Using WAVES in a Top-Down Design Methodology

Christopher J. Flynn, Frederick G. Hall, James P. Hanna, and Mark T. Pronobis

Rome Laboratory
Design and Diagnostics Branch
Griffiss AFB, NY 13441-4505
(315)330-2813
flynnc@rl.af.mil

Abstract:

A typical electronic circuit or system development cycle begins with designers who are tasked with developing, for example, their company's next generation processor. They will try to tweak as much performance as possible out of their designs and generally, when they are satisfied with the design, they will "throw it over the wall" to the test engineers, who then have to generate the test programs and vector sets. There is a need for a methodology that employs industry standards, such as the VHSIC Hardware Description Language (VHDL)[1] and the Waveform and Vector Exchange Specification (WAVES)[2], which encourages design and test engineers to work together throughout the design/test cycle. This paper describes a technology project in which the IEEE standards, VHDL and WAVES, are being utilized within a top-down design methodology, and throughout the design through test cycle of a modest size VLSI chip.

I. Introduction:

The use of VHDL in a top-down design provides many benefits including modeling at multiple levels of abstraction, technology independence and validation through simulation. The benefit of using WAVES is that it is a standard format of the stimulus and response information that can be freely exchanged between multiple simulation and test platforms. This project addresses the lack of a recommended methodology for using VHDL and WAVES in a top-down design. The project will develop and recommend a design through test methodology incorporating the use of VHDL and WAVES. This will demonstrate the application of the standard stimulus and response format, WAVES, for quickly verifying VHDL models, reducing the test development time, and the time spent developing the test vectors.

Behavioral and Register Transfer Level (RTL) VHDL models of a modest size VLSI chip will be developed in a top-down manner and WAVES data sets will be used as the stimulus for all functional and timing verification. The design will be fabricated in a 1.2 micron CMOS technology through the MOSIS foundry system. The Test Description Language (TDL)[3] data formats will be used to capture the test requirements and the INdustry Shared TEst Processor (IN-STEP)[4] will be used to generate the test program. The same WAVES data set used for design verification will be automatically translated into the vector format utilized by Rome Laboratory's Teradyne J953 microcircuit tester for electrical testing purposes.

Each of the following sections intend to provide a brief overview of a particular portion of this project. Section 2 describes what is being designed, section 3 describes the VHDL models being developed, section 4 includes a discussion of WAVES and our methodology for using it, section 5 describes the electrical testing portion of the project and section 6 describes the testability features of the design.

II. Multiple Technology Processor (MTP) Overview:

An architecture of modest complexity will be used for this effort. This will give us a design in which several technical research areas in the organization, including Top-Down VHDL design, WAVES, electrical test and testability could be addressed. This section gives a high level overview of the architecture that is being designed.

The MTP is a single-chip, 32-bit integer arithmetic processor (Figure 1). Resources include twenty nine 32-bit data registers and a 32-bit arithmetic logic unit. The arithmetic logic unit is capable of executing the MTP's 26 instructions, which include arithmetic (two's complement signed), Boolean (bitwise), and conditional operations. The data path is a quad-bus structure providing single cycle register data processing. The control, load, execute, and store states are implemented as single processor instruction cycles. The MTP is a static design and will be packaged in an 84 pin, Pin Grid Array (PGA).

