



mec

Promethean Toolkit
Centralized Design and Test

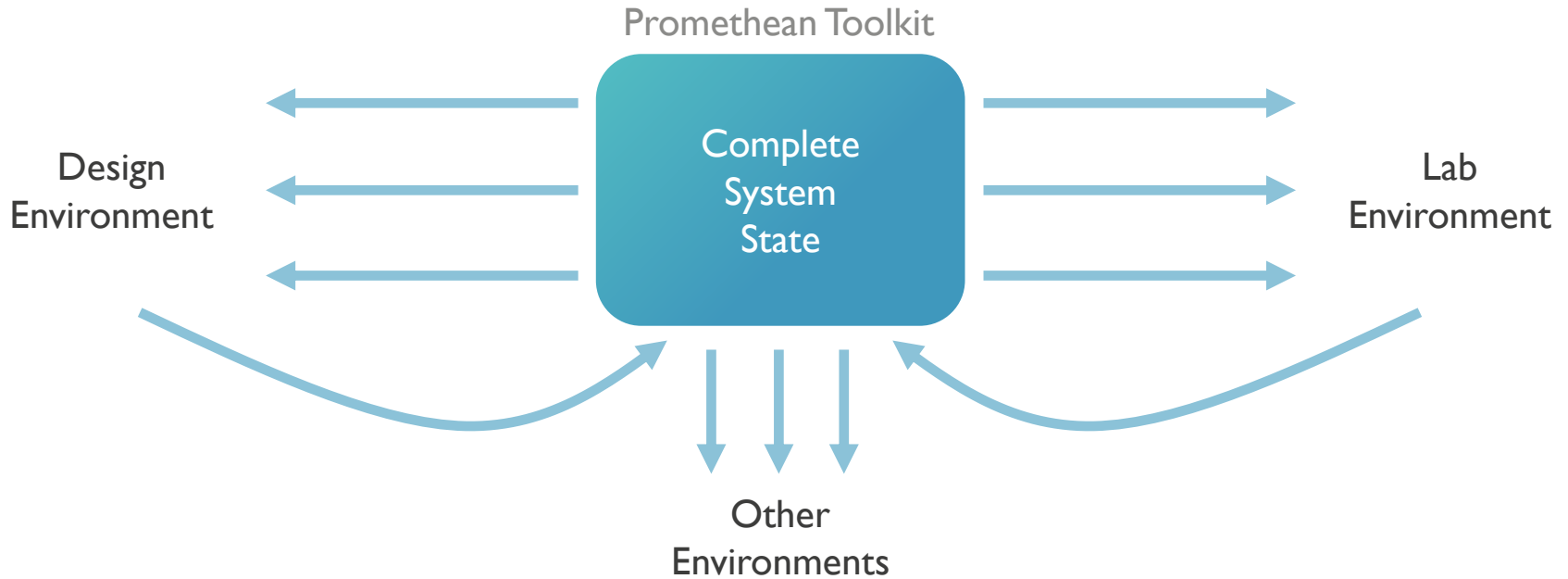
Promethean Toolkit

Centralized Design and Test

- Full-fledged application written in Matlab
 - 3 years of development
 - Already in use by ADC team
- Can be used throughout the design process
 - Design in Cadence (or other CAD tools)
 - Measurement in Lab

