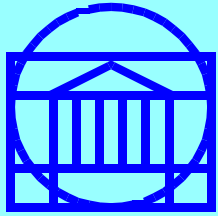


Zephyr
A National Compiler
Infrastructure

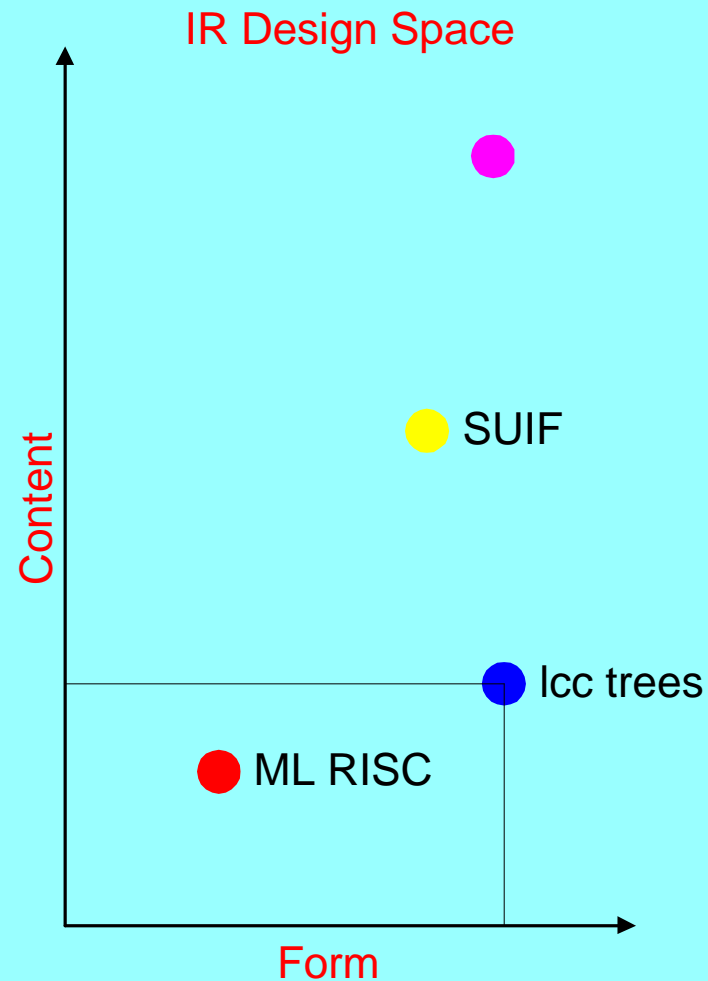
Princeton University
University of Virginia

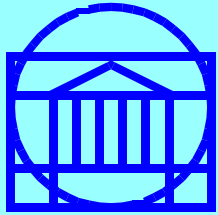


The Zephyr Approach

University of Virginia

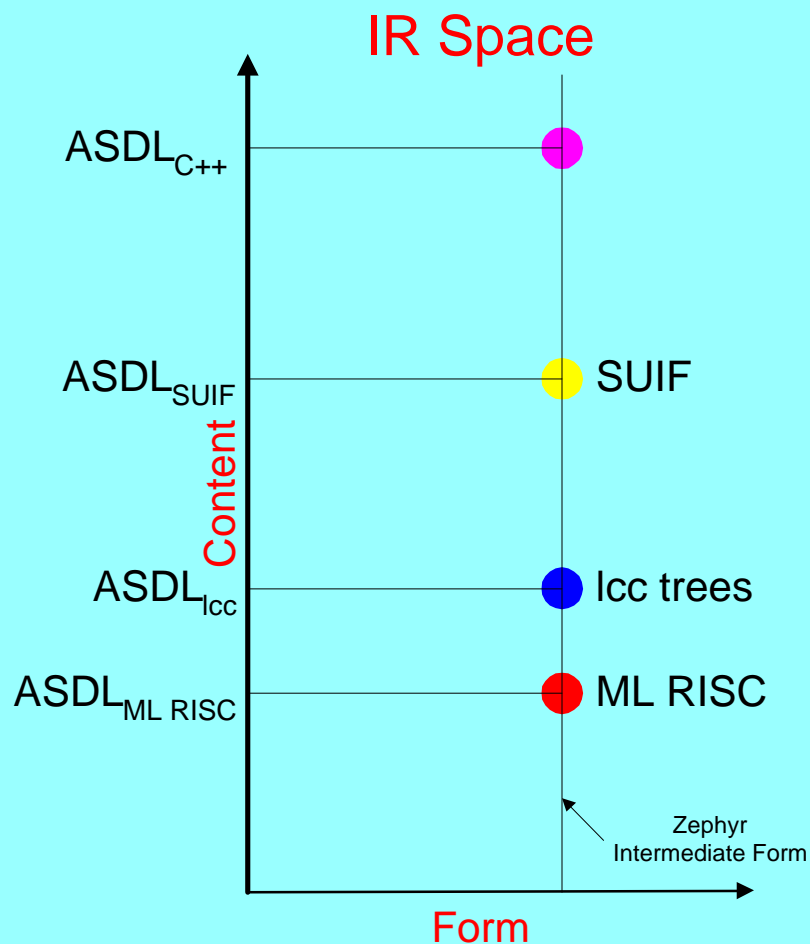
- ◆ **Intermediate representations (IRs)**
 - ▲ Form and content are inseparable
 - ▲ Fixed point in the design space
 - ▲ Cannot reuse tools
 - ▲ IR may not suit needs





