AEROSPACE REPORT NO.
ATR-94(4778)-2

SDVS/VHDL Application Program Plan

30 September 1994

Prepared by

I. V. FILIPPENKO
Trusted Computer Systems Department
Computer Science and Technology Subdivision
Computer Systems Division
Engineering and Technology Group

Prepared for

DEPARTMENT OF DEFENSE
Ft. George G. Meade, MD 20744-6000

Engineering and Technology Group

THE AEROSPACE CORPORATION
El Segundo, California

AEROSPACE REPORT NO.
ATR-94(4778)-2

SDVS/VHDL APPLICATION PROGRAM PLAN

Prepared by

I. V. FILIPPENKO
Trusted Computer Systems Department
Computer Science and Technology Subdivision
Computer Systems Division
Engineering and Technology Group

30 September 1994

Engineering and Technology Group
THE AEROSPACE CORPORATION
El Segundo, CA 90245-4691

Prepared for

DEPARTMENT OF DEFENSE
Ft. George G. Meade, MD 20744-6000

SDVS/VHDL APPLICATION PROGRAM PLAN

Prepared by

I. V. Filippenko

I. V. Filippenko

Approved by

L. G. Marcus, Principal Investigator
Computer Assurance Section

D. B. Baker, Director
Trusted Computer Systems Department

Contents

1 Introduction

This report delineates a plan for accomplishing the *SDVS/VHDL Application* task, an effort on the part of The Aerospace Corporation (Aerospace) to specify and verify formally properties of a moderate-size hardware description, provided by the National Security Agency (NSA), of production integrated circuitry.

The description is to be written in the VHSIC Hardware Description Language (VHDL), and the specification/verification tool to be used is the State Delta Verification System (SDVS). The SDVS/VHDL Application will be directed towards demonstrating the suitability of SDVS to the verification of realistic VHDL hardware descriptions, stress-testing SDVS, and formulating strategies for further research and development in formal verification generally, and particularly in VHDL verification.

Aerospace and NSA have an ongoing commitment to investigating the utility of formal methods for DoD Programs. In particular, Aerospace continues its multi-year project to build an automated system for formal verification that can be used at all levels of the hierarchy of digital computer systems: the State Delta Verification System. The goal is to verify hardware from gate-level designs to high-level architecture, and to verify software from the microcode level to application programs written in high-level programming and hardware description languages.

NSA is interested in broadening and deepening its experience with formal methods, in order to better assess the value of applying such methods to DoD Programs. At the request of the NSA Director (DIRNSA), a Formal Methods Special Projects Office (SPO) has been established within NSA/R2 to focus attention on NSA's formal methods work. The SDVS project reports directly to the SPO, which has a strong technical staff whose areas of expertise include hardware verification, specification languages, and trusted distributed systems.

During the past four years, Aerospace has developed the theory and implementation of SDVS for the purpose of proving properties of VHDL descriptions [1, 2]. A primary objective of the SDVS/VHDL Application is to demonstrate this capability for production-scale hardware models.

It is certain that various enhancements to SDVS would improve the system's applicability to real-world verification problems. Thus, two other important purposes of the task described herein are to exercise the system rigorously and provide direction for its subsequent research and development cycles.

2 Background

In FY93, Aerospace investigated potential hardware verification applications in coordination with NSA/R2. A candidate SDVS/VHDL Application was identified at the onset of FY94, consisting of VHDL descriptions — developed in-house at

NSA [3] — for a set of commercial standard parts, the *Am7968/Am7969 TAXIchip™ (Transparent Asynchronous Xmitter-Receiver Interface) Integrated Circuits* designed by Advanced Micro Devices, Inc. (AMD) [4].

The TAXIchip set is being used in a prototype cryptographic device for interfacing with ATM communication networks, currently being built by NSA. NSA proposed this device as a good choice for the SDVS/VHDL Application by virtue of its being representative and relevant: the bit manipulations are typical of those found in digital devices of interest to NSA, and the chipset is likely to be retained in the final production unit. Thus, the target VHDL descriptions, which constitute a behavioral model of the chipset, could well become part of a bidding procurement package.

The verification process entails, as an essential step, the creation of formal specifications of the subject hardware descriptions from informal documentation. Although the TAXIchip documentation supplied to Aerospace by NSA is quite extensive, even the best documentation can suffer from ambiguities and omissions (this underscores the rationale for carrying out a formal verification in the first place). Hence, Aerospace's direct access to the TAXIchip VHDL developers at NSA is an important factor in the prospects for success of the Application task; indeed, it has already played a role.