Centralized Data Source

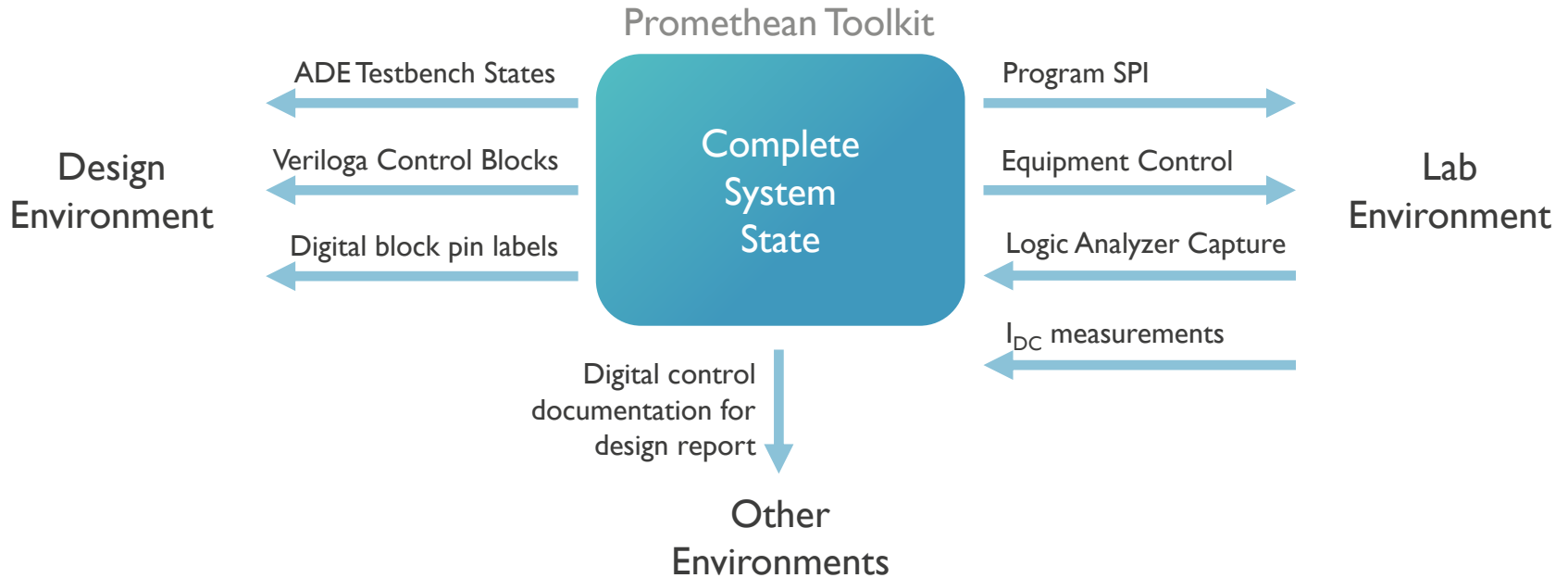
Everything feeds from a single, central source



Centralized Data Source

Everything feeds from a single, central source

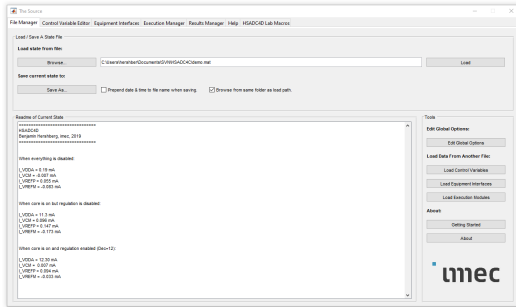
Real Example: ringamp ADC



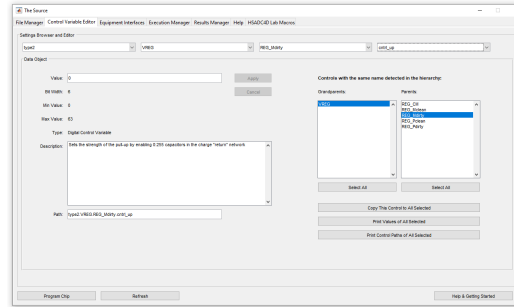
The Common Modules

Top-level System Architecture

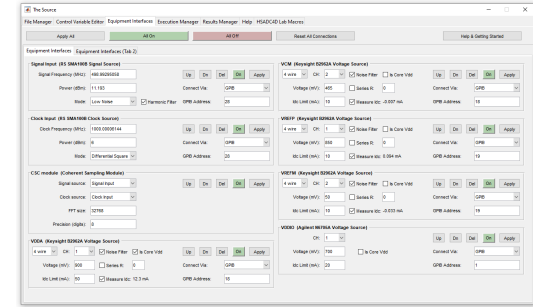
File Manager:



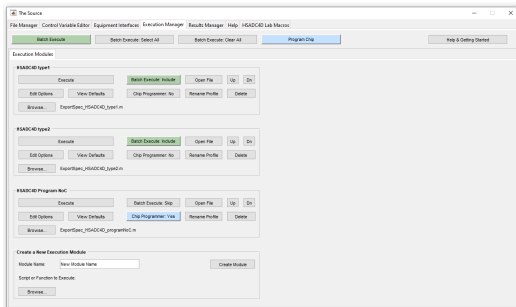
Control Variable Editor:



