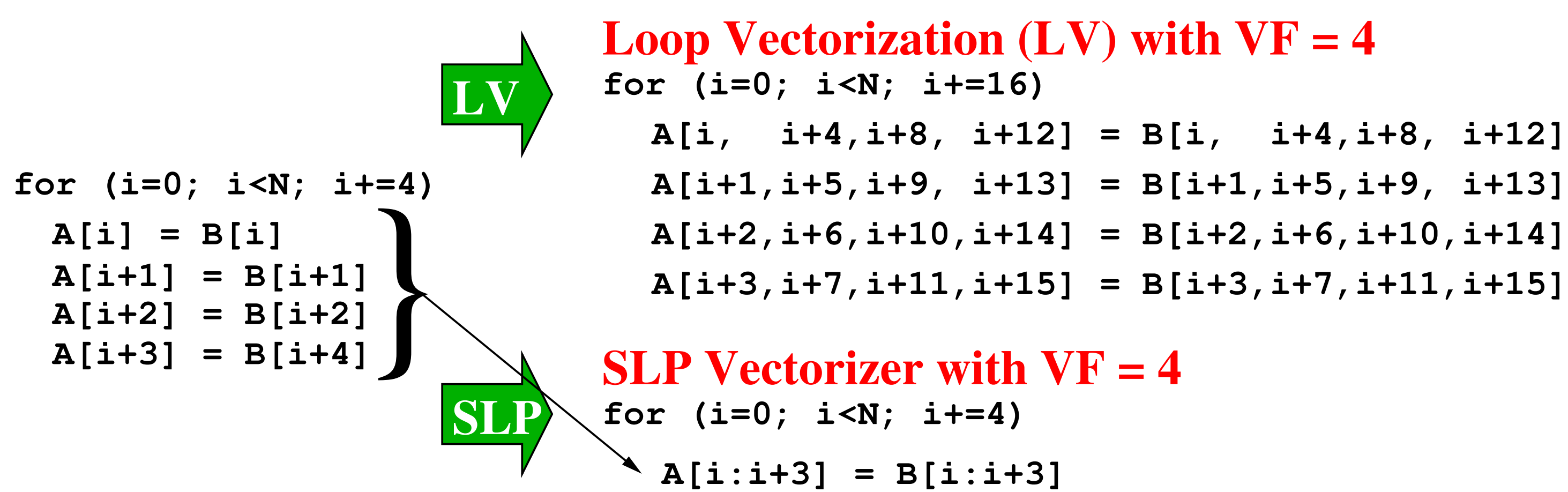


SLP Auto-vectorization converts straight-line code into vector code. Super-Node SLP (SN-SLP) is an improvement on SLP trunk, optimized for expressions that include a commutative operation (such as addition) and its corresponding inverse element (subtraction).

SN-SLP uses the algebraic properties of commutative operators and their inverse elements to enable aggressive operand reordering across groups of instructions, which we refer to as Super-Nodes. Super-Nodes extend the Multi-Nodes of "Look-Ahead SLP", presented in EuroLLVM'18. They form chains of both commutative operations and their inverse operations and allow for legal operand reordering across them.