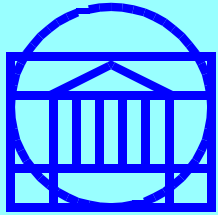
The Zephyr Approach

University of Virginia



◆ Separate form from content

- ▲ Fixed intermediate form: ZIF (trees)
- ▲ Content specified using ASDL and CSDL

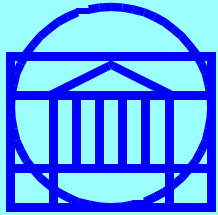


The Zephyr Approach

University of Virginia

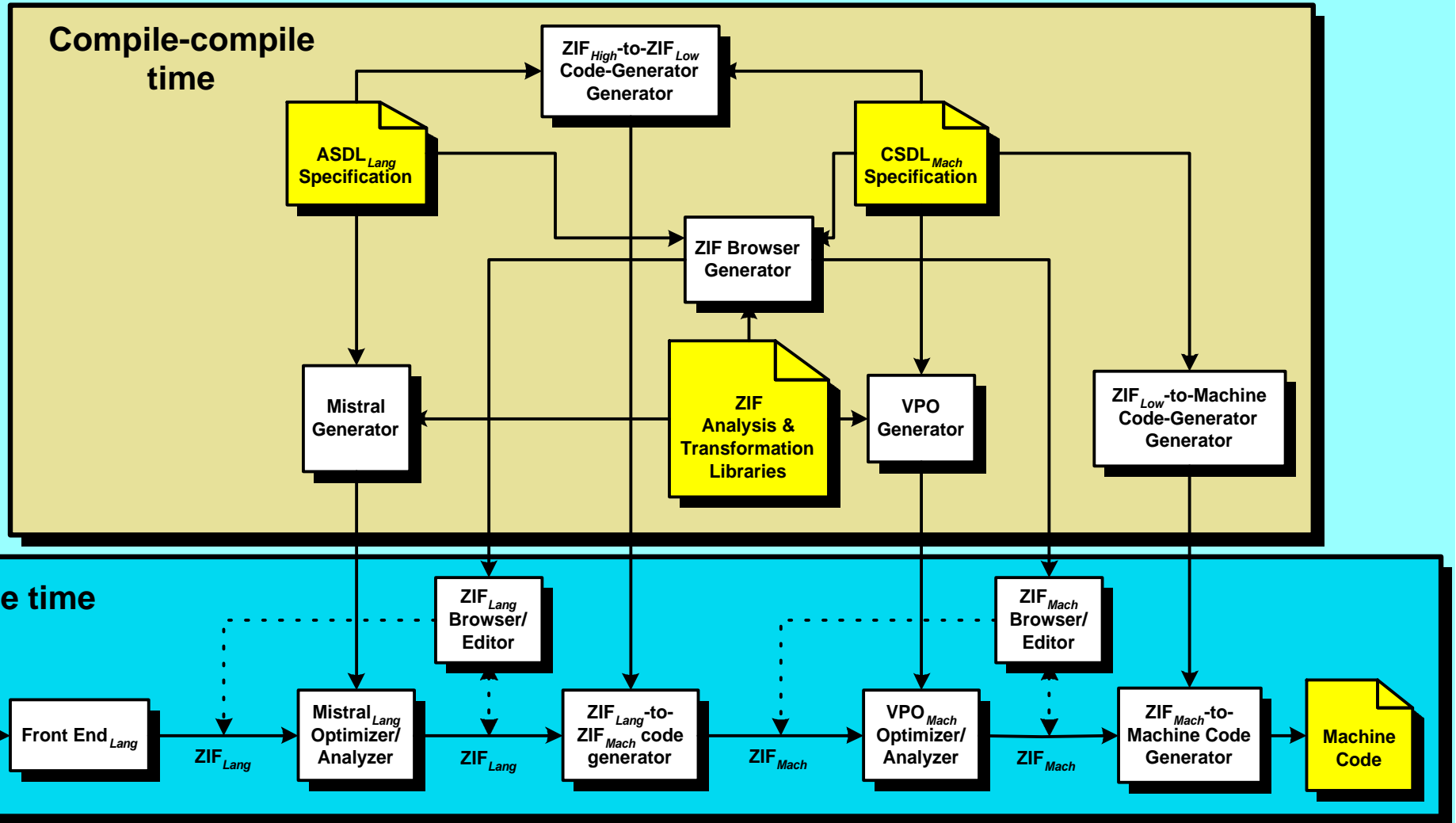
◆ Advantages

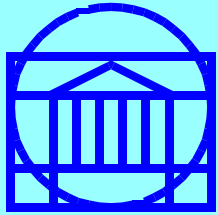
- ▲ Researchers can use IR most appropriate for application
- ▲ Zephyr language processors (picklers, browsers, editors, analyzers, optimizers) are generated from specification automatically
- ▲ Zephyr language processors will be flexible, extensible, and robust
- ▲ Zephyr accommodates existing IRs