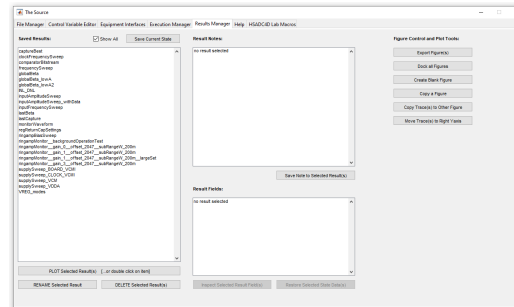
Equipment Interfaces:



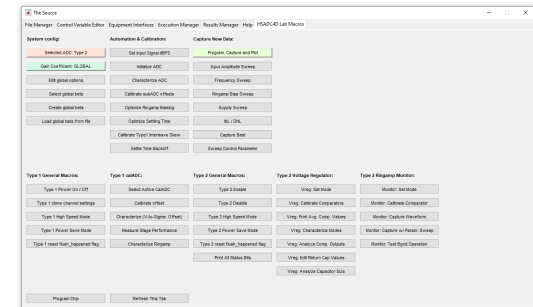
Execution Manager:



Results Manager:



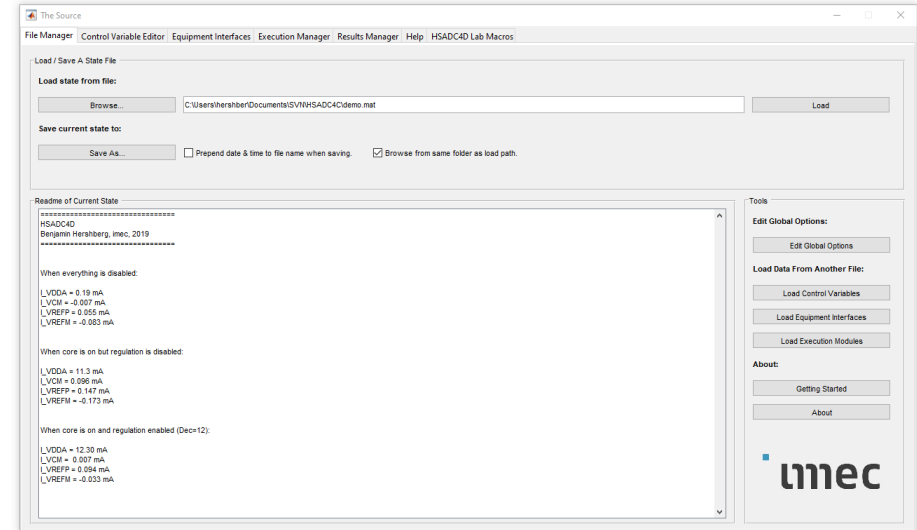
Custom (User) Defined:



File Manager

Tour of the Common Modules

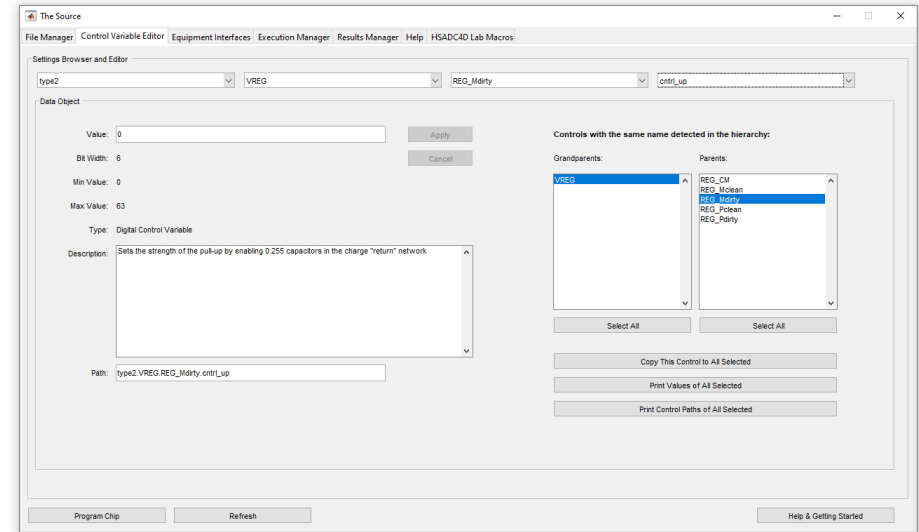
- Main Purpose
 - Load / Save states
- Features
 - “Readme” scratchpad for annotations
 - Load specific pieces of another file’s state
 - Configure the “global options”