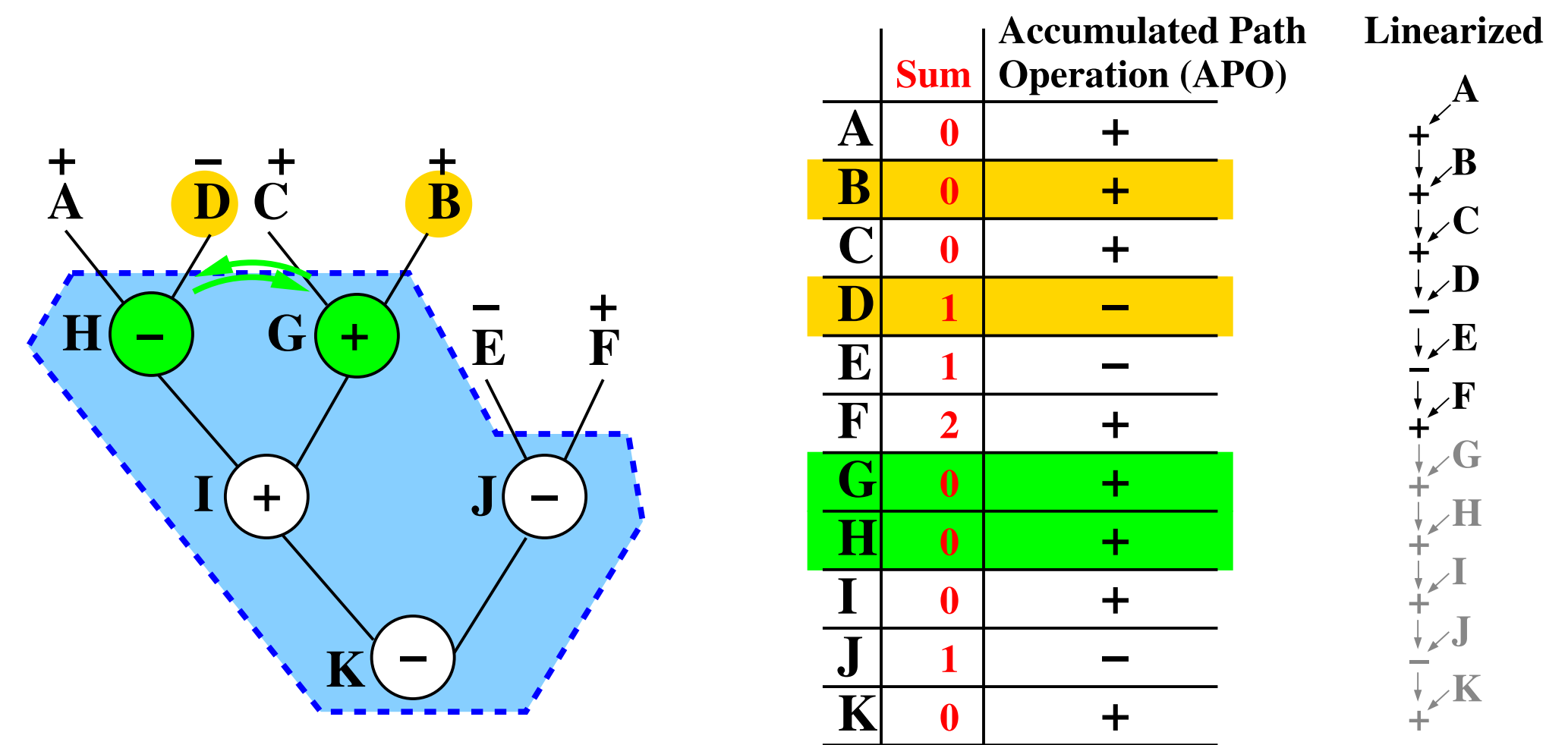
## SLP vs LV

The Loop Vectorizer (LV) is vectorizing across iterations.  
SLP is vectorizing across instructions.

