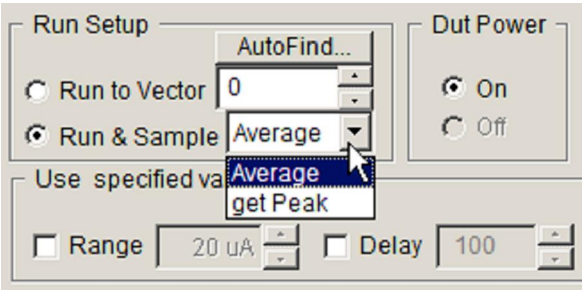
HILEVEL Technology, Inc.

Q'n Apps #E50

Q: What is "Run & Sample"?

Measuring IDD and IDDQ

In the DCPMU window, we find a feature called ñRun & Sampleò when either ñMeasure DUT Supplyò or ñMeasure Iddqò is selected. It appears in the lower portion of the window, like this:



This QôAnApp will illustrate this function by way of example.

IDD Measurements of NAND FLASH

In our quest to measure IDD during the programming phase of a NAND type FLASH (Toshiba part TC58128AFT) we discovered a degree of volatility we seldom see with any CMOS or bipolar device. The measurement mode was **Sample While Running**. Employing the original test files, we observed readings where the relative IDD measurements could vary by more than a magnitude. When changing the vector file so that the program sequencing of vectors provided tight looping on the programming function of the FLASH, the variations were significantly reduced. Nevertheless, we still observed measurements indicative of something less than good methodology.

Attempts to use the PC controller for the purpose of discerning the unexpected results were unsuccessful. In retrospect, we concluded this was so because of accumulated timing uncertainties ranging from the PCôs caching modes to response time aberrations inherent in interfaces between computer and external devices. To combat this deficiency, the HRISC ï a 64-bit Harvard type RISC processor residing in the periphery of the test head ï was utilized. Among the salient features of the HRISC is the quality that every basic instruction takes exactly the same time, with accuracy only dependent on the frequency accuracy of a crystal oscillator. Furthermore, the HRISC is tightly coupled with the DC PMU measurement resources*. Results of measurements along with their interpretations are the principal subject of the remaining text.

Test Setup

The tester employed was a HILEVEL Dragon with 128 pins and with USB connecting to the PC. HiLevel Technology Polska designed a special DUT board for the sake of establishing a MultiSite backdrop that in the lab would closely emulate a production environment. This investigation applied MultiSite mode with two sites, though the combined results are not always shown.

(Note: Subsequently, tests were also conducted using single site mode; the results are provided in an addendum.)

The aforementioned DUT board has sixteen sockets, each of which can house a 128Mb or 1Gb NAND FLASH. Connections between tester and DUT board are through FLEX circuits that combine the utility of high quality transmission cabling and a LoadBoard. The FLEX circuits connect directly to the pin-electronic cards, thus bypassing the standard LoadBoard. While a FLEX circuit looks like a

*While featuring hardware multiply and a ñbarrel shifterò, the HRISC was designed to act as the controlling device of high speed digital I/O devices.

cable, it is indeed a printed circuit board that facilitates a 50q environment; in a sense, the FLEX circuits act as a LoadBoard. All in all, the purpose of this setup is to create a MultiSite environment that meets or exceeds the electrical characteristics of ñDirect Dockingò; we call it ñHILEVEL Firm Dockingò.

Power and ground connections also extend directly from the pin electronic cards. The ground is a solid layer, while a 20 mil trace is used for connecting power. Power sense lines are merged with power lines at the far end of the cable.

Each DUT has two decoupling capacitors: One 0.1µF and one 0.01µF. There is no ñhand-wiringò on the DUT board, implying a 50q setting throughout, excepting at the socket and two connectors; both connectors, the one connecting to the pin electronic card and the other connecting to the DUT board, provide one ground pin for every signal. Indeed, the type of connectors utilized causes negligible discontinuity of the transmission line environment.