The Big Picture

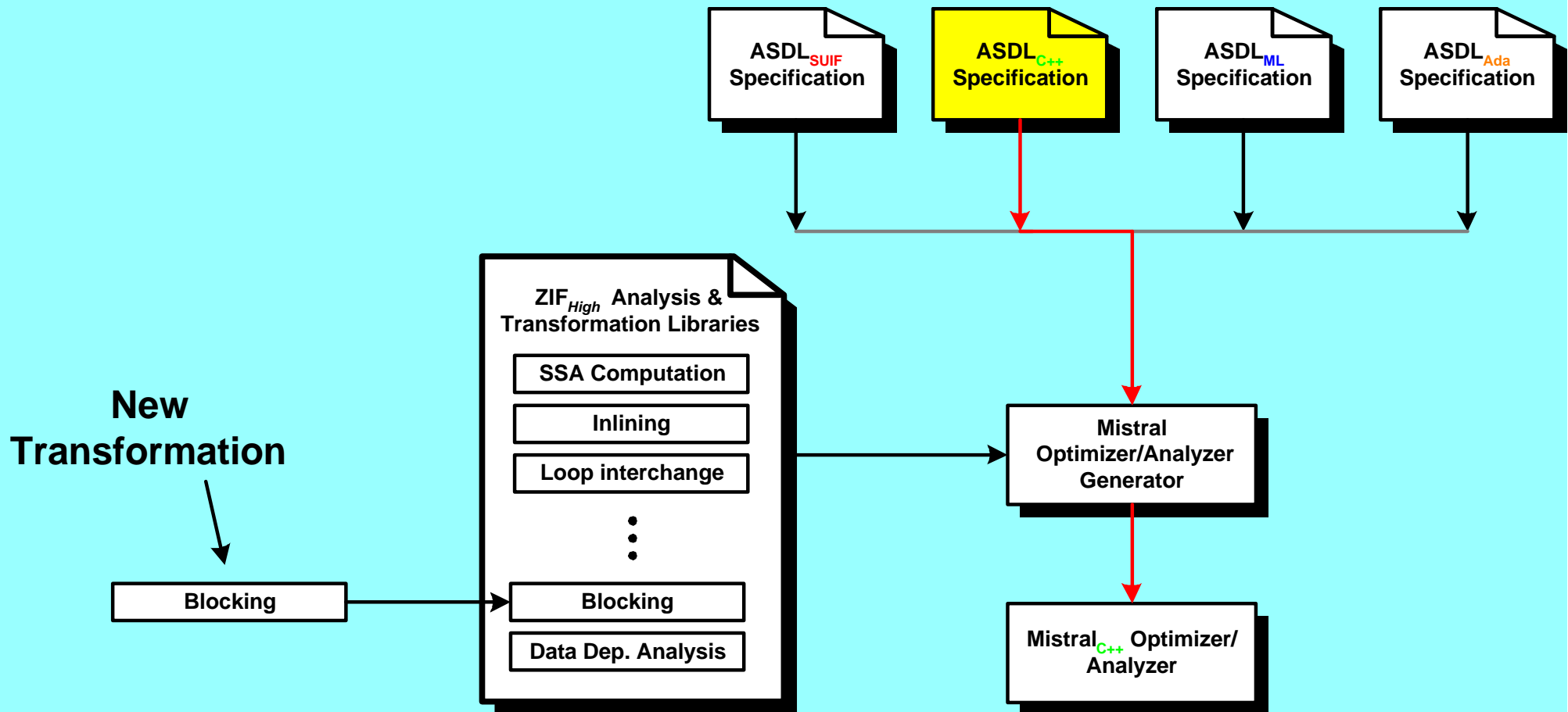
University of Virginia

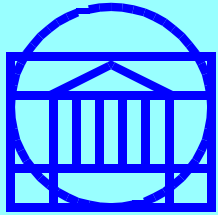




High-level optimizer

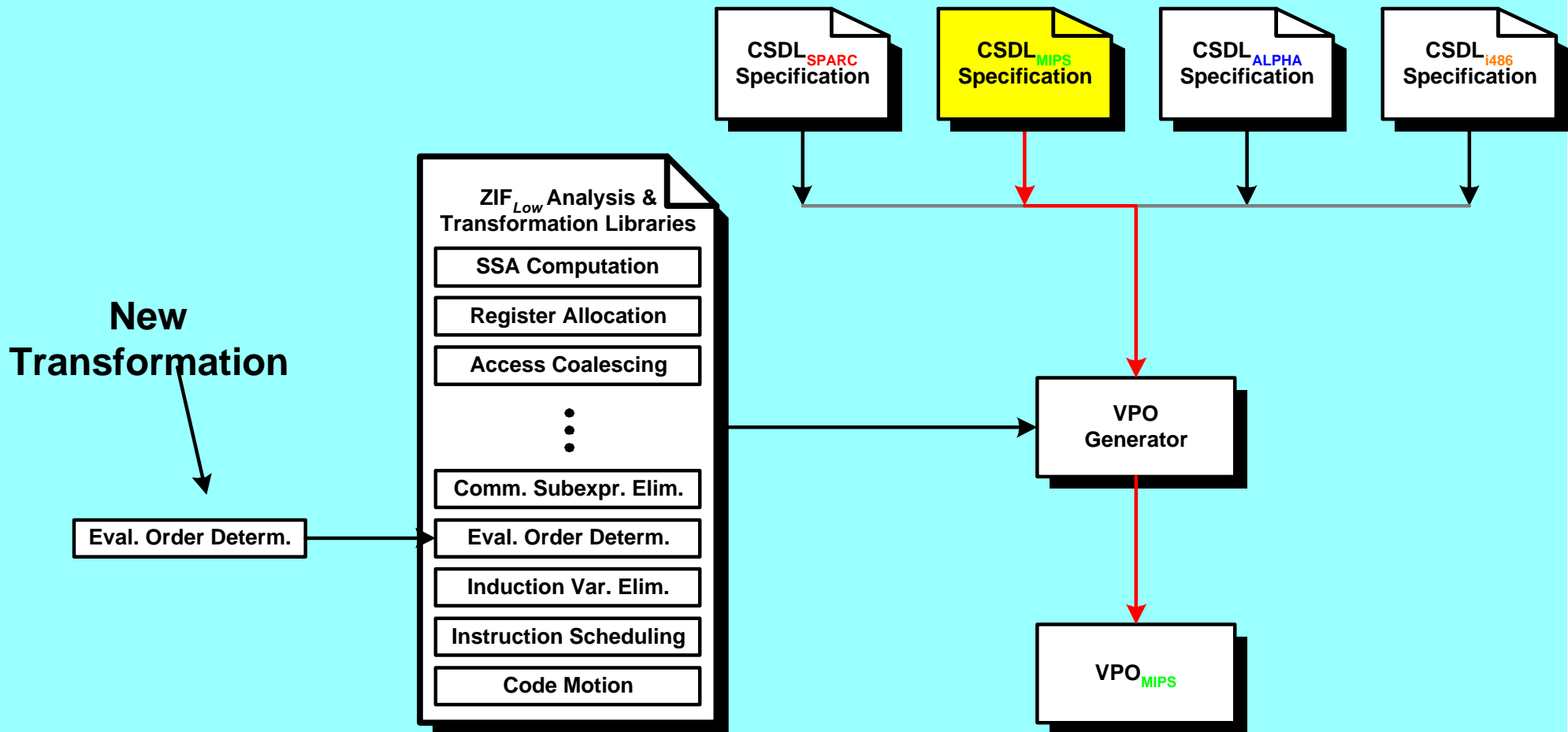
University of Virginia

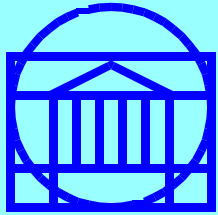