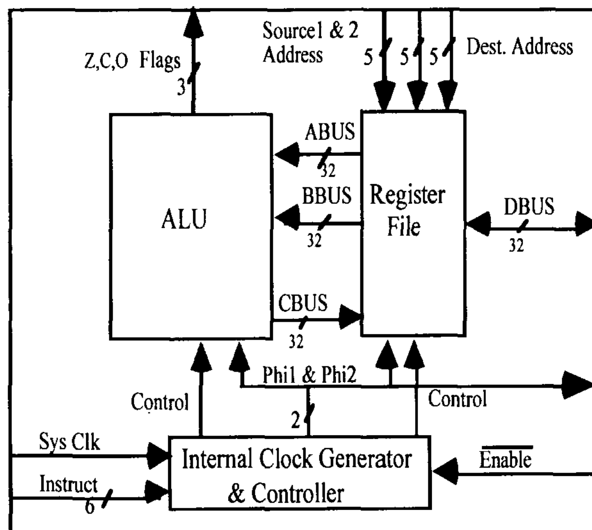

Figure 1. MTP Block Diagram

The MTP's register file, as mentioned above, offers twenty-nine 32-bit registers. The first eight registers, R0 - R7, are used as full function data registers. These registers are used for loading data into or reading data out of the processor. There is a "zero_register", R25, which is hard-wired to a default value of zero. This register can not be modified so it always contains the value of zero. All registers can be utilized for ALU operations, (i.e. read from, or written to, by ALU instructions). The internal clock generator/controller supplies the internal two-phase non overlapping clocks phi1 and phi2, as well as the required control signals for the ALU and register file. There are three internal 32-bit data busses, the A, B and C bus, and one 32-bit I/O bus, the D bus. There are three processor status flags, all active high. The Zero (Z) flag is set when the result of an instruction is zero, the Condition (C) flag is set when the result of a conditional instruction is true and the Overflow (O) flag is set when the result of the instruction can not be represented by a 32-bit signed integer.

III. VHDL Modeling:

VHDL (IEEE Std 1076) is a formal notation intended for use in all phases of the creation of electronic systems. It supports the development, verification, synthesis, and testing of hardware designs, the communication of hardware design data, and the maintenance, modification, and procurement of hardware[1]. The MTP design team has adopted the following definitions for the three levels of VHDL models in our top-down design methodology. *Behavioral Model:* A simulation model that uses abstract data types and enumerated types to convey the functionality of the design, without specifying implementation details. They are useful for evolving a specification before doing any detailed design. There will be multiple iterations of behavioral models, where each iteration will incorporate added design detail, before the refinement to the RTL model will be made. *RTL Model:* A simulation model which is concerned with the detailed verification of data and control sequencing both within and between blocks. They are described by functional operators and procedural constructs. *Structural Model:* A netlist representation that specifies which hardware components are to be used and how they are to be interconnected[5].

The highest level VHDL model uses Boolean and user specified enumerated types. It contains a single entity/architecture pair and uses only one process. This high level model

will be used to verify the functionality of the MTP. This will lead to a behavioral model which will include the individual components; the ALU, register file and clock/controller. Each one will be described as a separate behavioral model. This behavioral model will define the interface requirements between the components, as well as perform the first partitioning of the design. Further partitioning within each block will occur as we proceed to the RTL model. The continued partitioning will help in managing the complexity of each block, reducing the design into manageable portions to insure a successful design.

IV. WAVES for Design and Test:

WAVES (IEEE Std 1029.1) is the industry standard representation and exchange format for digital stimulus and response data. It provides a powerful support mechanism for concurrent engineering practices by allowing digital stimulus and response information to be freely exchanged between multiple simulation and test platforms. WAVES is defined as a syntactic subset of VHDL, and as such, can be simulated against the VHDL model during design to verify the functionality and timing of the design as it progresses. Further, when devices are fabricated, the same WAVES data set can be used in the electrical test process to assure that the same stimulus that was used during design is applied during electrical test.

A. Methodology:

The methodology for applying WAVES for design and test to the VHDL modeling process at each of the levels of the design hierarchy is represented graphically in Figure 2. It uses a VHDL behavioral testbench to verify the behavioral VHDL model. The behavioral testbench reads an external instruction file(s) that is similar to an assembly language program file. At the next level of refinement a WAVES Level II dataset is used. The WAVES Level II dataset reads the same instruction file(s) and translates the instructions to scalar signals to apply to the behavioral and RTL-level VHDL models. The next lower level of abstraction for the VHDL model is the gate level. The gate-level model is simulated with the WAVES Level II dataset. At this point a WAVES Level II to Level I translator is used to generate the WAVES Level I external file(s). For electrical test, the WAVES Level I dataset is translated to the Teradyne J953 microcircuit tester language and the fabricated parts are tested against the same vectors used during design. This methodology for using WAVES throughout the design and test process assures that the same vector information is applied across the various levels of design and test in a consistent manner. This allows rapid and consistent regression testing to be applied throughout the hardware development, fabrication, and test process.