Methodology & Measured Data

Importing the acquired data into Microsoft Excel, Figure 1 depicts a striking periodicity of power consumption of the FLASH device when subjected to repeated ***Erase*** operations. While varying between approximately 4mA to slightly above 10mA, the surges in current into 330µs spikes occur roughly in 1 millisecond intervals. Referring to Figure 1, a ñtime tickò corresponds to a program loop of the HRISC, and, for the sake of completeness, the corresponding HRISC microprogram is shown in Figure 2. Each ñtickò, the total time between samples, is 712 HRISC cycles. With each cycle being 60 ns, a ñtick-timeò is consequently 42.720 µs. The majority of that time (41µs) is just waiting for the ADC to become ready*.

* A mode of the ADC is available whereby the HRISC simply waits until the ADC has completed the conversion; in the final implementation this mode will be used. While it reduces test time, it may also have the somewhat disconcerting effect of test times varying for same type devices.

The HRISC operation is invoked when the PC issues an HRISC command. The HRISC responds by acquiring 1000 samples. The results are first stored in the HRISCôs local memory and later transferred, as a burst via the USB connection, to the PC controller. The PC in turn computes min/max values as well as an average of all 1000 samples. The ñrawò data accumulated for a total of ten operations (each of which with 1000 samples) are furnished in Appendix A (because of its length we suggest it be left unprinted).

Again let us emphasize that our investigation employed MultiSite mode, using two sites. In this mode, the sampling is done simultaneously for all sites (one ADC per site) and the data is actually gathered simultaneously for the two ADCs in our investigation. The result is depicted in Figure 3. The striking part of this picture is the skew between spikes for the two ADC readings. Seemingly, the erase operation for one FLASH is done more frequently than for the other FLASH! At least, one device lags the other, and indeed in a cumulative manner.

Measurements of IDD when ***programming*** the FLASH were also made. The picture that emerged as seen in Figure 4 is very different from that of Figure 1. Data appears almost random, though ñmodulated in some fashionò. However, the maximum reading of the current is very similar in the two operating modes. The differences between the two pictures are almost certainly the reflection of the different ways of writing data; yet and again, the max values remain basically the same for the two modes. On the other hand, the min value was measured to be virtually zero for the specified range during **erase** while clearly non-zero during **programming**.

Are these max values real? Would they be higher if unfiltered by decoupling capacitors? Or are they spikes that show an unrealistic representation due to the presence of wire inductance?

Above all, which quantities should be used as a representation of max IDD value? And how

many times do we need to sample? In answering these so imperative questions, we turn to the data in Appendix A. The peak values are repeated periodically and thus presumed to be a reasonable representation of worst case programming current for a specific device. Besides, this value also agrees with typical values reported in the device specification (by Toshiba). What constitutes a sufficient number of samples is a topic for statisticians to ponder; nevertheless, a thousand samples will take less than 50 milliseconds and seemingly gives adequate results. In 10 independent cases of sampling 1000 times, the difference in maximum reading for the 10 cases were no more than 4 ADC counts (50µA). In light of this we tentatively conclude that 1000 samples will be both economical and sufficiently accurate, though it may become subjected to digital filtering in the final implementation.