Low-level optimizer

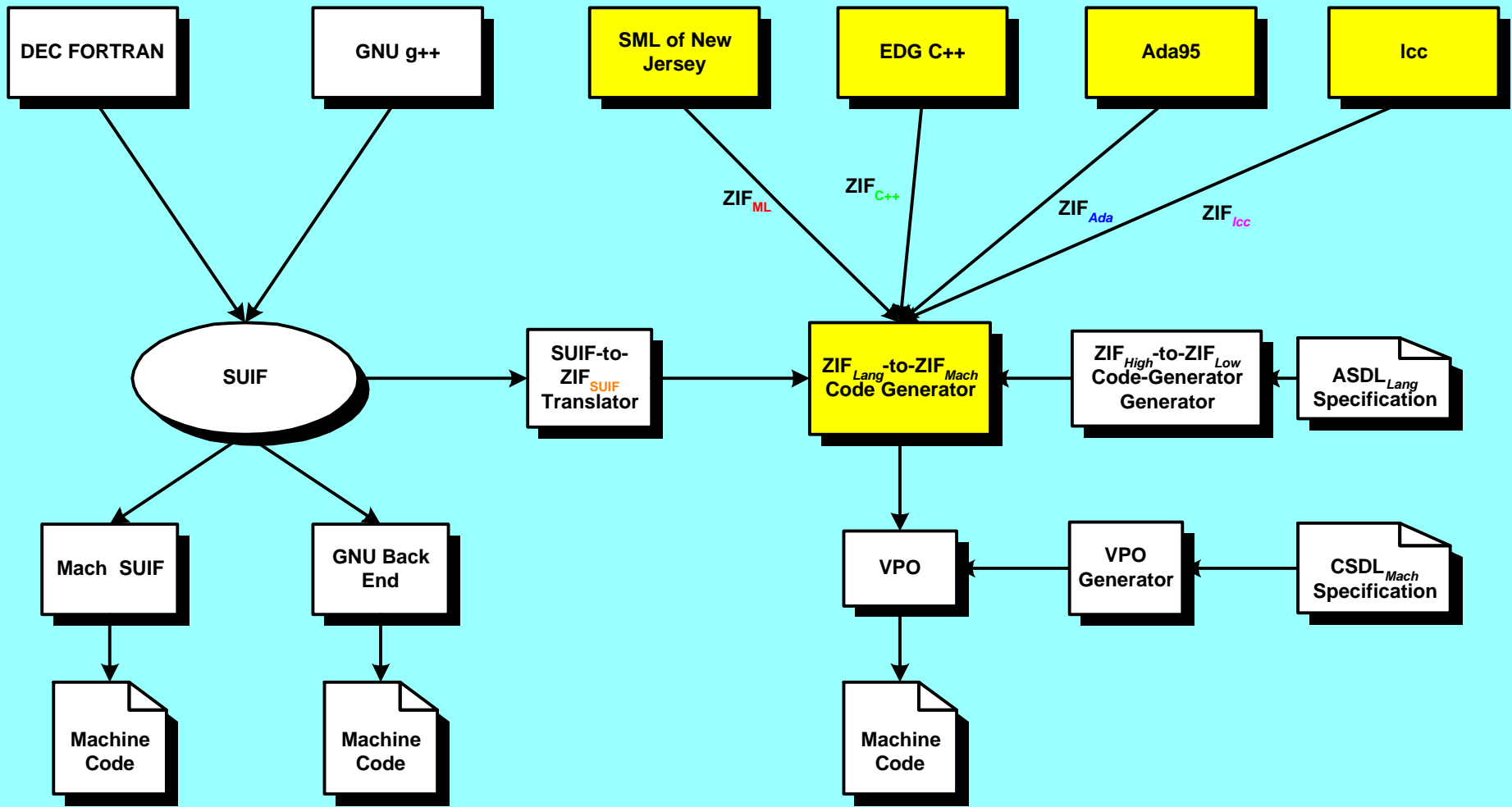
University of Virginia

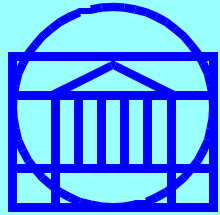




SUIF/Zephyr integration

University of Virginia





Zephyr team

University of Virginia

- ◆ Andrew Appel - Princeton (SML/NJ)
- ◆ Jack Davidson - Virginia (vpo, ANDF)
- ◆ David Hanson - Princeton (lcc)
- ◆ Norman Ramsey - Virginia (NJ Toolkit)
- ◆ Bill Wulf - Virginia (Bliss-11, PQCC, Tartan)

<http://www.cs.virginia.edu/zephyr>