

A Top-Down VHDL Design Environment

Hubert W. Mueller, Jr.
Hughes Aircraft Company
El Segundo, CA 90245

Abstract

A top-down VHDL design environment was developed to design a digital signal processor for a space application. VHDL was used for the entire design from behavioral simulation through the physical or structural register transfer level, RTL, descriptions. A unique built-in test approach used VHDL to design all test circuitry thereby enabling full gate array ASIC simulation via VHDL prior to synthesis. Custom SRAMs were also added to the gate array VHDL library to provide dense on chip memory.

Design Constraints

A demonstration of a top-down VHDL digital design environment was a key government requirement. VHDL was to be used for all levels of simulation and design. One ASIC had to be fabricated on an accelerated schedule as a proof of CAD concept. Other requirements for space operation were: low power consumption, light weight and rapid autonomous fault detection and recovery. The specified radiation tolerance could be met by a limited number of digital ASIC foundries for radiation hardened technologies. These technologies must operate after high total accumulated radiation dose levels, operate through high radiation rates, exhibit low single event upset rates and be immune to latch-up. Low power and weight are always required for space applications and were factors in choosing the approach for fault detection and recovery.

Design Approach

The digital signal processor was part of a complex system wherein a sensor provided

input to the signal processor and the results were output to a computer system for further processing. The design consisted of 360 ASICs of 9 types. A "hardwired design" of fixed form solution with programmable parameters was chosen since fully programmable signal processors were not power and weight efficient enough for implementing the required algorithms. The computer system controls the signal processor initialization, fault detection and configuration control. However, built-in self-test, BIST, was used in the signal processor since it was prohibitively expensive for the host computer system to be used as an ASIC tester to store test vectors and expected results.

Design Process

The following diagram illustrates the basic design flow. Functional test cases were provided from the system algorithm test bed. They augmented the written signal processor specification. These test case stimuli and responses were generated to exactly match the expected hardware response. An early partitioning to the board, multi-chip module, and ASIC level was also made to enable functional test case simulation of subsets of the design to reduce simulation times. The importance of having the "truth" from the algorithm test bed available at all simulation levels cannot be overstated for a complex design. The best interpretation of the written specification is subject to errors which are better caught during the behavioral simulations before detailed RTL design.

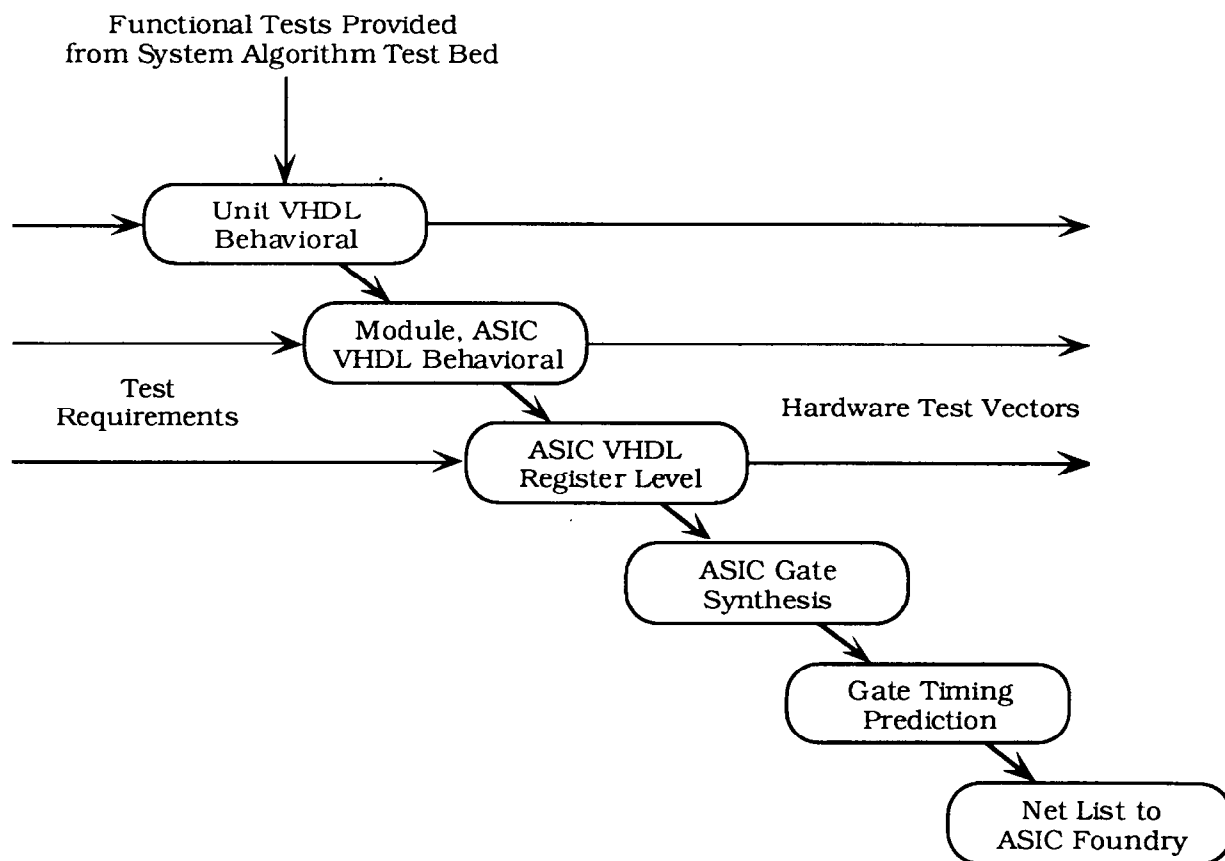
A benefit of using VHDL for the behavioral simulations is that the interface descriptions at all levels can be retained. Detailed timing was, of course, added by

