

How many of you have tried to cut on coffee? or even quit drinking coffee altogether? I guess a lot. Well, personally I've given up on trying but you know what? there's actually something worse than having 8 cups of coffee per day - it's the muffins that come with them! (or any other form of candy for that matter). The evil muffin of muffin-compliant just sits there in the cafeteria waiting to attack when you're most vulnerable - holding a cup of coffee in your hand.

Now things get really out of control at those professional seminars and conferences where the agenda [hopefully] varies but one thing remains in common - plenty of coffee and cakes. Recently I had to risk temptation yet again when I took the [Verification Leadership Seminar](#) at Ace Verification (given by Akiva Michelson, CTO & Founder).

I'll admit to it right away - I had a lot of coffee plus an average of 5 to 6 croissants a day (it was a 3-day thing) but it was really a small price to pay for an insightful view on various verification approaches and a series of interesting discussions on today's challenges in leading verification projects.

For example, we talked about the problem of marketing the verification profession to VLSI engineers. Traditionally, verification has been less appealing than other professions such as logic design, partly because verification as we know it today simply didn't exist before. This makes it a quite a challenge for verification leaders to recruit and keep good engineers on their team. Try to ask new graduates what they would prefer - design some module or verify it? What do you think the answer is going to be?

In another discussion we tried to come up with some guidelines for building a productive verification team, how to split the debug and environment writing work among the team - should there be a clear distinction between people who develop the code and the people who execute it? or maybe it's better that everybody does little of both?

It was also interesting to see how different companies have different views on the complex love-hate relationship that verifiers have with designers. I guess that subject alone deserves its own study. We tried to find out, for instance, just how deep you would expect a verification engineer to dig into the design when a failure occurred. Would you want your verifier to spend time on reverse-engineering the RTL in order to understand the root cause? or maybe it would

be enough to send the failure details over to the designer? but then what if the bug is merely a bad configuration or an environment issue?

Other discussions were quite technical in nature. For example, I remember one dramatic session about the pros and cons of aspect oriented HVLs vs. object oriented HVLs. I was surprised to see opinions varying quite a bit there. Seems like not everybody is so keen on AOP, even those who work with e. Oh, and we had a long talk about coverage and ways to make coverage goals more feasible, which reminded me of a rule I once tried to establish back at PMC-Sierra : Cross-coverage points with 3-dimensions or more must get manager approval.

As expected, we also mentioned the less-common verification techniques such as Formal, Assertions, etc. and found out yet again that there's probably a good reason why they still remain less-common. Ok, I'll stop here and refer you to the [course website](#) where some of the presentations are available, and more information can be found.

Summing up, it was a good seminar. It was interesting and also fun, and I think all of the participants are now feeling a bit relieved knowing that colleagues at other companies are facing similar problems and that there are no quick solutions to everything. I believe the challenges in managing and executing verification projects these days are manifold and call for creativity not less than adhering to proven methodologies. That's what makes it so exciting!

Disclaimer: This review was written in 2007