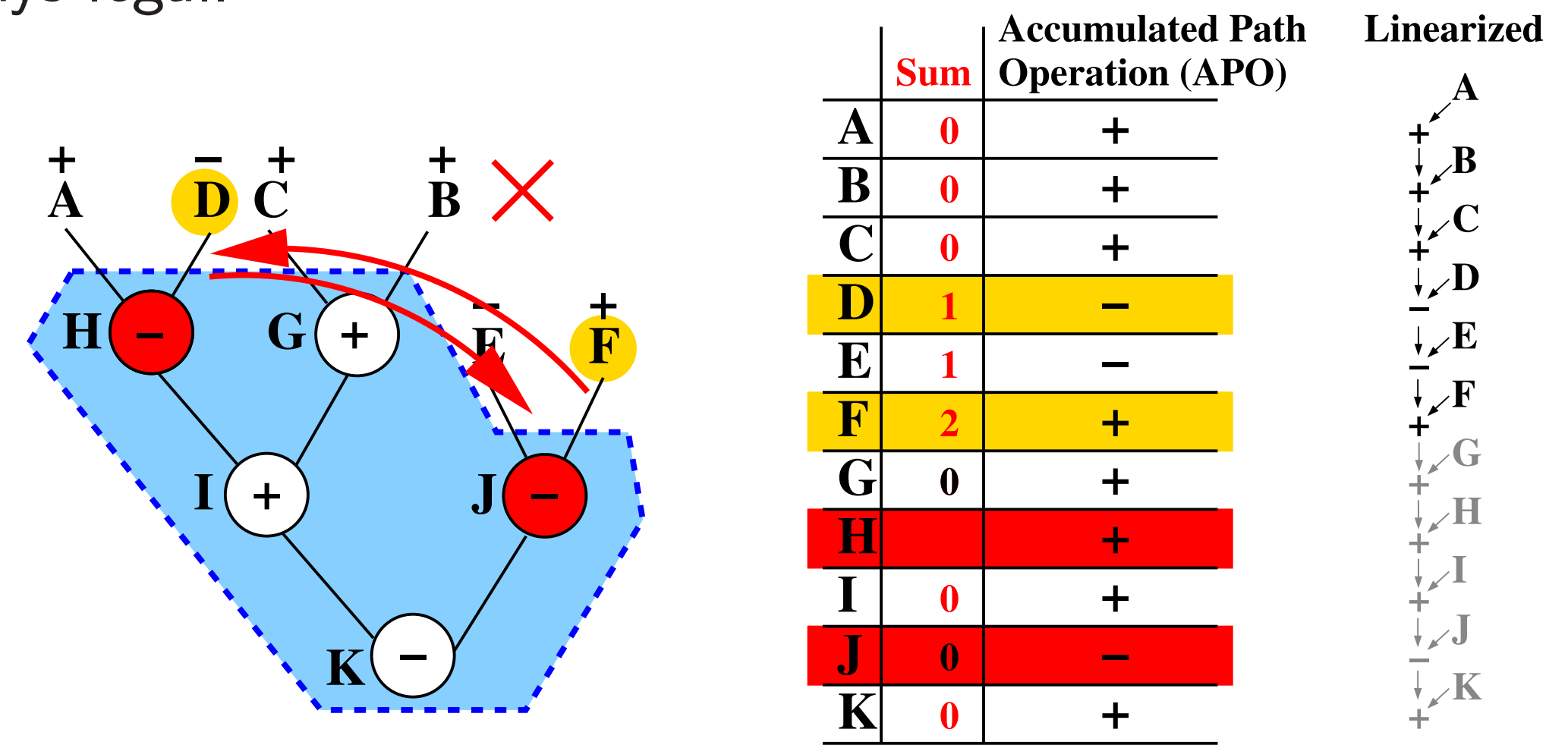


## Legality

Reordering operands (e.g., D and B) is legal if they have the same APO. Even if the APO is not the same, we may still reorder the successors.

