

MunEDA FAQ's

1 General introduction

MunEDA offers EDA software tools for transistor level circuit design. MunEDA tools are used to optimise circuit performance trade-offs and reduce their sensitivity to parametric process variation and operating conditions.

The software is used in production by analogue designers and full-custom digital designers world-wide. The software is mainly applied in leading-edge circuit design projects with difficult-to-achieve specifications, broad operating range, or particularly high requirements on stability and robustness.

Plenty of customer references and documented silicon results are available, from industrial and automotive analogue design, but also memory and full-custom digital.

2 Frequently Asked Questions

- **Q: We are happy with our analogue design results, no yield issues**

A: The tools are used not only for parametric yield improvement, but most often to speed-up the sizing task for difficult designs. Designers can achieve similar results quicker, especially for non-trivial circuits. There are customer reference cases of re-designs, where designers only thought they had found a good solution, but found a much better one with our tool during a re-design.

- **Q: What makes it different from DFM software?**

A: MunEDA software is used in the front-end flow, the output being a correctly sized transistor schematic. DFM tools work in the back-end layout process. .

- **Q: Flow Integration into Cadence Framework?**

A: The tools can communicate with Cadence DFII Framework, using information from analogue Design Environment and Virtuoso Schematic editor, but can also be run standalone on netlists, and even in batch-mode. Supported simulators are Spectre, Hspice, Eldo.

- **Q: Benefits from introducing the tool into the analogue flow?**

A: Main benefits from introducing the tools into the analogue design flow are

- Designers gain more insight into the sensitivities of their circuits to process variation and operating conditions
- Designers have more time to spend on the real creative design work like topology modifications, while MunEDA tools perform sizing in the background

Effects:

- to reduce the design effort significantly (from weeks/months to days)
- to improve the performance and robustness of the developed circuits
- to improve and optimize the parametric yield based on statistical process data

This implies

- shorter design time to tape-out
- increased chance of “first silicon right” -> reduced risk of a re-design being necessary

- **Q: How do we optimise a design to satisfy the requirements of multiple test benches?**

A: There are currently two approaches:

Alternative 1.

Put all the testbenches on a single schematic sheet, using multiple instances of the design. WiCkeD will spot the duplication of the designs and offer the variables only for one instance, with the others automatically tracking this chosen one. All the .extract statements must have unique names, for example gain1, gain2 etc. WiCkeD can then ensure all the gain* performance measures are in spec. Also .alter statements may not be used.

Alternative 2.

Use a set of scripts available from MunEDA to run the flow across multiple schematics. This is harder to set up but has the benefits of allowing different simulator configurations for each sheet, for example the details of transient analysis performed can be varied on a sheet by sheet basis.

The next release due out in June will support the reading of multiple sheets directly in the GUI giving the best of both of these two approaches.

- **Q: Can the tool automatically recognise memory bit cells?**

A: No.

- **Q: Can we add new rules for the tool to apply?**

A Yes, these are specified in a text file in XML format. They can then be applied using the GUI by selecting the relevant transistors and applying one of a pick list of rule names.

- **Q: Are there any issues with recognising analogue circuit elements (eg current mirrors) if the constituent transistors are in different components of a hierarchical design?**

A: In this case it is necessary to select the transistors by hand and then pick the relevant rule from our library of rule sets. This can be done easily in the WiCkeD GUI where selections across hierarchical levels are possible.

- **Q: If the PDK does not contain statistical information (just corner data) can we still optimise for yield?**

A: Yield optimizations, worst case analysis and mismatch analysis require at least one statistical parameter. With PDK's where only corner data is given and picked up as "slow" or "fast" processes there is a trick that helps. Take the multiple sheet solution from answer 1 alternative 2 and have each sheet reference a different process corner. Then have the tool do a "nominal" optimization using all the corners at once. Again its necessary to make sure the .extract names are unique so for example the gains from each corner are individually readable. It needs to be understood that this is not nearly as good as having statistical data because there may be a worst case combination of variables that affects your design particularly badly that is not covered by any of the provided corners. So where there is a choice, pick processes that provide the statistical information.

- **Q: Regarding optimisation after layout. How does this work:
Can we analyse with the parasitics?
Can we estimate yield with parasitics?**

A: The suggested flow is after layout export a dspf file. Then add the Eldo .DSPF_INCLUDE statement to your netlist to load this data and pass the design back into WiCkeD for analysis. All the WiCkeD analysis and yield estimates will use the layout information.

For RF designs it can reduce the iterations if you attach virtual elements with rough estimates for the parasitics prior to loading the design into WiCkeD. Then after layout remove these and insert the real values. After layout you may wish to run a further optimization but limit WiCkeD to changing only the most critical components. WiCkeD's sensitivity analysis can of course tell you which these are. If you get that info before going into layout, you can make your layout amenable to changes in the dimensions of, say the two most critical transistors.