Control Variable Editor

Tour of the Common Modules

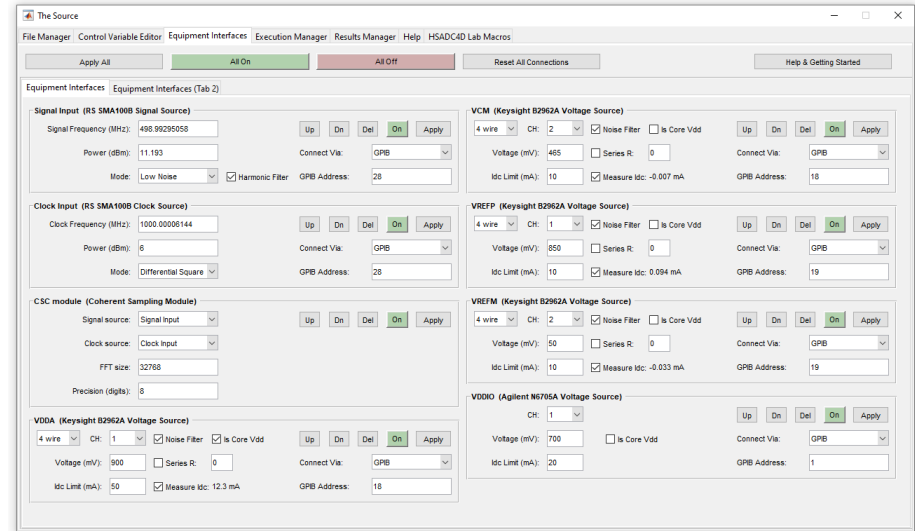
- Main Purpose
 - Browse / Edit the control variable tree
- Features
 - Built-in documentation for variables
 - Detects patterns in the control hierarchy and allows for batch analysis & edit
 - Supports both “hard” digital variables and “soft” cadence variables
 - Define control variables with a simple pre-defined hierarchical struct / cell format.
 - Hierarchy organization does not need to match physical system



Equipment Control

Tour of the Common Modules

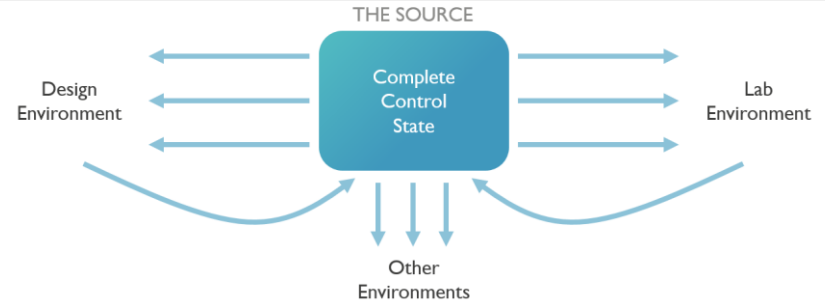
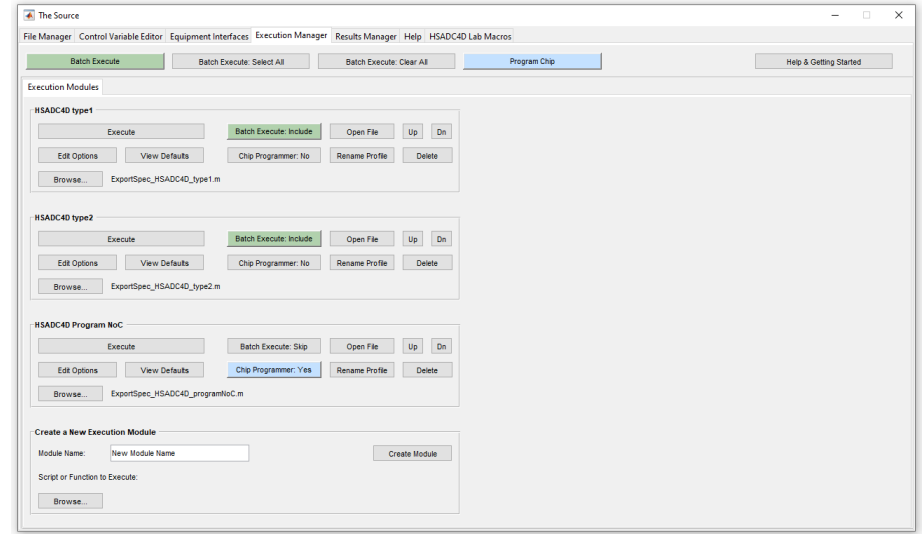
- Main Purpose
 - Control lab equipment
- Features
 - Interactive and fully reconfigurable
 - Any connection protocol (GPIB, LAN, Abstraction Layer, etc...)
 - Simple templates show how to add new instruments and interface with low-level drivers (TMControl)



