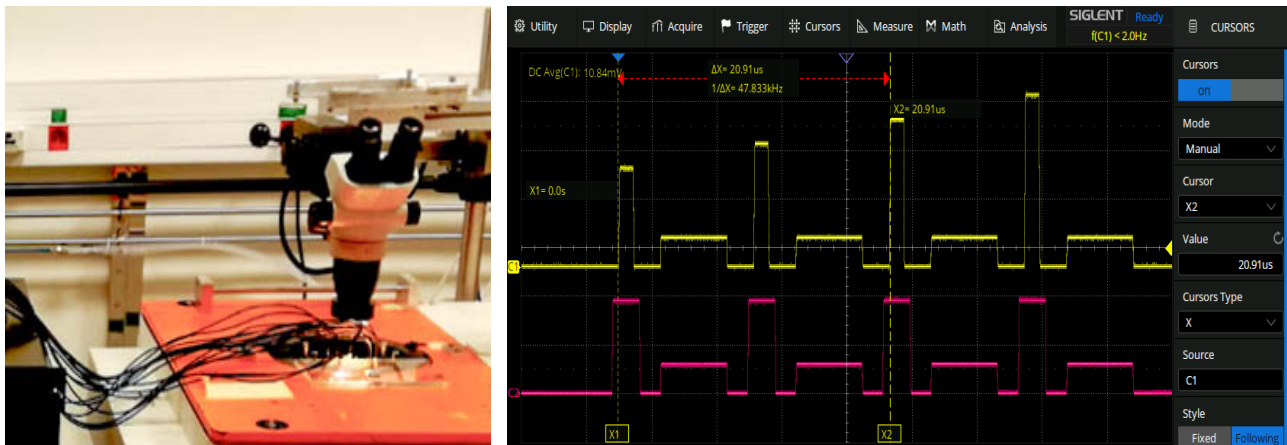




Millions of Cycles in a Flash

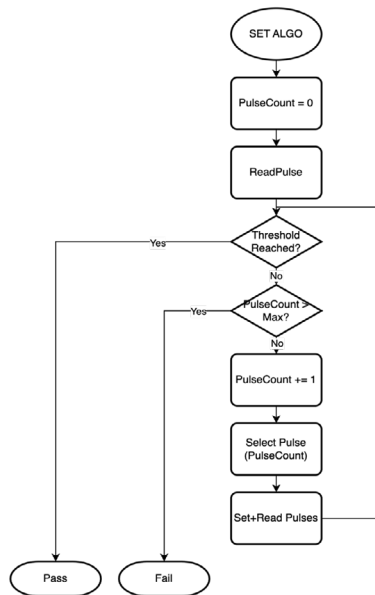
When developing a new non-volatile memory technology, experts agree that in the early phase the best strategy is to build and measure very simple devices like single cells and mini-arrays, without or with a minimal digital interface. I fully agree: this approach not only fastens time to silicon and reduce the risk of design errors, but it also increases the possibility to perform clean measurements, increasing thus visibility. Moreover, it provides an excellent platform to experiment with device management algorithms.



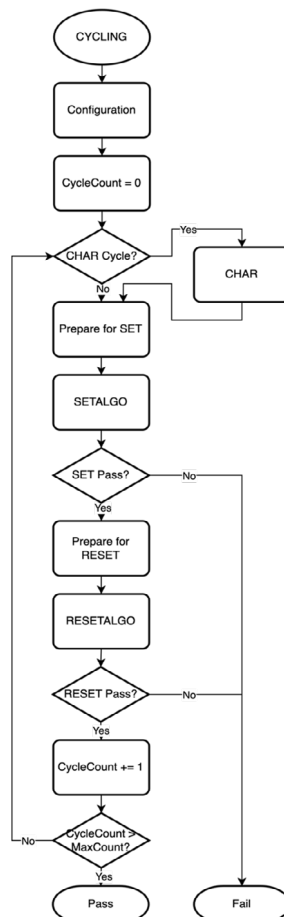
On the other hand, evaluation of such devices – as they might not include charge pumps and sense amplifiers, requires a fully analog environment, setting voltages, applying pulses, and measuring currents. Modern instrumentation can perform this task with excellent accuracy – but they start to suffer when execution time enters the game.

To understand the importance of the execution time, consider that we have a single cell, which, to change the state from 0 to 1 or from 1 to 0¹ requires a specific pulse followed by a measurement to verify if the required status is reached. Eventually, we need to apply a second, maybe different, pulse then check again, then apply a third pulse, and so on.

¹ 0/1 might mean low/high resistance or low/high threshold voltages depending on the technology, so instead of binary values we work in the Ohm and Volt domain



We might need to perform the 0/1 and 1/0 changes many times if we want to understand and confirm the endurance of our technology. “Many” can be a huge number, certain technologies are specified to reach millions or even billions of cycles, I heard even about a 10^{11} endurance target!