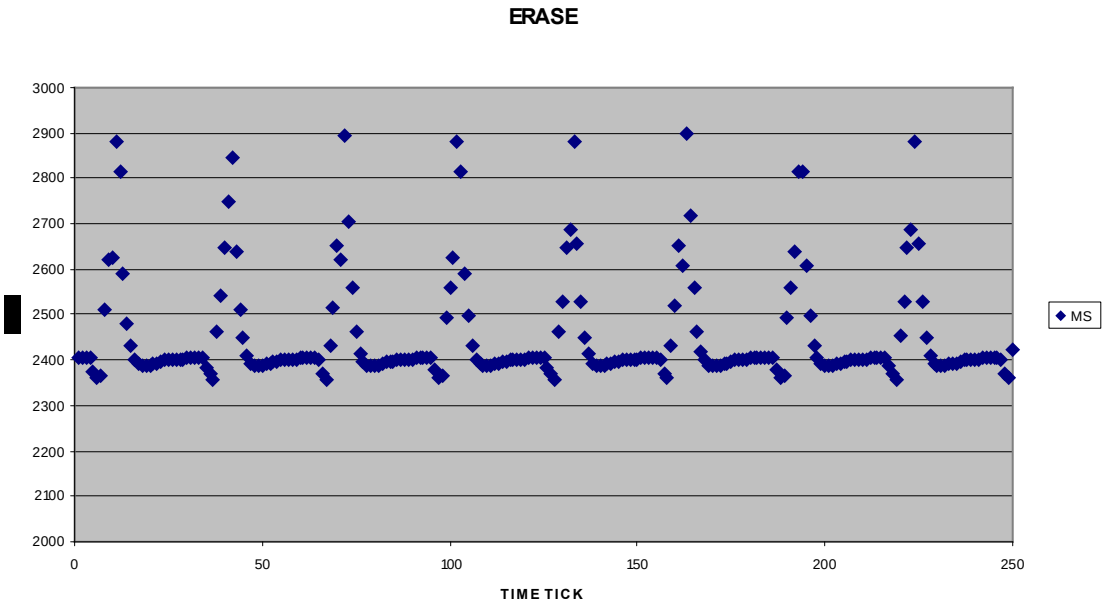


Figure 1 – ADC reading during repeated ERASE operations

```
DO_SAMPLE_POWER_WHILE_RUNNING:
  R(CACHE) = #FFF0FFF
  R(XFR) = 0
  MAR = L(FM_SYS_MAP_AREA)
  R(COUNT) = #3F0; SAMPLING TOO MUCH DOESN'T MATTER NOW
  REPEAT; sampling starts here
    CR = #24
    CALL LOC.PMU32_START_ADC; 4 cycles
    CALL WRITE.IT.SLOW, IOR = 0; 8 cycles
    REPEAT
      CALL DELAY.1MICRO, DECR CR; total loop: 684 cycles
    UNTIL ZERO.CNT
    CALL LOC.PEB_WR_DUTREL; Loc also used for reading ADC (4 cycles)
    CALL TURNAROUND.READ, SET HSI.INHBUS; 7 cycles
    FM = BR.DIRECT & R(CACHE), CALL ICM_INCR.MAR, SET HSI.INHBUS; 2cycles
    R(COUNT) = @-1, SET WR.DIR
  UNTIL ZERO, RESET HSI.INHBUS
  FM = 0, CALL ICM_INCR.MAR
  GOTO FROM.COMPLEX.COMMAND;
```