Export Control

Tour of the Common Modules

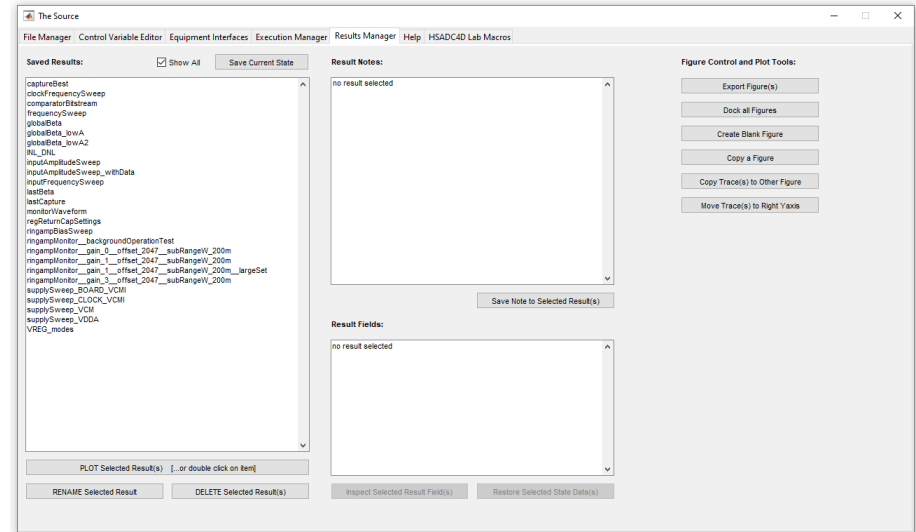
- Main Purpose
 - Export internal state to external formats
 - Execute any custom code in the local environment
 - Program your chip
- Features
 - No restrictions. Run whatever code you want.
 - No learning curve.
 - Allows for user-defined parameters to be defined
 - Built-in library with exporters for common formats



Results Manager

Tour of the Common Modules

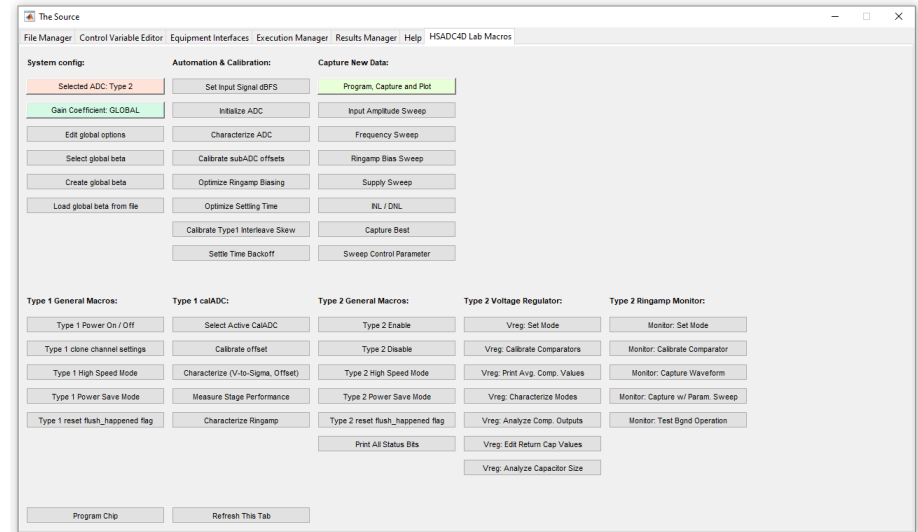
- Main Purpose
 - Plot, analyze, organize, and annotate measurement results
- Features
 - “Auto-magically” knows how to plot your result.
 - Interactive plotting and analysis.
 - Helpful tools for creating figures for external reports, publications, etc.



