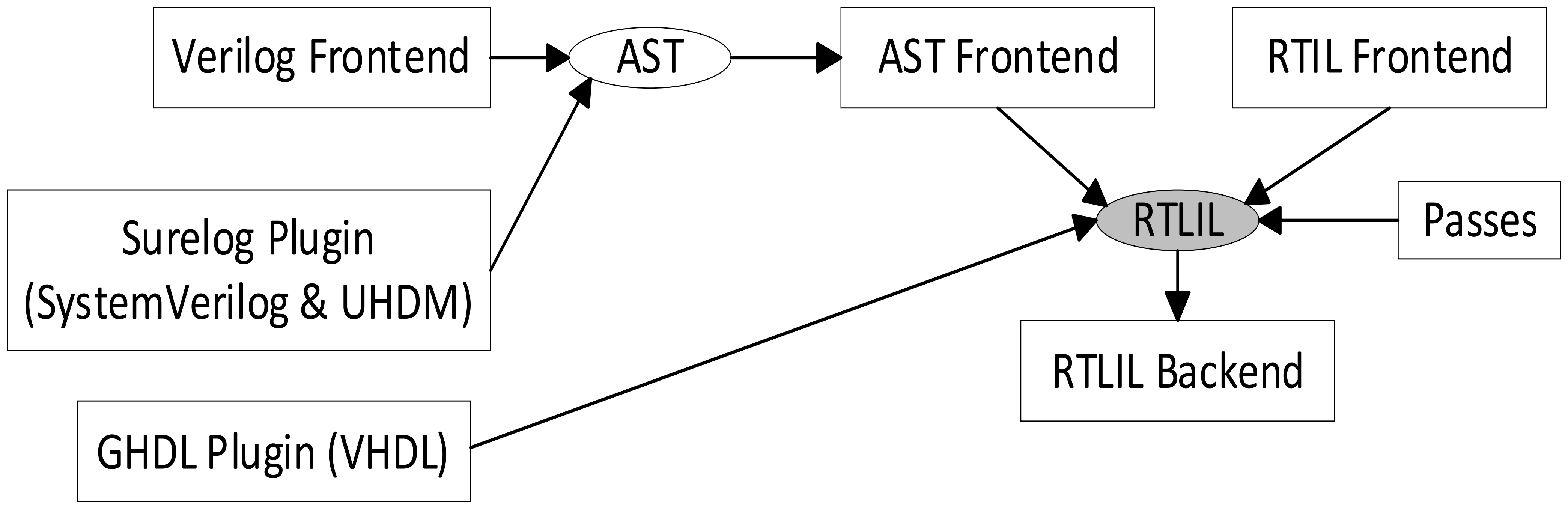
Open Hardware Development With Incremental Synthesis

Alireza Azadi, Kenneth B. Kent
Faculty of Computer Science, University of New Brunswick
al.azadi94@unb.ca - ken@unb.ca

