Active Texts: The Swiss Army Knife of the schematic editor SLED by Dolphin Integration!

It is commonplace that suppliers of EDA simulators allow configuring the testbench from some schematic editor. Meanwhile, a gap in terms of functionalities remains between configuring the testbench from the schematic versus from the simulator control file. To fill this gap, and to avoid restraining simulator usage without losing the focus on the schematics, **Dolphin Integration** provides the "**Active Text**" feature, an elegant approach to defining smart simulation features inside the schematic editor **SLED™**.

SLED allows circuit designers to perform graphic entry and to configure their designs in a shorter time. The flexibility of **SLED** enables the design of true mixed-signal circuits as well as multi-level and multi-physics systems.

In order to facilitate development, **SLED 2.3** notably improves the capability of the schematic editor interaction with the simulator through "**Active Text**": a symbol placed on the schematics suffices to insert text into the control file. The multi-purpose nature of such an "**Active Text**" provides any functionality accessible through the netlist. The four main assets of "**Active Text"** are:

- **Flexibility**: eases the development of multiple testbenches by defining relevant directives at the top of the netlist or inside sub-circuits.
- **Multi-language**: allows advanced users to add "**Active Texts**" in any language and to activate these texts only when the schematic is netlisted in a specific language.
- **Productivity**: in combination with "Back Annotation" and "Automatic Calculation", speeds common design development by avoiding going back and forth between the simulator and the schematic editor.
- **Adaptability**: enables adding specific directives to the schematicfor third-party tools such as layout editors or synthesis tools.

These “**Active Texts**” are text entry areas, which are placed in schematics so that the user can freely edit its contents. Such texts are integrated into the files when generated by the netlister.

There are two types of "**Active Texts**" symbols: "**Control Active Texts**" and "**Netlist Active Texts**".

With "**Control Active Texts**", designers can add directives and simulation options (see Figure 1) for test benches configuration. The related texts are placed in the simulator control file.

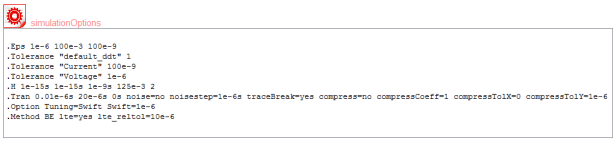

Figure 1: Control Active Text used for adjusting simulation options

It is then possible to define language instructions inserted into the generated netlist with "**Netlist Active Texts**" added in the schematic sheet. "**Netlist Active Texts**" support multiple languages (Spice, Verilog and VHDL) and can be used in an HDL module or a Spice sub-circuit. They enable high flexibility and innovative design methodologies. For instance, in a Verilog module, the designer can add specific lines in different sections for adding Verilog processes or dynamically updating a module (see Figure 2).

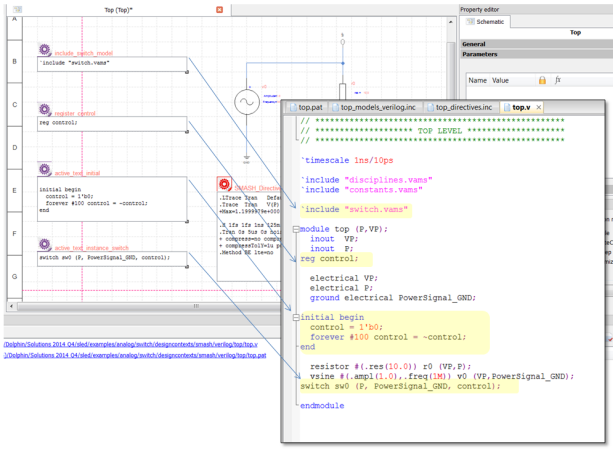

Figure 2: Active Text Netlist used for dynamically updating a module

An ADC testbench helps understanding the flexible and multi language approach enabled by "**Active Texts**". Analog designer sometimes need to perform signal processing measurements for validating their designs. For instance, an ADC integration verification requires Fast Fourier Transform (FFT), post-processing on the digital output for computing the Signal to

Noise Ratio (SNR) and Total Harmonic Distortion (THD). With "**Active Texts**", the designer can create a checker symbol and a schematic in which an Active Text Control with a dedicated .MEASURE directive is placed (see Figure 3).

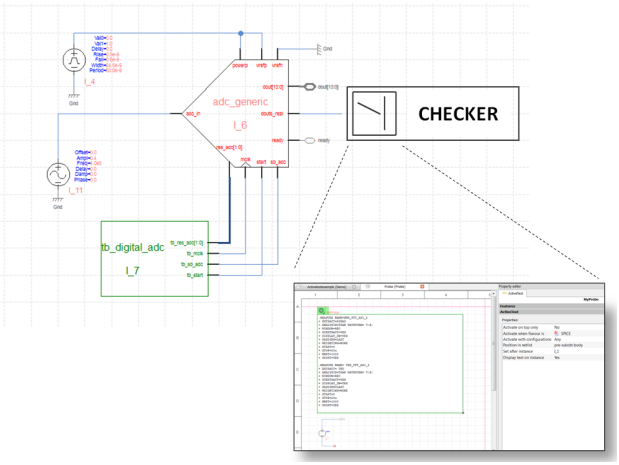

Figure 3: ADC schematic with the Checker Symbol using Active Text and .MEASURE directive.

With correct parameterization of the "**Active Text**" (e.g. Language: SPICE, position: Pre sub-ckt body), the designer develops a "Reusable Hierarchical Checker" dedicated to detect ADC issues. As illustrated in Figure 4, **SLED** netlists a sub circuit called Checker with .MEASURE directives. Thus, with the schematic editor, the analog team can build custom libraries used for in-depth analysis. In brief, the "**Active Text**" feature provides high flexibility for interaction with the simulator and enables developing innovative design methodologies.