Custom (User) Defined

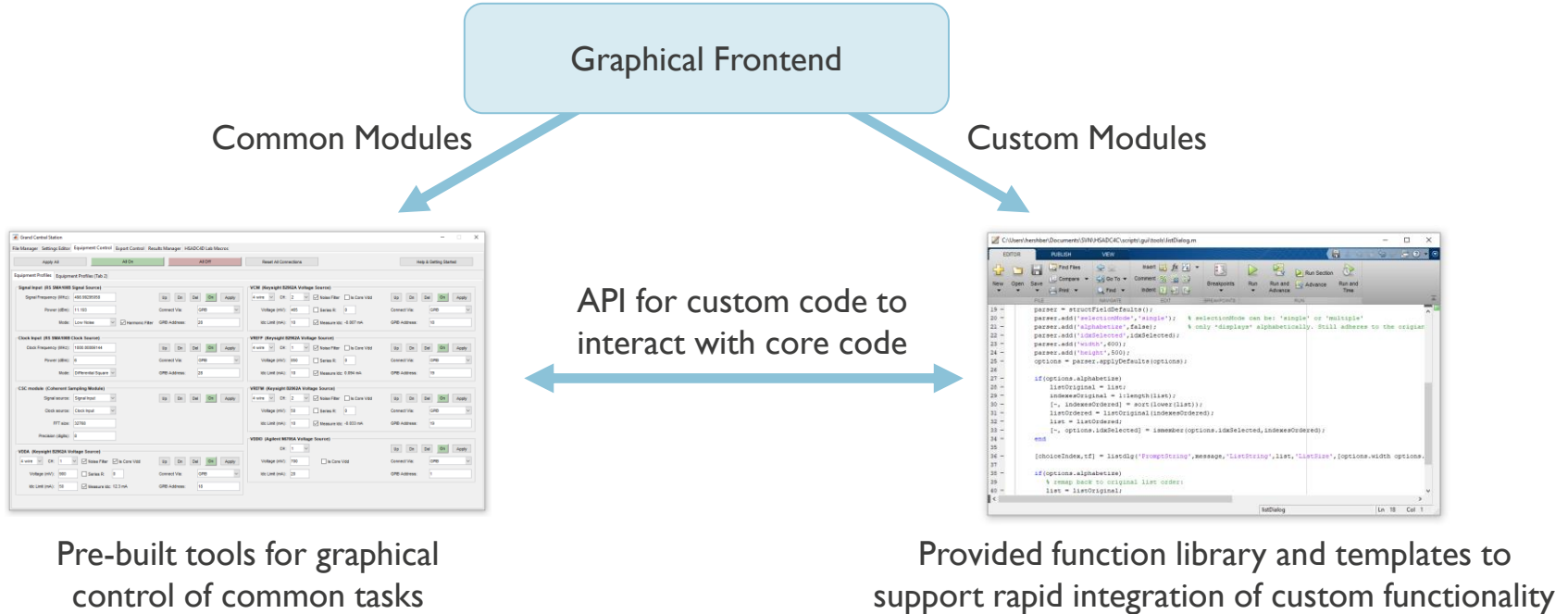
Tour of the Common Modules

- Main Purpose
 - Customize your setup
 - Whatever you need it to be!
- Features
 - Extensive tools, templates, and examples provided for integrating your custom code into the GUI
 - Method of writing your functions that makes them “auto-magically” interactive
 - Extracts the default options out of your functions and lets the user edit them “auto-magically”



Combining Common & Custom Functionality

Top-level System Architecture



Customization

Integrating Project-Specific Code

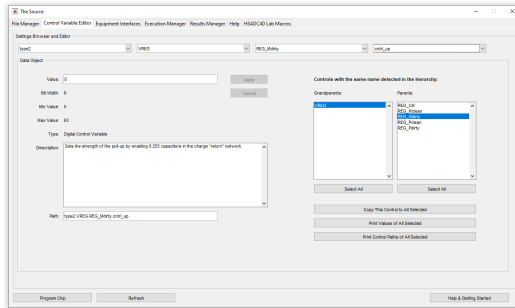
- Templates and examples to guide you the whole way
 - Follows a simple software engineering paradigm, with big benefits
- Code built using the provided templates will “auto-magically” support two modes of execution
 - Interactive mode - for use with the GUI
 - Function mode – for everything else
- More details in a future “hands-on” seminar