In particular, the AMD TAXIchip documentation characterizes the chipset at a high level as follows:

> The Am7968 TAXIchip Transmitter and Am7969 TAXIchip Receiver chipset is a general-purpose interface for very high-speed (4 - 17.5 Mbytes/sec, 40 - 175 Mbaud serially) point-to-point communications over coaxial or fiber-optic media. The TAXIchip set emulates a pseudo-parallel register. They [*sic*] load data into one side and output it on the other, except in this case, the "other" side is separated by a long serial link. The speed of a TAXIchip system is adjustable over a range of frequencies, with parallel bus transfer rates of 4 Mbytes/sec at the low end, and up to 17.5 Mbytes/sec at the high end. The flexible bus interface scheme of the TAXIchip set accepts bytes that are either 8, 9, or 10 bits wide. Byte transfers can be Data or Command signaling.

This choice of VHDL application was approved officially in the SDVS FY94 TO&P (revision: 4 February 1994), whose task list states that Aerospace will:

> **V.c.** Draw up and gain NSA/R2 approval on a "VHDL Application Program Plan" that defines the scope and deliverables of the effort to verify the AM7968/AM7969 Taxichip VHDL Models provided by NSA. Undertake work on the application, as outlined in the program plan. Identify what SDVS lacks in order to perform this verification, as well as any changes that must be made to the VHDL application in order to apply SDVS.

The present report addresses the first requirement of Task V.c, and constitutes the corresponding deliverable item of the (revised) SDVS FY94 TO&P:

VI.e. Report detailing a VHDL Application Program Plan. This report defines the scope and deliverables of the effort to verify the selected VHDL application. Draft report due within 30 days of identification and NSA/R2 approval of selected application. NSA/R2 comments due to Aerospace within 28 days after receipt of draft; final version of report due within 30 days of receipt of NSA/R2 comments (Task V.c).

3 Objectives

The following are the principal objectives of the SDVS/VHDL Application task:

- Evaluate the capability of SDVS to address real-world hardware verification.
- Use SDVS to specify formally various levels of properties of the subject VHDL descriptions, and expose pertinent specification issues.
- To an appropriate degree, verify that the subject VHDL descriptions meet their formal specifications, and prepare for additional work in this direction.
- Enhance the SDVS implementation, as required.
- Where necessary, extend the logical theory underlying SDVS.
- Develop a significant SDVS VHDL verification example, suitable for demonstration and publication.
- Improve Aerospace's and NSA's understanding of how to incorporate formal methods into the hardware design process, and assess the attendant value added.
- Identify and prioritize topics for further research and development.
- Provide results of the specification/verification effort to NSA.

4 Discussion

In this Section we aim to:

1. provide an overview of the TAXIchip system;
2. outline the TAXIchip VHDL models for the SDVS/VHDL Application; and
3. review the VHDL subset currently incorporated in SDVS, identifying salient features of the Application models that will require additional implementation.

4.1 TAXIchip System

The TAXIchip Am7968 Transmitter/Am7969 Receiver chipset is an interface for the serial connection of two parallel-data hosts. In normal operational mode, each Transmitter/Receiver pair is connected over a private *serial link*, which can be a fiber-optic or copper medium.

Figure 1 exhibits a top-level block diagram for the TAXIchip system. It has been adapted from [4], as has other material in this Section.

