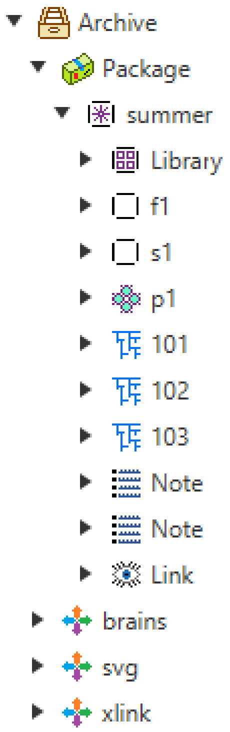
The Thinker™ - math simulator

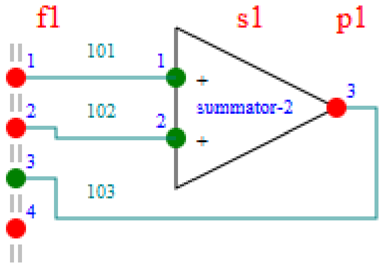
The Thinker is a math simulator instrument to perform visual examination of numerous computational processes running concurrently.



A math model is described as an unlimited hierarchy of math modules. Each function or group of functions inside a module can evaluate the result in its own computing thread. Modules exchange computed values in asynchronous manner. Each math value can be visualized at runtime as either a plain number or floating timeline graph.

Any complex math function can be computed. Arguments of these functions are taken either from enclosed nodes or external pins declared as data consumers. Top level node provides a computed value to a pin declared as a source of data, sending that value to other modules. A particular formula of math function is declared in a form of Java class code, implementing a special interface.

A model may include supplement information, both in textual and graphic form, namely in the SVG format. That gives a choice of visual representation of the module through the fragment. One method is to show the full contents of the module. Another variant is to display just a picture enclosed into a selected joint - the so-called “symbol” of the module (used on the picture to the down left). There is a third variant too, having no graphics at all, but just a text string enlisting pins of the module. In a special case, no information will be shown at all. Last three variants effectively hide the contents of a module when a concise representation is needed.



Definition of a math model uses a hierarchy approach to benefit low efforts to express what and how should be computed. In contrary to regular software, such a definition cannot be used directly during computing time, because there is no hardware to support these data structures directly. Very similar to integrated circuits production, this form of hierarchy description must be flattened.

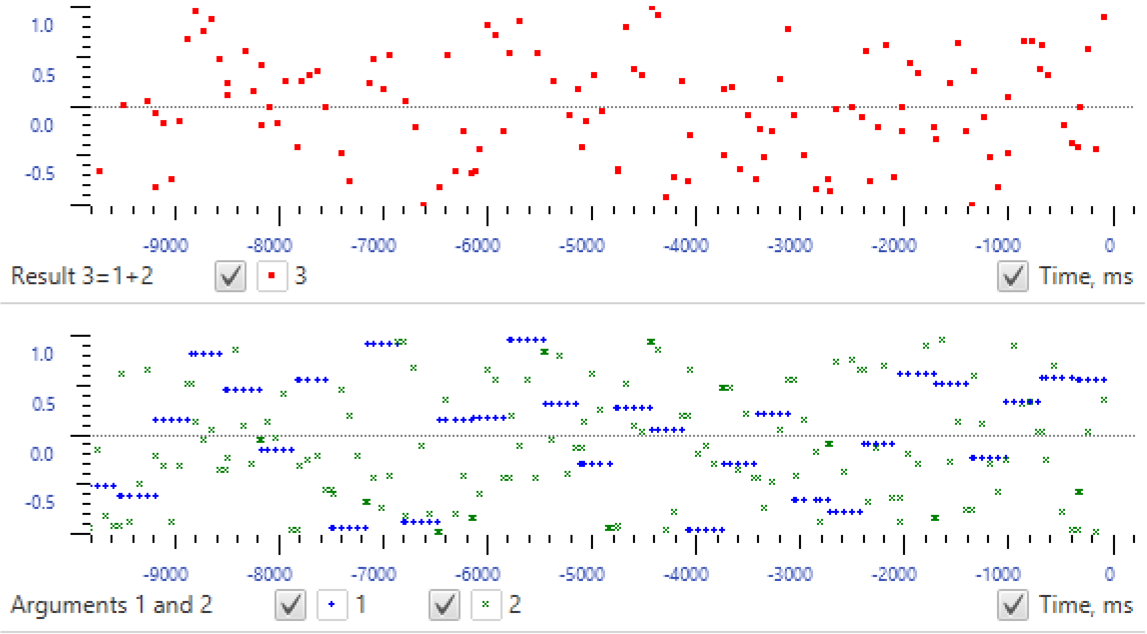
The process of conversion is performed automatically and doesn’t require assistance from the researcher. All needed parameters and modes are already included in the math model. After the model is built, it appears in another browser. It looks very similar to the original hierarchy and even keeps all the same labels, but at the background it is really flat.

The runtime model is composed of a very limited number of element types. They are: processor, transmitter, receiver and math function. Other elements, despite the different picture in the tree, are simply virtual containers, used for the purpose of easy navigation through the model.

Some elements of the computing media can have data flow controls. They are: processors, computing blocks, models and the whole media. Controls allow to start, stop and pause computation for each such element individually. Top level controls override settings of embedded controls.

There are no limits on how many times a single model is built and run simultaneously. Each converted model appears as a separate tree in the browser. Unused models can be deleted to free resources to the application. While a model runs, a researcher has an opportunity to continue other work with the application.

Flattening generator of the model creates every node observable in a graph timeline tool. Any computing node or transmitter/receiver node can be applied to a graph by simple drag and drop operation. Displayed objects can be assigned presentation and conversion features. The graph itself also has separate presentation settings and value range settings for each axis. For the timeline axis, built-in control allows researchers to make a snapshot. An example of a model at runtime is shown below.



This is a dynamic flow graph. The simulator refreshes the graph after every specified interval. Values in millisecond range appear normal even for large screens having 4K resolution. Setting a small interval value creates a smooth motion picture. A big interval value may create flickerings but saves CPU/GPU resources and helps in total performance.

This simulator is a convenient and easy tool to research, develop and debug complex processes described as math functions, where it appears very difficult to obtain an exact analytical formula for every part of a project. The Thinker allows researchers to define a compact math model for the entire project and simulate exact behavior instead, having a possibility to probe every model point.

Summary Of Benefits

- A compact definition for complex math models.
- Arbitrary basic functions to compute, defined by the user.
- Customized mode of parallel computing.
- Convenient tool set to maintain the model both at design time and run time.
- Backup and restore of database contents.
- Presentation and documentation features.

Roadmap to Future Enhancements

- Support for the remote database type.
- Virtual merge of multiple databases in a single data model.
- Visual editor for model elements.
- Streaming of computed data to the file.

Contacts

Scand Ltd.
Web: https://scand.com/
E-mail: info@scand.com
US: +1 (773) 831-4876
Poland: +48 (22) 219-98-19
Germany: +49 (212) 8807-9797
Belarus: +375 (44) 774-47-79