Key Benefits

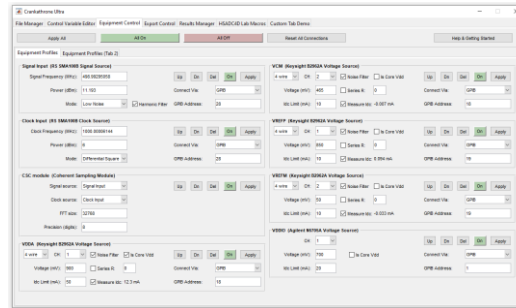
Conclusions

- Productivity
 - It cannot be over-stated how much graphical control provides insight and boosts productivity

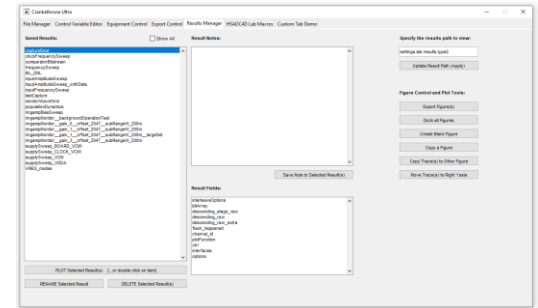
Visually inspect your chip's digital settings and experiment with different settings on-the-fly...



Click a button to sweep a voltage source and immediately see its effect on performance...



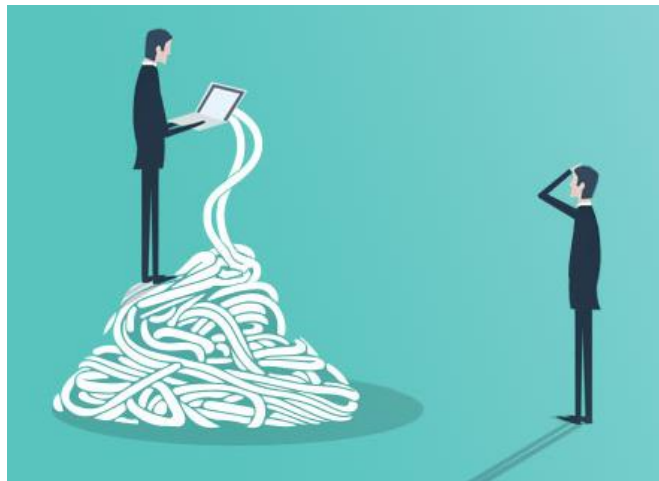
Run interactive analysis and plotting on the results you've acquired...



Key Benefits

Conclusions

- Reproducibility
 - Must Save / Load full representation of test setup
 - Must know what state was used to obtain a result
 - Must know what code was run to obtain the result, and with which user-defined parameters
- Usability
 - Graphical control and visualization of the complete system without needed to dig into code
 - Interactive execution flow of code that allows others to use your code without hacking it
 - Easy for others to understand how/where to make modifications



Will a colleague or external client be able to make sense of your work?

Key Benefits

Conclusions

- Transferability
 - Delivery of saved state(s) alongside physical IP to partners
 - Deliver pre-configured or pre-initialized states for each physical sample
 - With the results from our lab attached as a benchmark
 - Partners can send files of their state for debug / advice
 - You can send them files from our lab for them to try



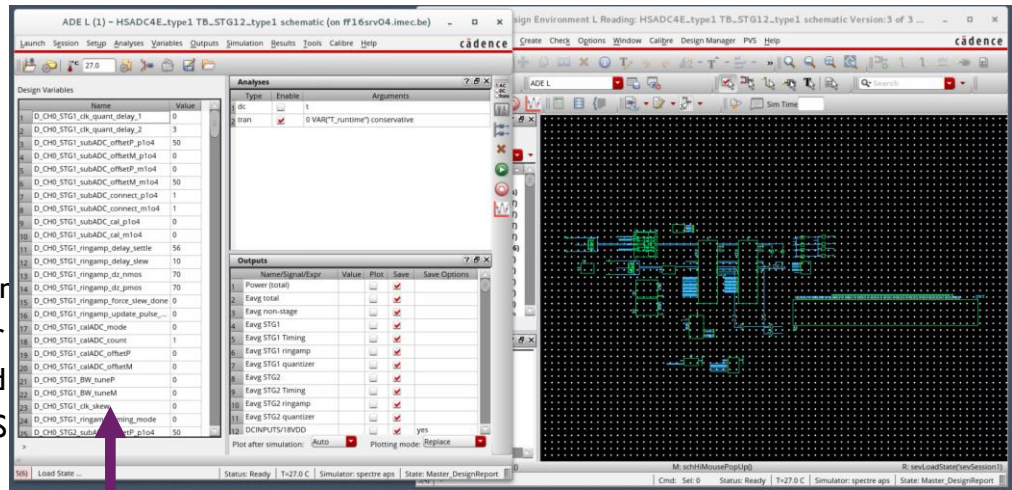
Will a colleague or external client be able to make sense of your work?