If you have a look at the flowcharts, these algorithms look rather simple. Indeed, they are. So, let's make a step ahead, a calculus, and look at the execution times, using an example where:

- We want to execute 1 million cycles.
- Every cycle has two operations to change the state and then back.

- Every operation is executed with 5 pulses on average.
- Every pulse means a change pulse, a readout pulse, a current sensing and a decision.
- We have 1 hour for this test.

We have got only 360 microseconds to execute a single pulse! And during this pulse we need to do a sequence of operations: setting up the hardware, triggering the execution, sensing the current and measuring, and taking a decision.

SMUs and common instrumentation are far slower, they will need days or weeks, not one hour.

And what happens if we want to run more cycles? To reach 10^9 , 10^{10} even 10^{11} cycles?? And if we want to get statistics and run the cycles on 1,000 cells???

There might be some shortcuts like blind cycles, parallel execution – necessary compromises to get results in a reasonable time with some data loss. However, the challenge to do the cycling extremely fast remains.

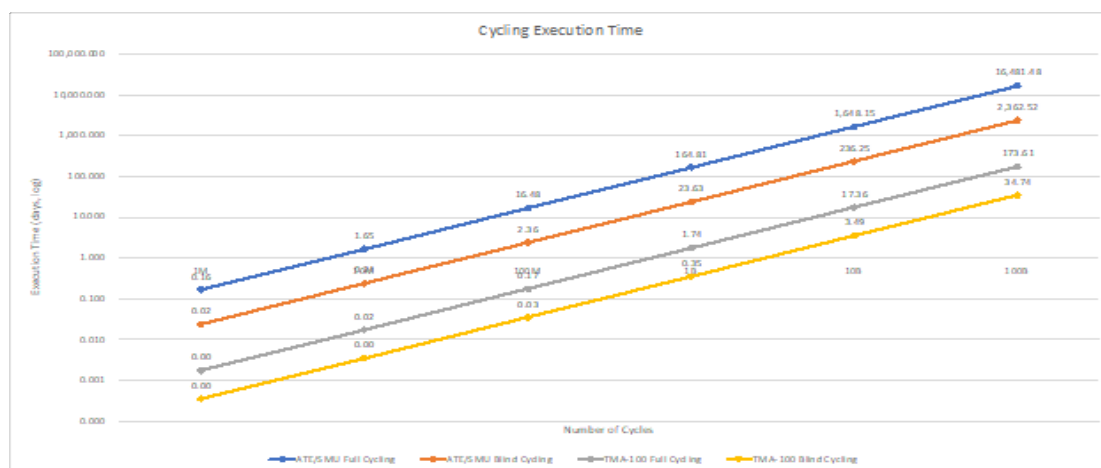
TESTMESH TMA-100 – Speed-Optimized Architecture



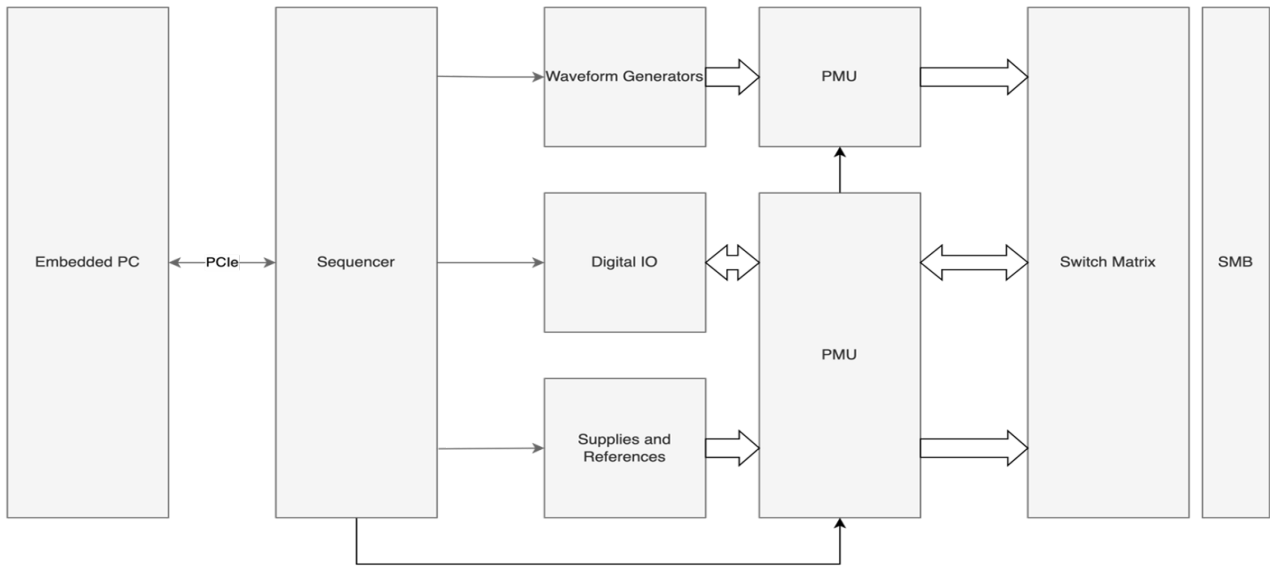
Our TMA-100 has been designed explicitly to do the cycling on test devices extremely fast. It merges the functionality of several instruments into a single one: analog and digital signal generations, signal captures, processing logic – fully synchronized among them.

