Draft Need _ 20180205B

SJTAG Need:

o *More implementation than things in the PAR (e.g., First Red block) Need more than what is just listed in first bullet. Need as part of design so it can be part of the end game (DFT, DFM, etc.).*

o *Still need some reference to "What is a good test?" "What constitutes a Test Coverage?" Everyone is working on the same thing (going in same direction) using standard as a template. Not in formal Need, but part of standard. May be some scenarios where user does not care about level of test coverage, but more just about if topology is operational. Some cases where test does not even come into the occasion (e.g., programming use case). These are still requirements for what the test infrastructure has to accomplish. Needs of a product may be variable, but may not be of interest to some people in the product life cycle. Probably don't need to dig into such details for the Need statement. Need to improve capability of what can be done as part of test access. But how can you validate such improvement? Can't deal with entire product development life cycle. The way we use SJTAG is going to be different whether we use it for DVT, Integration Test, Mfg. Test, etc. Ultimately, we need to know what are all the things we need to be handling as SJTAG. That is what should go in the Needs section. The Needs section is the only section where this kind of discussion can go. Do we need to flesh all this out in the PAR. Probably not. This is what the "word smithing" is all about. This is what the initiative group originally did with the perspectives from that team. There are additional perspectives now in this team. We are not driving a capability into hardware. Instead, we are attempting to leverage what is available in these devices and assemblies to perform enhanced capabilities. Need to know what is available to work with (Terry's top down idea). Some features in a device may not lend themselves to using at a higher level. It cause more problems at the assembly level then help. 1149.1-2013 tried to identify the Test Mode to help caution for cases of catastrophic state change of a board requiring a total reboot after a test. None of these features are visible with the latest interfaces on market.*