OpenVAF

An Open-Source VerilogA Compiler

Pascal Kuthe

Outline

- Motaivation
- Live demo of VerilogAE
- Current State
- Plans for NGSPICE integration

Motivation

- initial problem: need equations defined in Verilog-A source for parameter extraction
  - before: implement those equations by hand
  - solution: extract all information from Verilog-A source using a compiler
- ADMS not well documented and unmaintained
- ADMS and VAPP are not fully featured compilers
  - no real logical representation of the source file
  - logic for simulator integration put into XML files (ADMS)

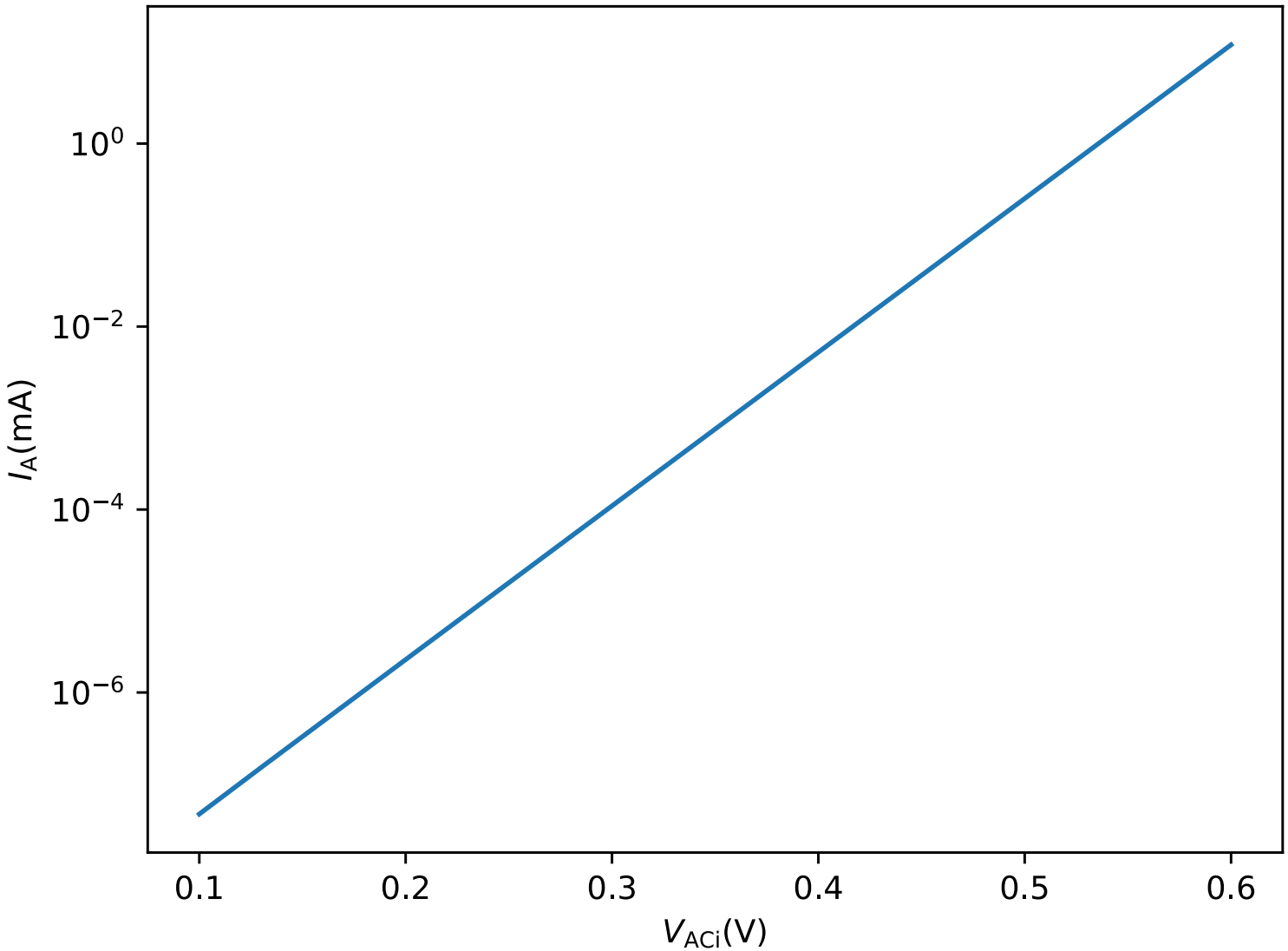
goal:

- develop real compiler to generate machine code
- have a logical representation of a given Verilog-A source
- integration of the generated machine code using an API