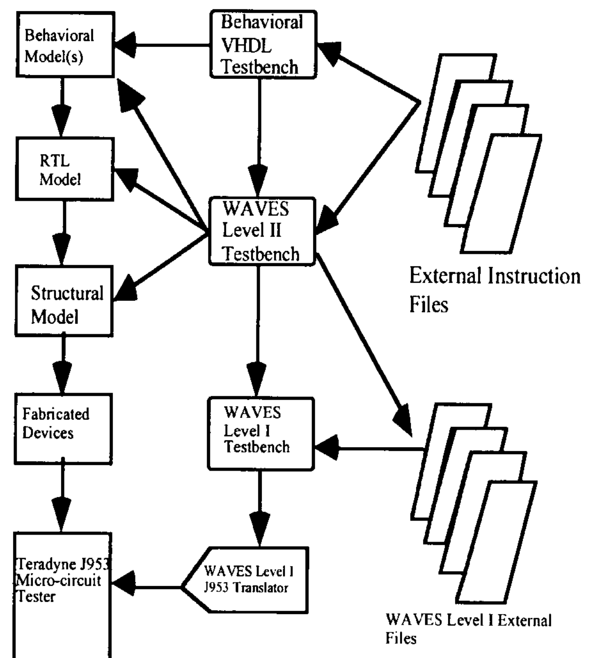

Figure 2. VHDL Models and WAVES Design through Test Flow

B. WAVES Dataset Development:

It is recommended that each individual site that intends to use WAVES build their own WAVES Site Configuration (WSC). A typical WSC includes a set of WAVES packages that contain such things as logic values, pin codes, and frame sets that are specific to the simulation environment and test equipment being used at the particular site. The contents of the WSC will be common to every WAVES dataset developed at the site. It should be developed by the site's WAVES expert and will eliminate the unnecessary reduplication of

common declarations and packages used by every WAVES dataset developed at the site. There are many other benefits for developing a WSC, some of these include: decreased WAVES dataset development time - as a lot of the work is already completed - ease of conversion of WAVES datasets to test environment, as they are tailored to a particular tester, and a shorter WAVES learning curve - as many of the details of WAVES are hidden to the user. The generation and usage of a WSC will result in a consistent usage of WAVES at a particular site.

The completion of a particular WAVES dataset requires the WSC and some additional UUT specific WAVES files. These include an external file, which contains the test vectors to be applied to the VHDL model being verified, a test pins declaration, which is an enumerated type which names all the pins that are in the VHDL model, and the waveform generator procedure, which reads the external test vector file and generates the waveforms to be applied to the model at the appropriate times. These three WAVES files and the WSC are all that is required to generate a WAVES dataset. Finally, a simple testbench is written to run the simulation of the VHDL model using the WAVES dataset.

The Rome Laboratory WSC has been specifically written for the IEEE STD 1164 and the Teradyne J953 tester. WAVES datasets are being generated to individually test some of the more complex components of the MTP, such as the VHDL model of clock generator/controller, as well as the complete MTP VHDL models.

V. Electrical Test:

A test philosophy for the MTP was formulated in discussions between the design and test engineers. A standard battery of dc and ac parametric, as well as functional tests will be performed. Functional testing of the register file of the MTP will based on a memory test model; patterns of various complexities will be stored and read from the various registers to see if there are any bit/word anomalies caused by the stored pattern. The ALU portion of the MTP will be tested using functional test vectors generated from the instruction set. It is from these test philosophy discussions that the WAVES data set, discussed in the previous section, will be generated.

The electrical testing portion of this project ties together various test automation concepts. The test program(s) will be generated using the WAVES data set (for functional vectors and timing) and the TDL data generated in a collaboration between the design and test engineers (for parametric information). One of the key reasons for generating the test information in these formats is their "tester-independence". Using the appropriate translation tools, the information can be transported and run on any automated test system that meets the minimum performance requirements. In addition to tester independence, having the test information in these standard formats allows the use of another tool, IN-STEP, to automatically generate the test program in the native language of our target test system. By automating the generation of the test programs (using proven test-code templates), we reduce the chance of manual programming errors and have standard code for each individual test (i.e. VOL, VOH, etc.), leaving no room for an individual programmer's interpretation of how a test should be written.