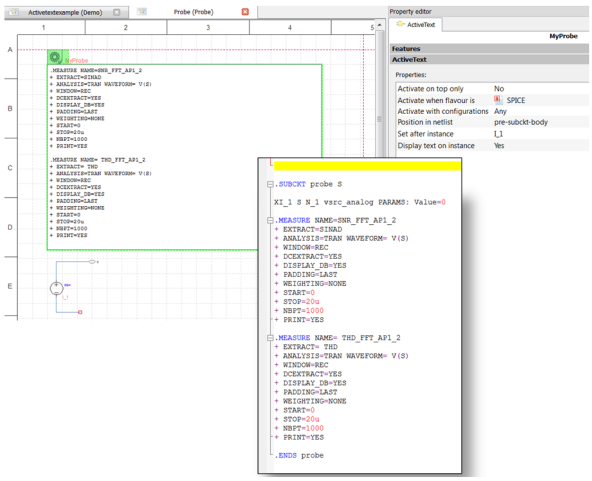

Figure 4: Checker schematic using Active Text and related SPICE netlist

Moreover, to fully benefit from "**Active Texts**", **SLED** provides 2 additionalfeatures:

1) With "Back Annotation" (see Figure 5), used to display the results of **SMASH** operating-points in **SLED**, designers can observe, depending on the back-annotation filter settings, the results of "**Active Text"** changes directly on the schematics.

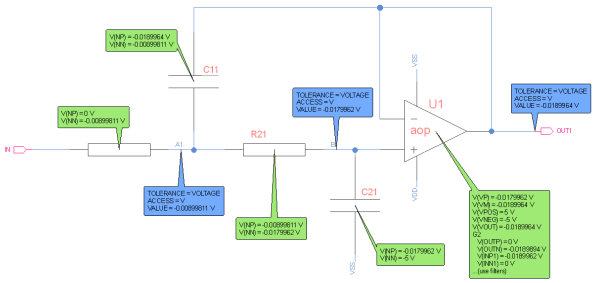

Figure 5: Schematic with back annotation bubbles

2) With "Automatic Calculation" (see Figure 6), used for running simulations and displaying results in **SLED** automatically, designers can observe schematic changes written in the "**Active Texts"** directly in the waveform viewer (see Figure 7) and in the report page generated by"Automatic Calculation".

Simulation results

Circuit: Bandgap

Configuration: Smash/spice

Run #	R1:R		VREF_MIN	VREF_RELDRIFT	VREF_MAX	VREF_ABSDRIFT
Run 0	10k	→	0.8782	1.1802	1.0366	0.1583
Run 1	20k	→	0.6692	1.3799	0.9235	0.2542
Run 2	50k	→	0.5264	1.6060	0.8454	0.3190
Run 3	100k	→	0.4656	1.7439	0.8120	0.3463

Figure 6: Automatic Calculation report in an html page

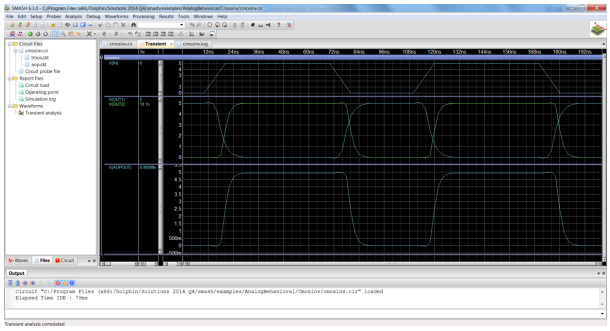

Figure 7: SLED and SMASH waveform viewer SWAVE

Simulation results and analysis are thus displayed by the schematic editor. This tool is very powerful and simple, leading to accelerating development and improving the quality of the design.

SLED 2.3 notably improves the capability to add simulation directives directly in the schematics. It contributes to reinforcing the link between **SLED** and **SMASH**. Designers can integrate simulator control directives into top-level schematics, paving the way to the setup of the test bench and the configuration of all simulations directly from the **SLED** schematic editor. The bundling of the schematic editorand a mixed-signal simulator like **SLASH** provides the perfect "**Front-End Solution**" for designing logic and mixed-signal Silicon IP and multi-physics systems.

For testing the best in class schematic editor SLED™, don't hesitate to download it right here: http://www.dolphin.fr/index.php/eda_solutions/eda_downloads

About the Authors:

Emmanuel Fuchs *has been working as Product Manager for the MEDAL product line at Dolphin Integration since 2014. From 2010 to 2014, Emmanuel has sound entrepreneurial experience by managing a mobile software start-up Bio2Imaigng. He began his professional career as Spice Modeling engineer at NXP and Nanoident Biometrics, where he worked for almost 4 years. By working at ST Microelectronics and CEA LETI (2002-2005), Emmanuel earned his PHD in Microelectronics from the University of Paris XI, France.*

Fabrice Peltier *joined Dolphin in 2000 after a master's degree in electrical engineering (with a strong focus on information and signal processings) from Phelma (ex-ENSERG), Polytechnic National Institute of Grenoble, France. He was involved then in charge of the Software developement for the Dolphin Layout related EDA solutions for 6 years. He has been in charge of the Software Architecture for the Dolphin Schematic related EDA solution since 2007*

Frederic Poullet *joined Dolphin Integration in 1991. After several years spent on the development of analog silicon IP (Memories, ADC, DAC, PLL…), he joined the EDA development team in 2004 where he is now Product Manager of the Analog and Mixed product line. He received the Ms Sc. degree in Electronic Engineering from the Polytech Montpellier (ex ISIM) in 1991. Holder of many patents and international publications.*