Conclusions

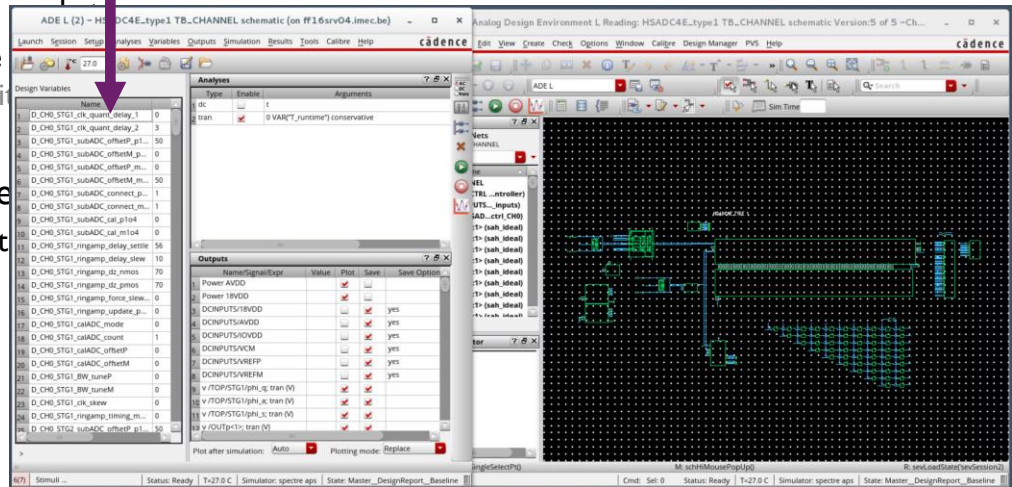
- Your life in Cadence becomes easier
 - All ADE variable names/values in your testben
 - Ever simulated the wrong thing because your
 - Keeps multiple users on a team synchronized
 - Automated generation of digital blocks (e.g. S

- Moving between Cadence and the Lab is trivial
 - When the time comes to test, you're already in the Lab
 - No problem going from lab back to Cadence (Example: using lab settings to conduct parasitic analysis)

- Remote testing is awesome & very convenient
 - Reproducibility naturally leads to remote testing
 - Just load up a state file and go!



How do I maintain a synchronized state?



Key Benefits

Conclusions

- Your life in Cadence becomes easier
 - All ADE variable names/values in your testbenches stay synchronized across the many levels of design hierarchy
 - Ever simulated the wrong thing because your testbench variables weren't setup how you expected?
 - Keeps multiple users on a team synchronized
 - Automated generation of digital blocks (e.g. SPI)
- Moving between Cadence and the Lab is trivial
 - When the time comes to test, you're already ready to go.
 - No problem going from lab back to Cadence either.
(Example: using lab settings to conduct parasitic extraction investigations in simulation)
- Remote testing is awesome & very convenient
 - Reproducibility naturally leads to remote testing capability
 - Just load up a state file and go!

Key Benefits

Conclusions

- **YOU** will love this because it will make your job easier and certainly more fun
- **YOUR COLLEAGUES** will love this because they will be able to use your work without a master's degree in Spaghetti Code Interpreting
- **OUR PARTNERS** will love this because they will be able to test physical IP in a way that is reproducible, graphical, and intuitive

Okay, I'm sold! How do I get started?

- Right now: contact me directly
- Version 1.0 released end of November
 - Video tutorials for all key topics will be made available
 - Templates and Examples included in the codebase
 - A hands-on “Getting Started” seminar at imec will be scheduled
- Give back to the project
 - Looking for new project mods and code contributors
 - Hoping to make public open-source

Questions?



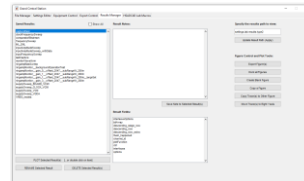
Custom GUI tab allows user to graphically execute custom functions



User graphically specifies the “options” that determine execution flow

`results = yourCustomFunction(options);`

Custom code executes



Results are available in the Results Manager for interactive plotting and analysis