Building Testability into FPGA and ASIC Designs

By Matthew Noonan, Project Manager, Resource Group’s Embedded Systems & Solutions

Introduction

This paper discusses how the architecture for FPGAs and ASICs should be defined to improve the testability, quality, effort and hence cost of the verification activities.

The Problem

Firmware within the safety-critical industry is a no-man’s land. As complex electronics, the FPGA and ASIC of a design can be deemed part of the hardware engineer’s role. However, in structured code using software architecture, they can be seen as best developed by software engineers. When producing an FPGA/ASIC based product, whether you treat it as hardware or software can make an enormous difference to the cost and effort of verification or even whether your product will be certifiable or not.

Verification of a hardware platform tends to be based on a ‘black-box’ approach. This is normally fine for small and simple ASIC/FPGA designs, but what can be done when this doesn’t give the required test coverage of a complex system? As the size of ASIC/FPGA cores is rapidly growing this problem is only getting worse. The more functionality between the pins that can be driven and the pins that can be observed, the harder it is to develop tests that are robust to change and fully demonstrate the functionality under test.

ASIC/FPGA architectures mean it is not easy to drive full test coverage entirely from the outside.
The Solution

Software practices, long established for the certification of microcontrollers in high integrity systems can, through the use of good architecture, break the target down into its constituent parts. This allows the definition of the black box to be given to a piece of functionality or even an RTL file. Through a comprehensive suite of unit level and integration level tests Resource Group achieve full statement and branch coverage as well as MC/DC if necessary.

Recent global statistics put the estimated debug time of an average FPGA development as 50% of the overall effort if no testing is employed. The benefits of a good architecture and using unit test followed by integration level test is that overall effort is reduced and hence costs are saved compared to a debug based verification. With unit testing, components can be tested in smaller blocks, allowing design faults to be found and removed earlier in the process.

With the architecture allowing smaller block sized testing simulation times for tests are faster, allowing the tester to refine their test quicker and more productively. This can also allow identification and removal of all functional issues earlier in the program, allowing timely gate level netlist simulation runs to be entered into with higher confidence of finding no errors.

Unit level tests are also more easily maintained as each test is only dependent on the functionality of the individual unit design and not the whole unit under test. As such, tests don’t need rewriting every time external components in their path to the outside world are changed.

These cost savings increase with more complex implementations as software test methodologies allow the observation of internal signals. With this visibility, there is less need for complex test stimuli to drive combinations at a whole target level. Test combinations can be exercised at a smaller unit level, which are easier to control.

The Practice

To allow these practices to be used in an ASIC/FPGA implementation, the testing must be planned in at the design stage.

To allow the appropriate visibility of unit’s output signals, some outside world access must be granted to see those signals. One (of many) possible solutions is to provide visibility of the output signals of a component to the outside world either through space I/O pins or memory interfaces. Another approach can be to build each component into its own test harness and test it in isolation.
‘Keep it simple’ with Spare I/O

Let us consider the approach of making the output pins available to the outside world by using spare I/O pins. Attaching these to the critical internal signals will give a window into internal nodes and allow the splitting of one black-box into several smaller ones through detailed design testing. This obviously only works if you have the pin capacity to cover all the necessary internal signals.

---

Pros
- Simplest solution
- No additional on-board logic
- No additional test scenarios
- Greater test maintainability

Cons
- Need spare I/O pin capacity to cover all necessary internal signals
- Tests still need to be driven from their external inputs so changes to other functionality within the target may affect completed tests
The Memory Solution

An alternative method could be to define an area of internal memory with external test port access which will allow key values to be either passed through memory for observation or simply copied to memory for record. This will take a lot less pins than the I/O pin solution but may have some impacts on your timings. Also, as part of the implementation, the memory itself will need testing.
Fully integrated test harness

By integrating each unit with an on-board test port harness, complete control can be given to the tester. This can allow full robustness testing of each unit, on target through analysis of internal signals and through easy insertion and observation of test cases. Due to the level of internal access provided by this solution, the test harness definition and test execution should be performed by an FPGA verification team experienced in such architectures, such as Resource Group, to ensure that certification arguments are not violated. This process involves careful co-ordination and planning between unit test and integration test level scenarios to ensure that end-to-end functionality is fully demonstrated.

Isolation component test using a test harness
The Conclusion

With the growing complexity and size of ASIC and FPGA solutions, the verification methods derived from hardware assurance processes make providing the necessary level of testing to allow certification of systems for high integrity applications harder and more costly. Software methodologies, developed over many years, can be applied to solve these issues but require planning from a design stage to ensure that the correct on-board facilities are provided or that the testing approach is considered from the outset.

Our Embedded Systems & Solutions division can advise and support you on all aspects of Validation and Verification, at system, software and firmware level. Our team has many years of hands-on experience across a range of industry sectors, including Aerospace, Automotive, Rail and Defence. Whatever your system or toolset, our team is ready to discuss your needs and deliver you a solution.

Contact us on: matthew.noonan@ResourceGroup.co.uk

+44 (0) +44 (0) 1905 368 556