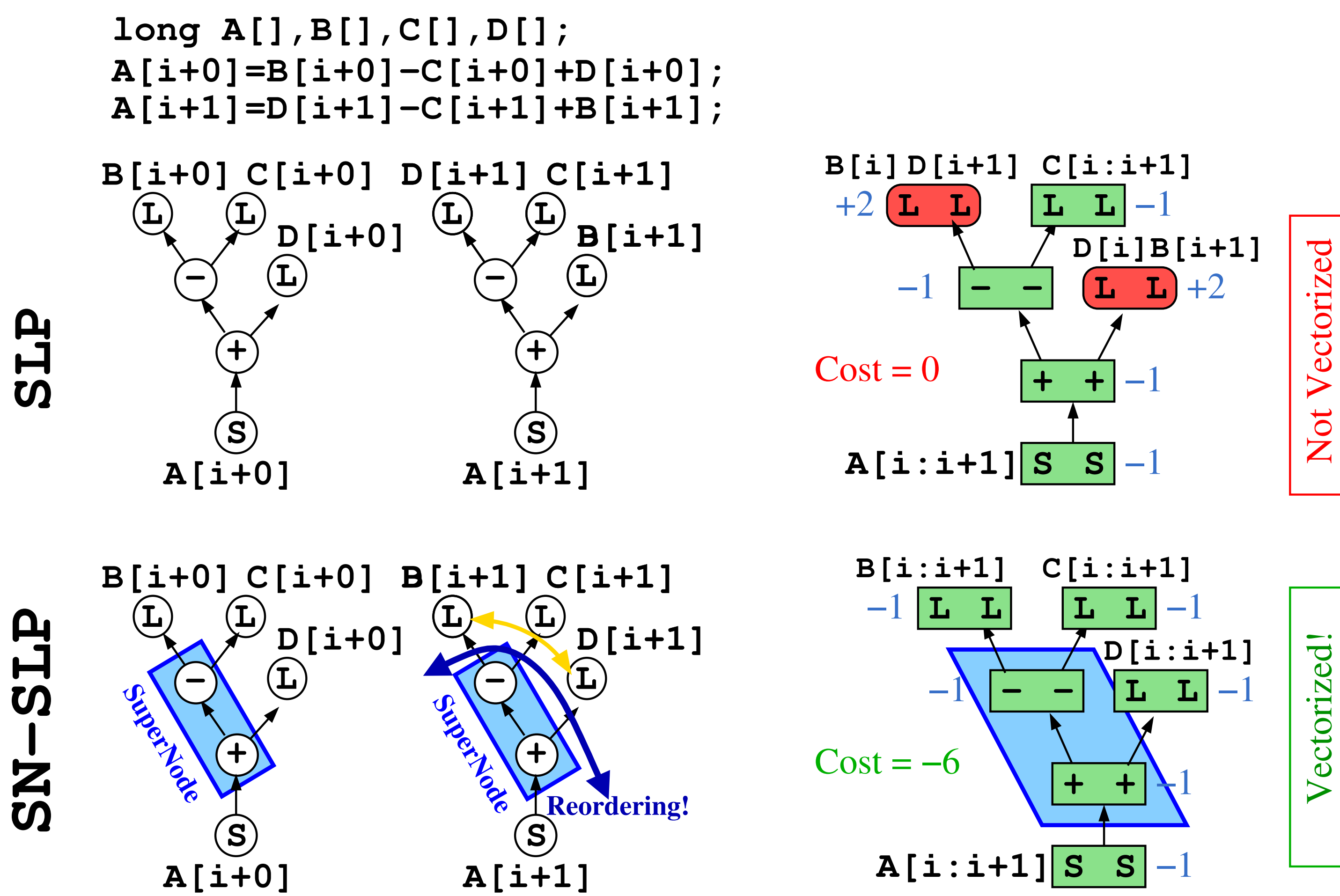


Not always legal.



## SLP Can Fail on ADD/SUB Chains

SLP cannot reorder operands across chains of ADD/SUB (or MUL/DIV). SN-SLP forms a "Super-Node" and reorders across them.



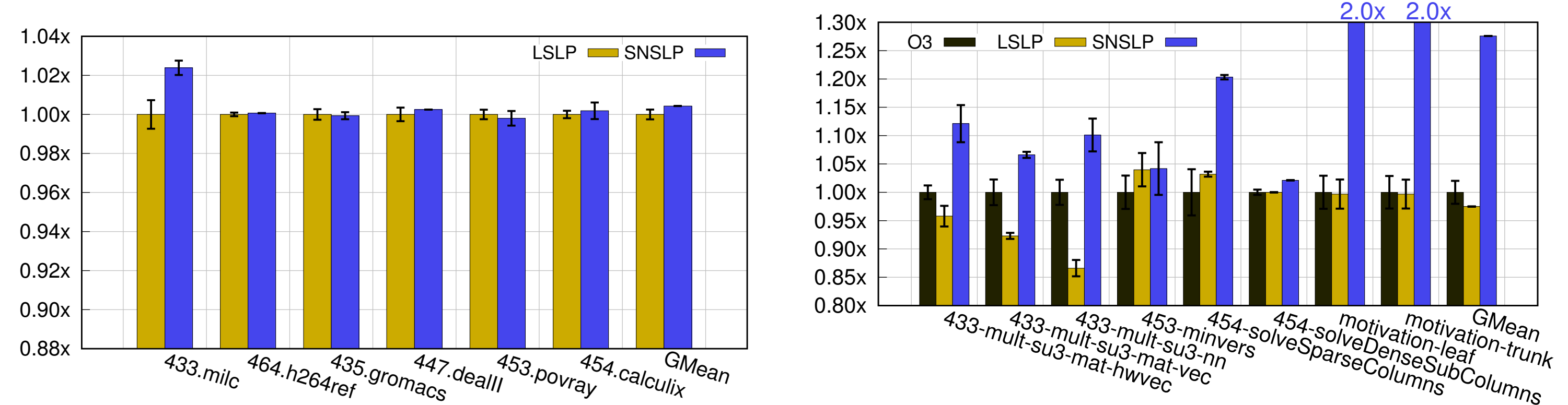
## Performance

Target: Intel® Core™ i5-6440HQ CPU

Compiler flags: -O3 -ffast-math -march=native -mtune=native, CPU2006 O3: Trunk LLVM with all vectorizers disabled.

