

Let's unveil the truth behind some of the common myths out there!

1. **This is legacy code no need to verify it** - Hold your horses! Are you 100% sure that you're dealing with silicon proven code? Are you sure that nobody's touched it since it last worked?
2. **I can come up with a patch in 5 minutes** - That's OK as long as you make sure you don't end up with a verification environment that looks like a bunch of patches hooked together. Ask yourself how easily it would be to modify or fix your environment a week from today? Isn't it worth it to take a couple of more minutes and write a robust code?
3. **Start checking and plan as you go** - A big no no! Don't be tempted into this. Even if your task looks like a piece of cake, always plan in advance. You'll be amazed to see how many problems can be completely avoided. Remember the 5 P's: Proper Planning Prevents Poor Performance.
4. **This is really simple, no need for a test plan** - Consider your test plan document as your working contract. Whatever you put in it defines the requirements for your current work. If it's a simple task then just write a test plan on half a page.
5. **Verification is not the product, no need to keep software standards** - True, verification is not the product but still - when you're dealing with thousands lines of code, you'd better make sure there is a certain level of consistency, let alone programming errors.
6. **Don't have time to add comments** - Remember the last time you spent half a day on reverse engineering somebody else's code? How about your own code? Better yet, start each test with comments explaining the steps in the test and keep them up to date.
7. **Oh I know! Let's just force the signal from outside** - Forced wires have this tendency to get forgotten along the way and then reappear at a later stage, usually a week before tapeout. So be extra careful.
8. **Must have regression running in the background all the time** - Regression runs alone can't do the job. You must have a Regression Sitter to monitor and analyze the results or else - you're just wearing out your servers.
9. **We have reached 100% coverage there's no point in running more tests** - Not really. Your coverage model can only capture what you thought about in advance . Obviously a random test bench is capable of generating additional scenarios that might reveal a bug, so don't stop at 100%. Instead - enhance the functional coverage model.
10. **Verifiers should be looking for bugs** - This is a common misconception of what

verification is all about. Verifiers should be focused on building a well constructed, robust and complete test bench. Bugs will come out on their own.