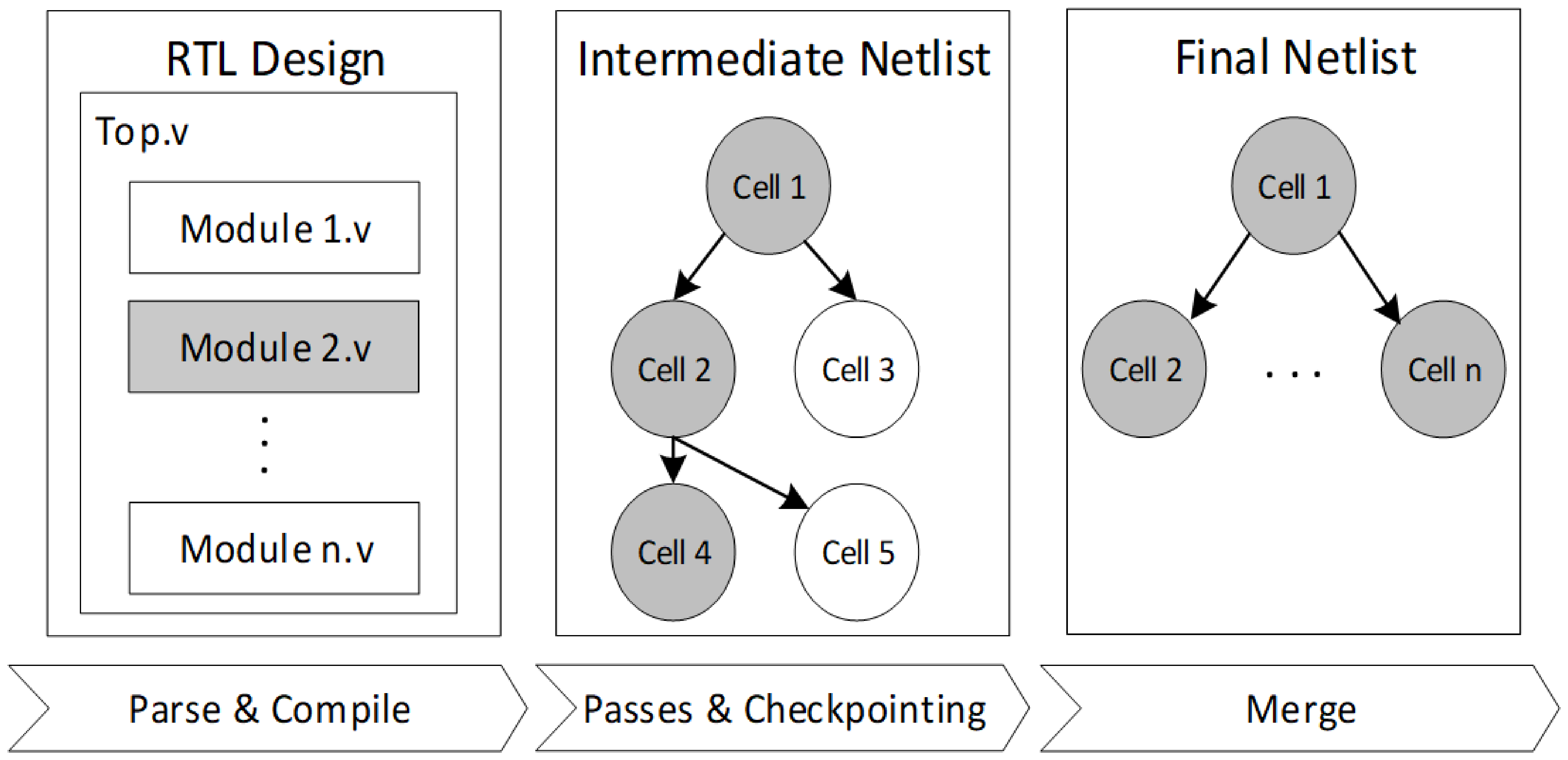
INTRODUCTION

Hardware designs have become more complex, leading to modular designs made up of many submodules. Modifying a small part of the design requires re-synthesizing the entire design, which is time-consuming and computationally intensive. Incremental synthesis is a solution to this problem, where only modified modules are synthesized, and unaffected modules can be reused. Commercial EDA tools such as Intel Quartus and Xilinx Vivado already support incremental synthesis. The proposed method aims to add support for incremental synthesis to the Yosys open-source synthesis suite.

METHODOLOGY



- Bazel build system chosen for adding incremental building support to Yosys.
- Input designs in Yosys data flow are synthesized into a representation of a netlist.
- The proposed methodology targets the netlist level, since, at this specific level:
  - Support for read/write of netlists while allowing arbitrary passes to be applied to them.
  - Enables incremental synthesis of all current and future supported HDL languages.



To effectively implement incremental synthesis the following key steps must be followed:

- Break down the design into smaller modules.
- Generate a dependency tree of the design modules.
- Define a checkpoint structure to reuse unmodified modules.
- Process modified modules and their dependencies.
- Merge all the submodules into a final netlist.

EXPECTED OUTCOMES

Reduced Build Times

Reusability

Scalability

Better Collaboration

Flexibility

Improved Productivity

CAS–Atlantic