Figure 2 – HRISC command to execute 1000 samples from ADC

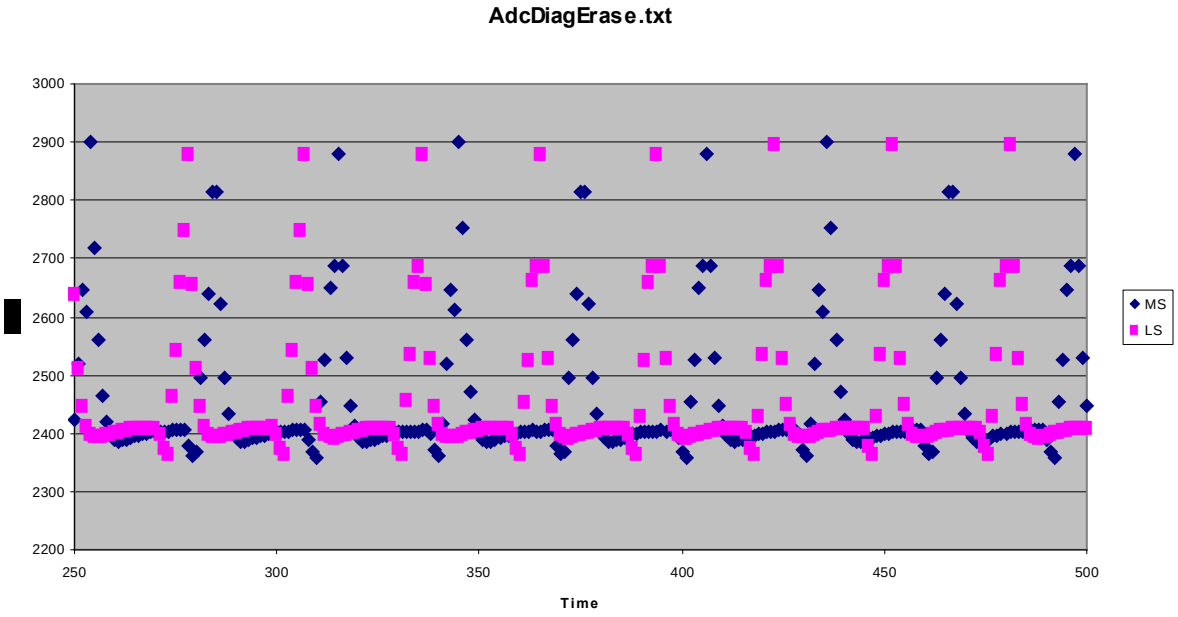


Figure 3 – Simultaneous ADCs readings during ERASE operations

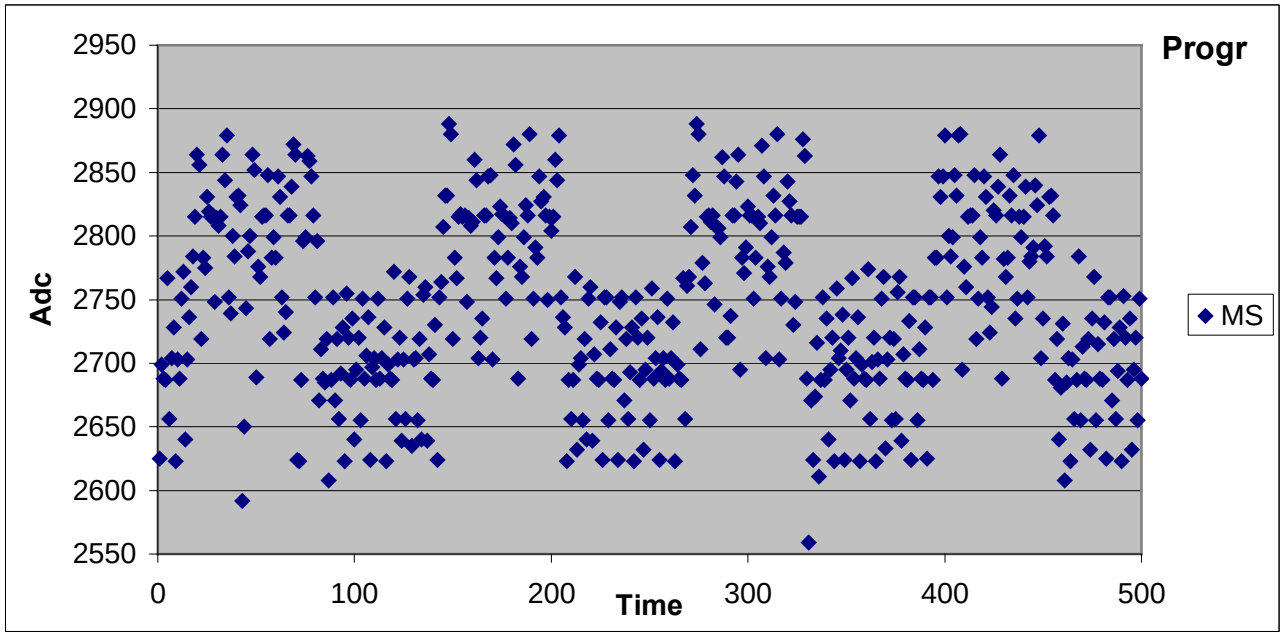


Figure 4 – ADC reading during repeated Program operations

Conclusions and Further Questions

To reach the final conclusions we solicit the opinions of those who by their experience can shed further light on the topic.

It would for instance be nice to know the type of measurement methodology used by Toshiba.

It would also be nice to know what kind of resources other testers have to make the above measurements; in particular, is the statistical approach suggested here necessary. And is it sufficient?

On the specifics, we would like to know if there is any *a' priori* ñdigital wayò of determining when the IDD will be the highest ï in which case we would only need to make the measurement at that moment. Suspecting that the device ñindependently and capriciously chooses its own agendaò, we reluctantly and hesitantly conclude that sampling is the ñonly way to goò.