Using the various test automation tools, we will be performing both detailed characterization testing (ac/dc parametrics, functional, and timing tests) and less stringent (pass/fail) "production" testing. From the conception of this effort only vague top-level performance goals for the MTP were specified. Data gathered during characterization testing will be the primary process used in generating the "hard numbers" used in the product specification. Due to the small sample sizes involved in this effort, it will be difficult to generate the product specification based on traditional statistical methods. At this time, setting the minimum and maximum specification limits will be based on the mean, variance, and standard deviation values of our sample population.

One of the differences of characterization testing versus production testing is apparent in the early phase of this effort. The only benchmarks the test engineers can use for timing and parametric data are those found during

simulation of the device. With these values as a starting point (as opposed to a product specification), the test engineer's job is made a little more difficult because the detailed electrical characterization testing will have to find the "actual" device parameters. We will consider the first MOSIS production run to be characterization samples, with our specification numbers based on the data from these devices. Once the MTP design and performance has been verified using detailed characterization testing, future MOSIS runs will be treated as "production" devices. These devices will be subjected to a suite of production-type electrical tests to demonstrate the degree of success our characterization tests had in generating accurate product specification values.

VI. Testability:

The testability strategy for the MTP will include Built-In Test (BIT) technology designed to allow for the easy manufacturing verification of the fabricated chips. Internal test points will be accessible for easy functional verification. The testability architecture for the MTP implements IEEE-1149.1 boundary scan, a full scan path design, and an on-board Random Test Generator (RTG) and Signature Analyzer (SA). The capability to capture the state of the A, B and C bus is included through a multiplexer and a parallel to serial shift register

The RTG provides random test vectors to be used for manufacturing level test of the ALU. The RTG will be implemented with a Linear Feedback Shift Register (LFSR) comprised of scannable flip-flops. The SA provides for the compression of the scan vectors to a 24-bit unambiguous signature implemented with a LFSR. The register file has externally accessible address and data pins and will be algorithmically tested from the external pins. A WAVES test set will be developed to simulate the testability architecture in VHDL. Fault grading will be performed using gate level Hierarchical Integrated Test Simulator (HITS)[6] models reporting to the Mil-Std-883, Method 5012, "Fault Coverage Measurement for Digital Microcircuits".

VII. Status & Conclusions:

A modest size design is being used do develop a methodology for using industry standard data representations and methods (VHDL, WAVES, IEEE 1149.1) in a top-down design through test project. Industry standards should be adopted whenever possible as they provide a multitude of benefits. First and foremost, dependency upon one specific tool vendor will be avoided, secondly, industry standard formats are much more stable that vendor formats, and finally they provide a free exchange of information between multiple tools and host platforms. This project will demonstrate an example where test engineers and design engineers have discussed aspects of interest to both parties while still early enough in the product development phase to make a difference. Testability analysis is also being incorporated into the project to evaluate the manufacturing process used to fabricate the design.

At the time of paper submission, the highest level VHDL model, which uses Boolean and user defined enumerated types has been written and functional verification of this model is complete. The behavioral model has been written and is being verified with the test vector set developed jointly by design and test engineers. The WAVES Level II dataset is being developed based on the definition of the instruction codes that will be processed by the behavioral model. The RTG and SA have been designed, integrated and tested together.

VIII. References:

[1]. "IEEE Standard VHDL Language Reference Manual (VHDL-LRM) IEEE Std 1076-1987", 1988, The Institute of Electrical and Electronics Engineers, Inc. New York, NY.

[2]. "IEEE Standard for Waveform and Vector Exchange (WAVES)," Institute of Electrical and Electronics Engineers, IEEE Std 1029.1-1991.

[3]. "Working Paper for the Test Specification Language (TSL)", prepared by James P. Hanna, Rome Laboratory, Griffiss AFB, White Paper, March 1991.

[4]. "Final Technical Report IN-STEP User's Guide", Contract No. F30602-89-D-009, Task 3, Rome Research Corp., CDRL Sequence No. C007, Rome Laboratory, May 1992.

[5]. N. Berry, "A Flexible Architectural Synthesis Paradigm For VHDL," VHDL Users' Group Conference Proceedings, April 1991.

[6]. L. Hosley and M. Modi, "HITS - the Navy's new DATPG system," Proceedings, AUTOTESTCON, 1983, pp. 29-35.