# SoC Estimator—A Fast Exploration Tool for Different Architectural Choices in SoCs

E. Azarkhish, S. Emery

*The level of complexity of the integrated circuits (IC) is growing exponentially. Modern system-on-chips (SoC) are composed of several subcomponents with a wide range of functionalities. Sensors, amplifiers, processors, and voltage converters are just a few examples of components which can be integrated into a modern SoC. Today, the component-based design approach is widely adopted, in order to prune the design space and deal with the ever-increasing complexity of the ICs. This allows for selecting pre-designed and pre-validated IPs and integrating them to build a system depending on the required functionality and characteristics. To this end, having an early design exploration platform providing a fast estimation of power, area, and cost is highly desirable. In this report, we present SoC-Estimator and briefly introduce its capabilities.*

A typical embedded SoC consists of different components provided by external/internal IP providers. For example, a random-access-memory (RAM) block can come directly from the fab or from an IP provider with a wide range of parameters (e.g., technology options, supply voltage, memory size). Figure 1 shows a simple SoC composed of components such as filters, data converters, power managers, and processors. When designing such system it is desirable to be able to perform comparisons between off-the-shelf IP components in terms of power consumption and silicon area, and to choose one from a unified library of components. It is then useful to perform a fast evaluation of the system-level power consumption, area, and cost. This can help designers with the choice of the IP blocks and guide them through the subsequent steps.

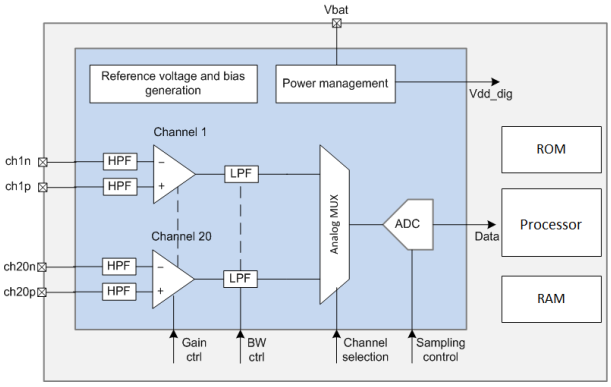


*Figure 1: A simple system-on-chip (SoC) with different components.*

Commercial platforms exist for system-level power estimation and exploration. Docea Power [1] (acquired by Intel in 2015) provides Aceplorer for power modeling and AceThermalModeler for thermal modeling. InCyte [2] by Cadence does a similar job, however, it has been recently discontinued. Also, many other industrial and academic platforms exist for system level explorations. Most of these platforms follow different methodologies and are not easy to learn, integrate, and use. One simple way to obtain early estimates is to use Excel sheets. Excel sheets are easy to work with, but tedious to modify and extend. Formulas, also, are not self-explanatory, and are difficult to remember and modify later:

```
=IF($C$7="Typical";VLOOKUP($C$4;$AA$120:$AT$147;3);
```

We propose SoC-Estimator, a python-based environment capable of parsing self-explanatory models, with enough flexibility to support new components.


*Figure 2: An overview of the SoC-Estimator framework.*

An overview of the SoC-Estimator environment is shown in Figure 2. It receives inputs in CSV representation format and allows for defining typed parameters. It supports a wide range of units (e.g., Watts, Amperes, Volts), and allows for combining them (e.g. $uA/MHz$ , $uW/MHz/mm^2$ ). It automatically performs unit conversion and checks for errors and inconsistencies. Complex expressions and formulas are supported, as well:

| | | | |
|---|---|---|---|
| param | I_total | uA | (I_ampbase + I_amp*Nch*fH + I_adcbase + I_adc*SampleRate + I_lvds + I_imm + I_temp + (Naux+1)*I_aux ) |
| output | power | mW | (I_total * VDD) |

Apart from the expression of nested and dependent formulas, SoC-Estimator also allows for defining characterization tables:

| entry | word | io | mux | area | readc | writec | leakage |
|---|---|---|---|---|---|---|---|
| entry | 128 | 16 | 4 | 7357.838 | 5.656 | 5.121 | 0.093 |
| entry | 128 | 32 | 4 | 11870.65 | 10.081 | 8.928 | 0.154 |
| entry | 128 | 48 | 4 | 16383.47 | 14.507 | 12.734 | 0.214 |
| entry | 128 | 64 | 4 | 20896.29 | 18.932 | 16.541 | 0.275 |
| entry | 128 | 80 | 4 | 25409.1 | 23.357 | 20.347 | 0.336 |
| entry | 128 | 96 | 4 | 29921.92 | 27.783 | 24.154 | 0.396 |

Assertions are easily specified as well, allowing for consistency and range checks:

| | | |
|---|---|---|
| 22 | assert | (-40 <= temperature <= 80) |
| 23 | assert | (1.0 <= VDDA <= 1.32) |
| 24 | assert | (1.0 <= VDDPA <= 1.65) |
| 25 | assert | (0.9 <= VDDD <= 1.1) |

Finally, SoC-Estimator allows for hierarchical instantiation of components. This way, a SoC platform can be built easily from the existing components and model. As an example, a complete bio-sensor interface and SoC (see Figure 1) was modeled and compared against its specifications.

In this report, an early exploration tool for different architectural choices in SoC platforms is presented. Several components have been modeled using this platform and validated versus their datasheets. SoC-Estimator is easy to learn and adopt, and its component library can be easily extended to be used for future explorations.

[1] Intel Docea Power: https://www.intel.com/content/www/us/en/system-modeling-and-simulation/docea/overview.html.

[2] Cadence InCyte: https://www.design-reuse.com/news/19019/power-analysis-pre-rtl-exploration.html.