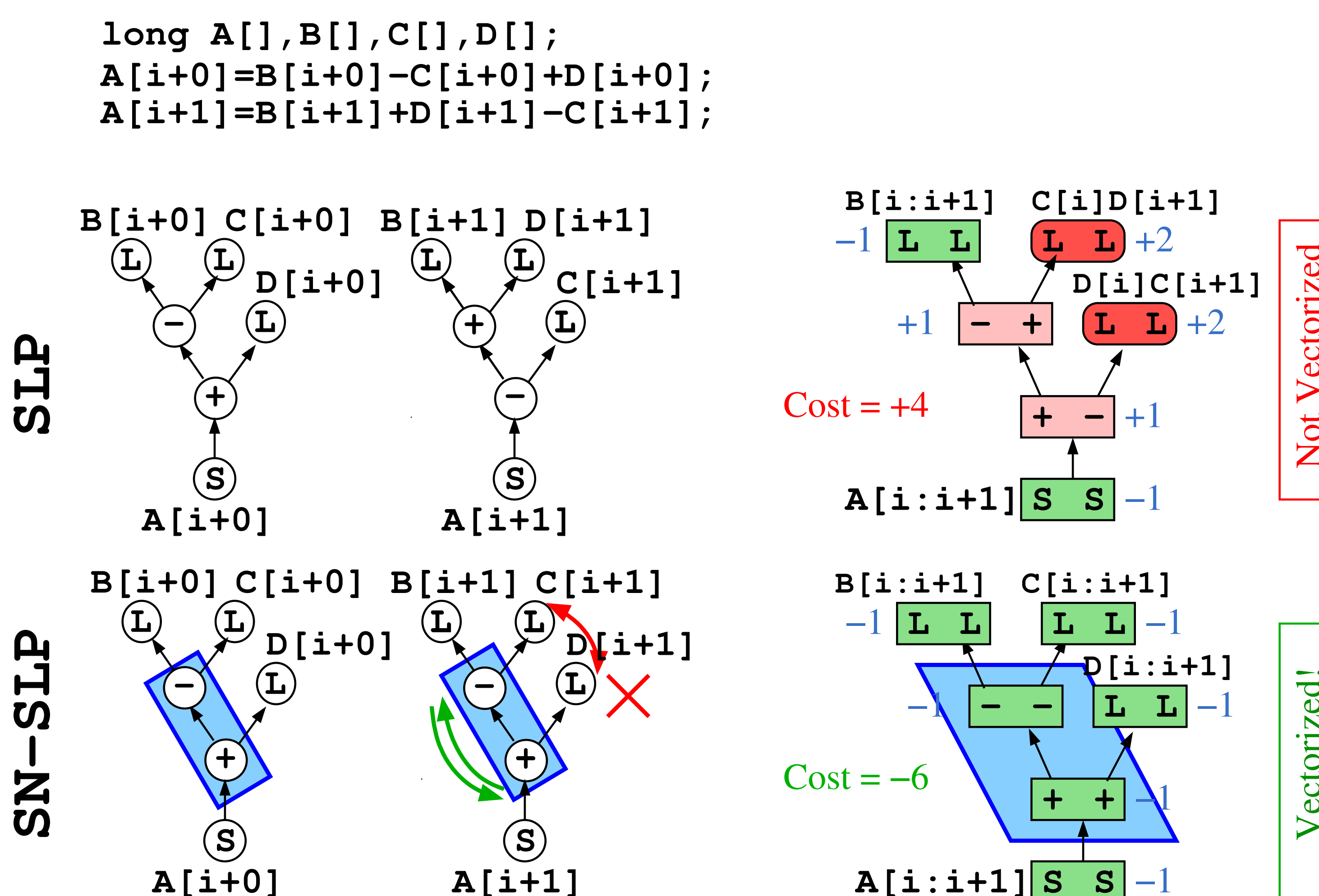
LSLP: Trunk SLP + MultiNodes (patches under review).

SNSLP: Trunk SLP + SuperNodes (patches coming soon).



## SN-SLP Can Reorder Internal Nodes of the Super-Node

When operands cannot be reordered, we can try reordering the internal nodes of the Super-Node.



## We are Upstreaming it!

[SLP] Patches for commutative instructions are out. They are adding support for Multi-Nodes, i.e., chains of commutative operations.

Super-Node patches coming soon!

1. Code refactoring ✓
2. Refactor operand reordering functions
3. Operand reordering across MultiNodes
4. Support for SuperNodes (ADD/SUB)

## Conclusion

SN-SLP improves SLP on code with ADD/SUB MUL/DIV chains. It forms Super-Nodes of commutative operations and their inverse elements. It performs legal operand reordering, guided by the Look-Ahead heuristic.

Please check out our CGO'19 paper:

"Super-Node SLP: Optimized Vectorization for Code Sequences Containing Operators and Their Inverse Elements. Vasileios Porpodas, Rodrigo C. O. Rocha, Evgueni Brevnov, Luís F. W. Góes, Timothy Mattson."

[http://vporpo.me/papers/snslp\\_cgo2019.pdf](http://vporpo.me/papers/snslp_cgo2019.pdf)