- The arbitrary and functional waveform generators can be preloaded with multiple pulse sequences. All potentially necessary sequences can be created in the initialization phase. The selection between them, as well as triggering their execution, is in the microsecond range.
- The current sensing circuit has a short (microseconds) setup time. Then, multiple samples can be taken at a 10 nsec rate. The measured value is the result of a programmable, on-the-fly filtering, which optimizes integration time to the required accuracy.
- The patented architecture of the sensing circuitry reduces noise, resulting in lower integration times.

- The processing logic provides also a pass/fail information (1-bit ADC), which helps the execution engine to understand if the required level is reached after the pulse.
- The selection of the current measurement range – essential to obtain the required accuracy for both the high and low current states – uses a fast, dedicated circuitry, allowing selection in microseconds.
- The execution of the algorithm can be programmed into the hardware, without the need for interaction with the embedded PC, for a further saving of time.



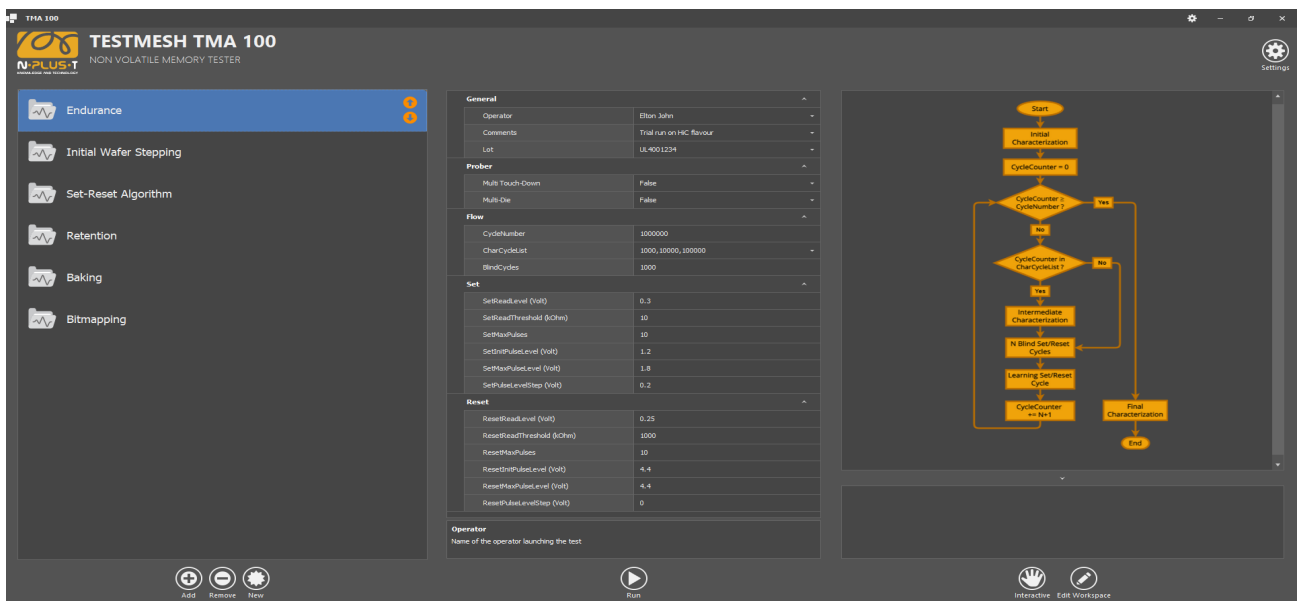
Beyond the analog instruments required for fast cycling and characterization, TMA-100 embeds also:



TESTMESH TMA-100 – Ready-to-Go

TMA-100 comes with productivity tools allowing the users to focus on their characterization tasks rather than developing the toolset.

- Predefined configurable characterization flows.
- Editor for waveforms and algorithms.
- Easy-to-use graphical interface for test setup and execution.
- Topologic display of the measured values of every cell of an array (bitmapping).
- Hundreds of functions processing array data.





Using TESTMESH TMA-100, technologists can focus on their measurements and device evaluation rather than spending time to configure, set up and debug the characterization environment – and they can run the measurements at an unparalleled speed, millions of cycles in a flash!