Q: Why Static Test Vector Memory?

The trend in test vector memory design is dynamic ó in a double sense. The high-speed SRAM, the traditional component of Vector Memory designs, is being replaced by DRAM. So whatøs the reason for that? Cost and Marketing. The ratio of DRAM cost to SRAM cost is approaching 1 to 10 ó it is increasing fast. Secondly, the vector depth mantra of the industry calls for deeper vector memory. So why not dynamic memory (it also *sounds* better than **"static"** memory). We prefer to call them õhighly volatile memoriesö. Volatile since the content is lost if powered off; highly volatile since a dynamic memory looses its mind even with power on. Either we have to access their content or we have to apply REFRESH. The latter is a vile concept that over the years has caused enormous grief in system designs (in a footnote we refer to a U.S. Navy problem).

Fortunately, provided we can guarantee sequential reads through an entire page, we can actually dispense with refresh. This is often nice for vector memory since in so many cases we simply donøt care for branching, looping and those sorts of things. But sometimes we do. And often we can reduce the required memory size if we employ looping.

More importantly, there are examples when looping is virtually compelling. One such application is match mode. We wait for the device to initialize itself before continuing on. For how many vectors? Canøt always tell, although there is a maximum. To always apply this maximum (and it is troublesome to estimate this maximum for every mode) would be a waste vector space and test time. Indeed, using dynamic memory may not even suffice; often millions of vectors will pass before the chip is stable.

But we cannot escape the fact that Dynamic is more than Static. Or can we? Yes, we can. Typically, simulation involves a lot of repetition, i.e. looping. So should the test engineer ascertain where the loops are and how to apply them? No, we donøt think so. These days, that sort of drudgery is done by machines. So the HiLevel computer does it for you automatically. It is called vector compression. And that is the subject of another QønApp.

_____

Footnote:
A U.S. Navy memory refresh problem was documented in a paper, õCMM Collision Test on the TOPAZVö by D. Gardner, J. Mikoljai and S. Wisher. Here is an excerpt from that paper:
õThe CMM refresh collision problem occurs when the wrong data is read from CMM memory due to a =collisionøbetween an incoming memory read and an internal memory refresh clock. Collisions between two asynchronous signals, like read and refresh on the CMM, are normal and an arbitration circuit is provided to decide which signal has priority. This arbitration circuitry on the CMM however, occasionally allows the two signals to interfere with each other and causes the read address to be corrupted. This causes the read cycle to return the wrong data. This error occurrence ranges from 1 in a million to 1 in several billion.ö
An acceptable amount of error? Not when such refresh problems cause a õblipö to appear on a sailors SONAR screen and then disappear. Using a HiLevel TOPAZV test system with its exclusive pattern-match mode and automation tools, the Navy engineers were able to create the unique conditions required to manifest the collision phenomenon. Their discovery led to a redesign of the CMM arbitration circuitry; a measure that might have required an act of Congress, but in reality only required some clever users of a HiLevel tester to prove the proper course of action. It may be worth mentioning here that the HiLevel TOPAZV systems employ some DRAM as vector memory.