Upon feedback to all the aforementioned assertions (perhaps of dubious quality) we will embark on the ñnew and alternateò implementation of the ñSampling while Runningò mode of IDD test.

See also:
Q'nApp #E3: Production test
Q'nApp #E5: Power Ramps
Q'nApp #E9: IDDQ
Q'nApp #E11 & 12: The HiLevel TestBox
Q'nApp #E35: Programmable loads, DCPMU

Addendum
Single Site investigation

Surprisingly, using the original single site test we found the results to be almost identical to those of MultiSite. However, when comparing results obtained using the *same* SET file as for MultiSite, we found the measured values to be approximately 30% higher. Then by adding decoupling capacitors to the DUT board, the difference was narrowed to less than 10%. Details are provided in the subsequent text. While we did investigate **erase** mode as well as **program** mode, we focus on the former since both modes exhibit virtually the same maximum readings and the former (**erase** mode) provides a simpler basis for analysis.

The tester was an ETS770 and a general purpose DUT board was used with signals hand-wired to the socket. Power and ground were provided using a flat copper tape in series with round wires (< 3cm). A total of four decoupling capacitors were initially employed (2 of 0.1µF, 2 of 0.01µF), each connected between power and ground of the strips of copper tape[*]. Also, a fixed range (20mA) was applied for taking DCPMU measurements.

At first we ran the test that was initially designed for power measurements (it didnôt have a tight programming loop). When running the test a thousand times (using Repeat Test in AutoTest), we got the following results:

10.200mA (988 cases)
10.010mA (7 cases)
9.810mA (5 cases)

While stable and remarkably similar result to that of MultiSite, the measurements nevertheless exhibited an ñinterestingò anomaly discussed below. In light of later discoveries, the ñremarkable consistencyò between single and MultiSite just seems like ñlucky coincidenceò[**].
Another crucial experiment makes us suspect that we can only hope for adequate results, and not necessarily accurate results. By adding two 0.68µF capacitors, the above 10.200mA reading was reduced to 8.330mA. As one could expect, the peak was dampened by the extra capacitor. What would the optimum capacitor be?

The HRISC microcode for *Sample while Running* in single site mode is now tentatively completed. In the 20mA range, the time to measure takes about 75ms, which includes starting the test, taking a thousand samples, and computing the proper result. Almost half the time (33µs) goes to starting and completing the test; 4ms is what the HRISC uses to compute the max value while the rest (38µs) is dedicated to the actual *Sample while Running.*

---

[*] A detail of potentially great importance in explaining the qualitative differences between measurements in MultiSite and single site.

[**] The above stability would indeed have been remarkable if the measurements were off by 1 and 2 counts. But as it turned out, the aberrations differed by 16 counts and by 32 counts (one count representing about 12µA in the 20mA range); curiously, they were off, relative to the first case, by 16 times 1 and 2 respectively. Perusing through the collected data, the appearance of a fundamental flaw was so strong that we were compelled to run a basic test with resistors to check out the hardware. For DC measurements pertaining to the resistor, the results were as predicted; one is almost left to ask: Is the behavior of the ADC reflecting ñsomething likeò it cannot handle rapid changes? We received the answer from our expert. The 574 ADC (a well-known ñwork-horseò) is not very fast and does not have the sample-and-hold feature that we sorely need in this dynamic application. Fortunately, another ADC, the 1674, does have sample and hold circuitry and features a 10µs max conversion time (versus 35µs for the 574). To what extent HILEVEL may make corresponding changes in its hardware is partially subjected to further testing using the 1674 ADC.

We need to emphasize that the single site test executed so far was not the same as the one applied in MultiSite. When converting the test used in MultiSite to a single site test, we got a big surprise: The results differed by approximately 30%! For erase mode, the results (showing ADC counts) are depicted in Figure 5.