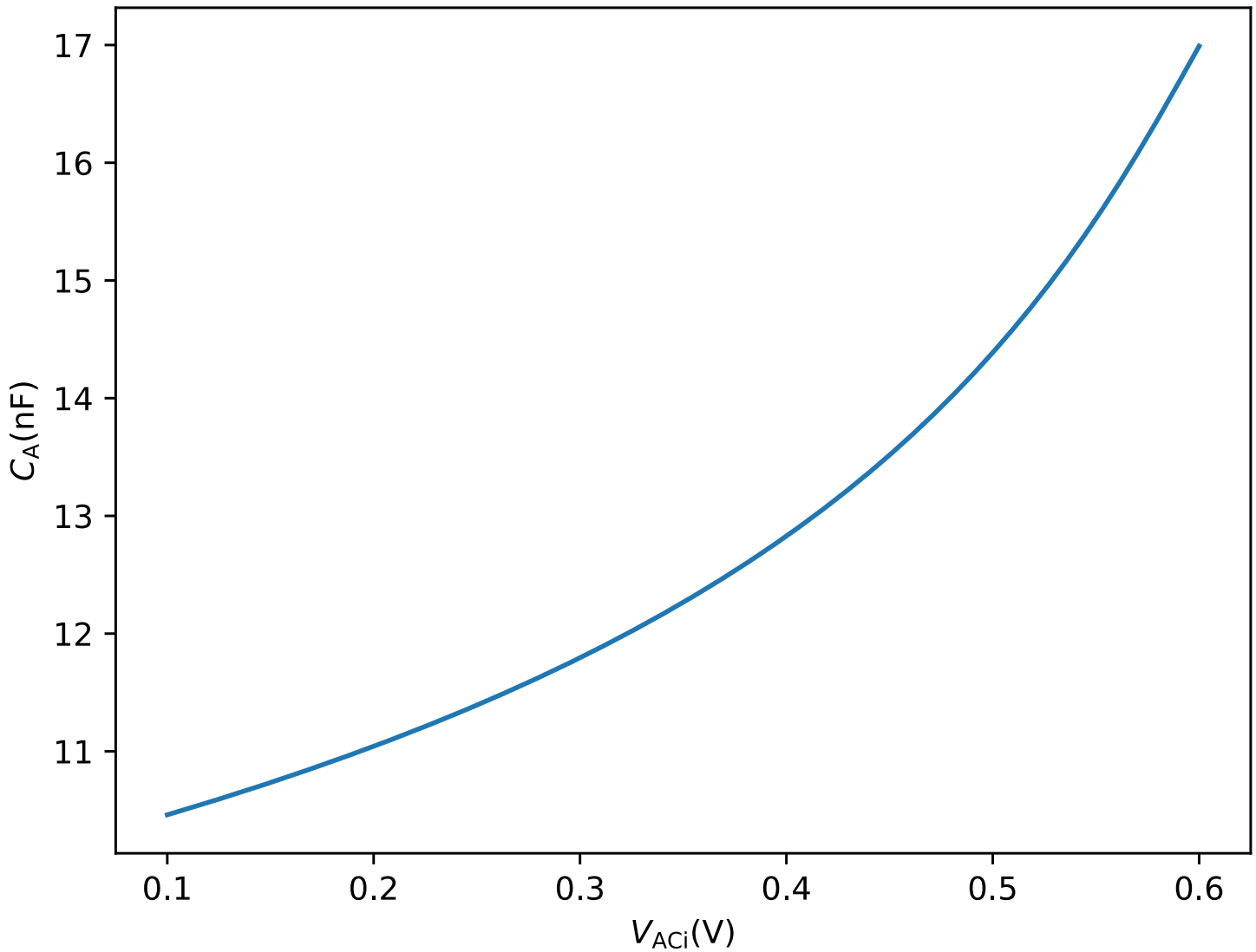
Current State

live demo

Current State



Current State



VerilogAE

- OpenVAF used for Veriog-A compilations
- interface for model equations, parameters and so on with an API
- target language: Python
- already works (well tested for HICUM)

OpenVAF

- 99% of the codebase
- compiler that implements everything needed for Verilog-A code analysis:
  - preprocessor
  - parser/schemantic analysis (-> MIR, logical representation of the code)
  - advanced MIR manipulations
  - analysis of the source file
- first open-source fully featured Verilog-A compiler
- high quality error messages

Application to NGSPICE

- Ngspice does not have a well functioning Verilog-A interface
- OpenVAF can be used as a compiler for Verilog-A
- “only” things needed now
  - interface from ngspice for machine code
  - machine code can be tailored for that interface
- inital goal: static compilation
- key feature to make NGspice attractive for usage in industry and academia

Challanges

- Verilog-A models is just one continuous function
  - understand correspondence of Verilog-A code with Spice
  - -> Use existing analysis to seperate program
- ngspice models split across multiple spice functions
  - many parts of the code repetitive
  - C macros for allocation
  - -> write model interface for ngspice that abstracts over these concepts