the designers as the design progressed to the ASIC level since the algorithm test bed provided the data stimuli independently of the timing. In fact it is disappointing to many when they learn that the interface descriptions are often all that is retained of the behavioral VHDL when the ASIC RTL descriptions are generated. Two unusual design approaches were implemented during the RTL description:

- All test structures were defined via VHDL - unlike the approach where test is added after synthesis.
- Custom SRAMs were added to the vendor's VHDL library.

SRAMs implemented on gate arrays are normally not area efficient since array gates are used to configure the memory cells. Two

foundries also produce 256k radiation hard SRAMs and these were the basis of the added custom VHDL SRAM library elements. The wafer fabrication was much like full custom wafers. Since the SRAM architecture was known well before the ASIC design completion it was possible to start wafer fabrication early to minimize possible schedule impact. The SRAMs were placed and surrounded by gates for later personalization. The normal gate array power, ground and clock distributions were disrupted by the custom SRAMs and had to be custom routed for proper operation. SRAM built-in-test structures were designed in VHDL. The SRAMs were fully tested after the ASIC setup, control and test interface and BIST were completed. Memory tests performed all tests necessary to completely verify all memory cells.



Design Flow – VHDL was used for the entire design description

Testability was considered for all phases of the life cycle from ASIC foundry testing

through on-orbit fault detection and isolation. A single interface design was time

shared for both the ASIC setup and control as well as test control to minimize circuitry. Some inherent fault tolerance not requiring reconfiguration, such as memory error detection and correction for critical parameters, was included in the design. However, power and weight constraints limited this and also eliminated self check pairs for error detection. Fortunately robust algorithms downstream of the signal processor and the operational modes permitted periodic hardware testing for faults. The on-orbit testing was based on the testing needed for ASIC, multi-chip module, module and unit testing. Foundry ASIC testing and burn-in tests require coverage of greater than 95% but test coverage of 80% to 90% may well be acceptable for subsequent testing. Little failure mode data after the "infant mortality" failure period is available. The on-orbit testing and test control had to be simulated and therefore all test structures on the ASICs were designed as part of the VHDL descriptions. This approach also maximizes design reuse if the ASIC is fabricated at a second source.

The first level of test consisted of loop tests through the ASIC setup, control and test interface. Then boundary scan, BIST and memory test were designed via VHDL. Boundary scan was used to verify the inter ASIC connections on multi-chip modules and modules as required. Each ASIC was partitioned so that unique BIST structures could be designed for each partition to reduce test time and provide higher coverage. Each partition had a BIST structure containing a test length clock counter and pseudo random pattern generator which is seeded from the computer. The resultant partition test signature is read to the computer for verification. This approach allows the test coverage to be improved, for foundry tests for example, by refining the combinations of pattern seed and test length. This is possible even after the ASICs have been fabricated. ASIC speed degradation due to radiation effects can be detected on-orbit since BIST runs at normal clock speeds.