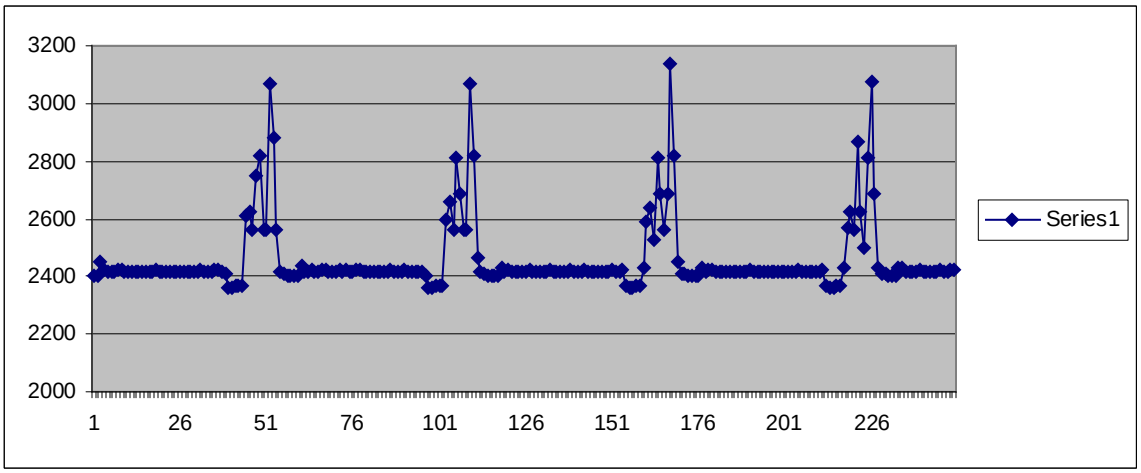


Figure 5 – ADC reading of erase in single site using original decoupling caps

The results show instability akin to transients of voltage readings of a bad high-speed pin driver. By adding two 0.68µF decoupling capacitors, the results changed significantly as shown in Figure 6. And with the addition of these capacitors, the measurements got much closer to those obtained with MultiSite ï the difference was reduced to less than 10%. We suspect more decoupling caps will further ñclean upò the spikes and might indeed render results that match those of MultiSite.

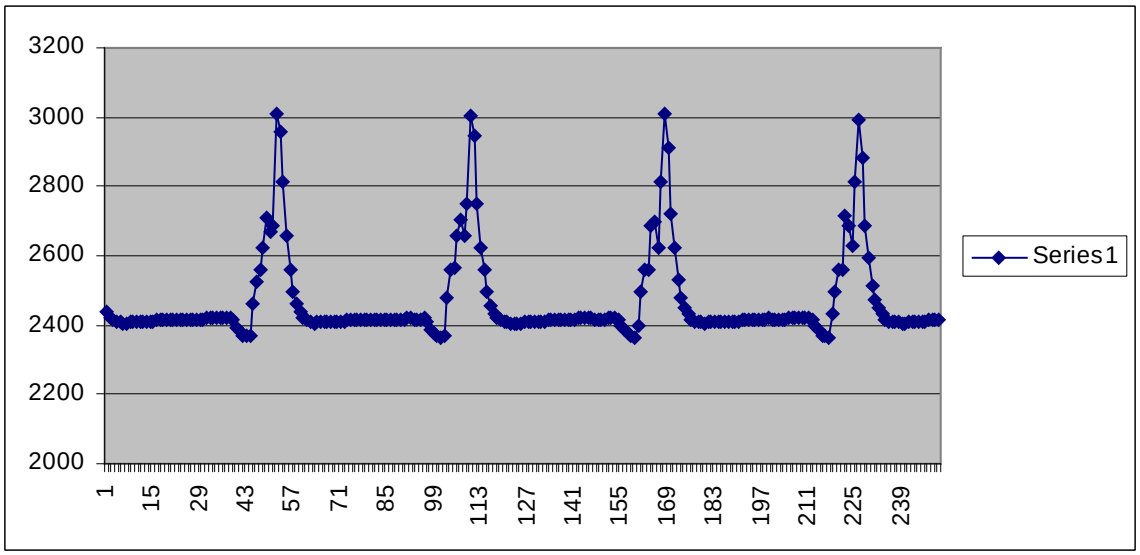


Figure 6 – ADC reading of erase in single site when adding decoupling caps

Final conclusion is yet to come!