Proof of Concept ASIC

A dual filter ASIC was fabricated to demonstrate the design environment. The

following table shows the CAD software used for each design phase.

Function	Software Source
VHDL Behavioral / RTL Simulation	Vantage
Logic Synthesis	Synopsis
Timing Analysis	Mentor QuickSim
Statistical Fault Coverage	Mentor QuickFault
Deterministic Fault Coverage	Mentor QuickFault
Test Data Filters	Hughes

CAD Software Packages

The design contained 50,866 gates and 73,728 bits of SRAM. The memory consisted of eight memories two each of the following configurations: 1012x20, 512x20, 128x20 and 128x28. These would have required 26 off-chip devices if byte wide SRAMs were used. Memory bandwidths of over 20 MHz precluded a shared memory architecture. The VHDL boundary scan implementation was unique to this design since it did not target the existing vendor I/O scan cells since they were not yet in the library. SRAM I/O were also brought out to allow independent memory tests since this was the first time this much memory was added to a gate array and the ASIC foundry wanted to independently test the memory. The portion of gates used for test is shown in the following table:

Item	Gate Count	%
Total ASIC	50,866	
Core Test	6,135	12.1
Test/Control Interface	2,466	4.8
VHDL Boundary Scan	3,152	6.2
I/O Boundary Scan	1,345	2.6

Gate Counts for Test Structures

The 1,345 I/O boundary scan gates shown in the table were not used but are included in the gate count generated by the CAD software since they are included in the I/O cells. A VHDL boundary scan was coded in the core of the chip. In this configuration 25.7% of the gates were used for test. Future designs will use the scan built into the I/O cells and not require custom VHDL boundary scan. The test gate percentage would then have been 19.5% for this design. This ASIC was partitioned into 16

areas to improve test speed. This resulted in the relatively high gate count for the core tests. The number of partitions can be traded against test time to lower the gate count. Note that the entire Test/Control Interface gate count is included in the test gate count when in fact many of these gates are time shared between ASIC control and test. The actual test gate percentage could be reduced by approximately 2%.

An important aspect of test coverage evaluation is the workstation run time. It is easy to overlook this time when considering the total design cycle time. The fault coverage and the simulation times for several functional tests and BIST were run on an HP-735 with 128 Mbyte of RAM. The results are shown in the following table:

Test	Fault Coverage %	QuickSim Run Time, hours	QuickGrade Run Time, hours	QuickFault Run Time, hours
Typical Scene	51.4	0.8	3.9	
Sawtooth	43.0	0.7	3.6	
14,000 Gate	45	0.1	0.6	168*
BIST 1,000,000 clocks	79.9	25.5	127.5	

* Test was stopped, before completion, after 7 days

Fault Coverage and Simulation Times

The functional tests, a typical scene, a test sawtooth wave and a 14,000 gate test clearly do not provide sufficient test coverage. The 79.9% BIST coverage is lower than the actual coverage will be for four reasons: 1) The test/control interface gates normally verified via loop tests are included in the total gate count; 2) The unused boundary scan cells are included in the gate count; 3) Some memory control circuitry was not tested since the memories were tested externally; and 4) Some VHDL coding, constant values for example, resulted in gates which could not be tested. Future designs are expected to have BIST coverage of 85% to 90%. BIST simulation times for one million clocks required 25.5 hours in QuickSim and 127.5 hours for QuickGrade. It would have been prohibitive to use QuickFault to evaluate BIST since a test case on 14,000 gates ran for 168 hours without completing. Design cycle time is directly affected by the workstation processor speed and memory.

Lessons Learned

The key lessons learned during the demonstration were:

- VHDL top-down design is invaluable for verifying the signal processing architecture algorithm implementation early in the design process.
- Custom SRAMs can be incorporated on a gate array and in the VHDL library.
- Workstations can never have too fast a processor or too much memory.
- BIST can be designed via VHDL thus providing full simulation prior to synthesis. It also provides a greater degree of transportability of designs to alternate foundries.