N = 8, 9, or 10 parallel (data) bits

M = 4, 3, or 2 parallel (command) bits

N + M = 12

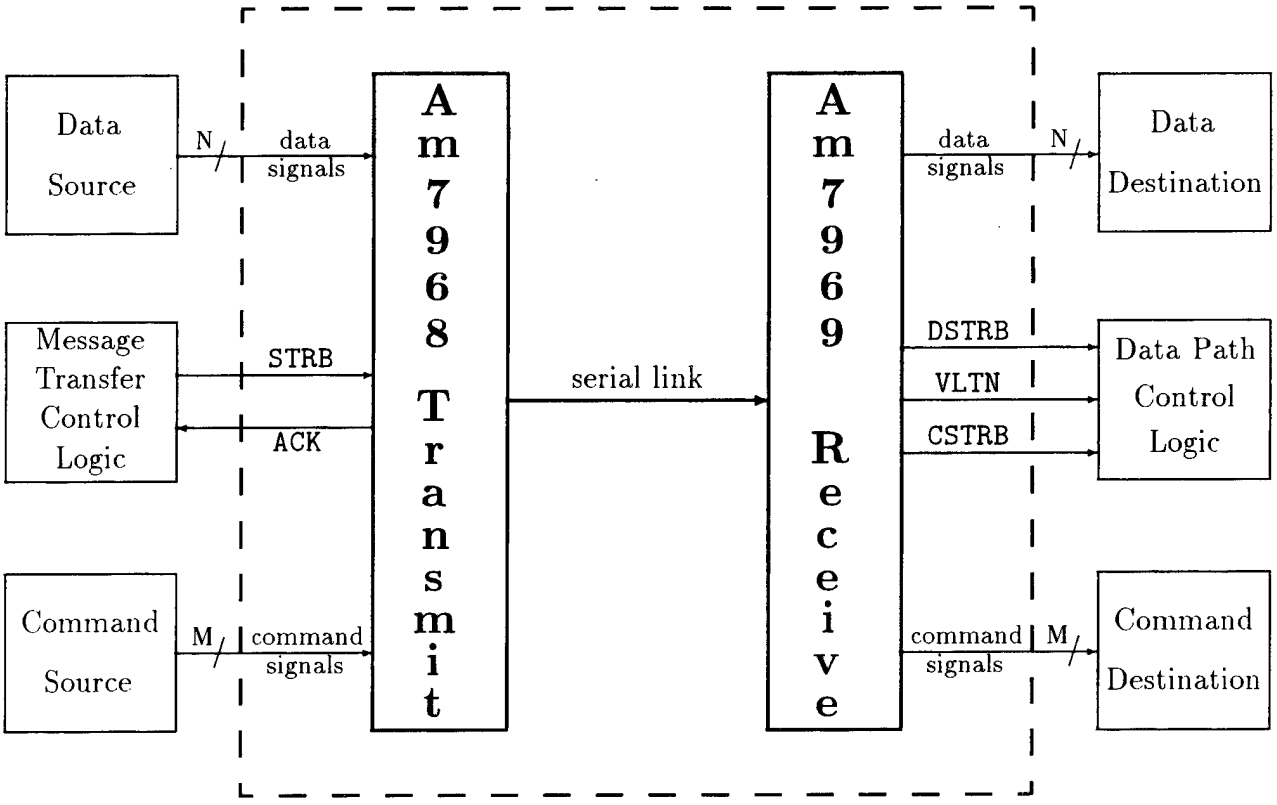

Figure 1: TAXIchip System Block Diagram

Each chip has twelve parallel interface pins designated to carry either *Data bits* (the usual data channel between the hosts) or *Command bits* (communication control from the sender). Three operational modes are selectable via the Data Mode Select pin on each chip: eight Data and four Command bits, nine Data and three Command bits, or ten Data and two Command bits. Commands, though occurring relatively infrequently, are given priority — their transmission overrides that of Data when both are present.

4.1.1 Transmitter Functional Description

The Transmitter accepts a parallel message from the sending host on its twelve input pins, using a *strobe/acknowledge* (STRB/ACK) handshake protocol. Incoming parallel bits are latched into the device, encoded in two stages (see below), serialized, and shifted out to the serial link. Switching between Data and Command is accomplished on the basis of the Command input pattern: if all Command bits are 0's, the Data bit pattern is latched; otherwise, the Command bit pattern is latched while the Data inputs are ignored.

4.1.2 Receiver Functional Description

The Receiver continuously deserializes the arriving serial bitstream, decodes the resulting parallel bit patterns, and routes the decoded Data or Command message appropriately to its twelve output pins. Latching a new Data output pattern does not affect the state of the Command output bits, and is accompanied by the synchronous pulse of a *data output strobe* (DSTRB). Similarly, any new Command pattern is latched without affecting the state of the Data outputs, and a synchronous *command output strobe* (CSTRB) is pulsed. Finally, a violation (VLTN) is flagged when any output is latched that does not correspond to a valid Data or Command pattern.

4.1.3 Coding Method

The Transmitter employs an encoding scheme based on the ANSI X3T9.5 Committee's *4B/5B* ("4-bit/5-bit") code, in combination with *NRZI* (*Non-Return-to-Zero-Invert-ones*) encoding. The Receiver uses the same method to decode incoming bitstreams, rendering the entire encoding/decoding process transparent to the host (source and destination) systems.

In 8-bit mode, the Transmitter uses two 4B/5B encoders to code the 8-bit parallel Data pattern into a 10-bit symbol, by dividing the pattern into two 4-bit nibbles which are each encoded into a 5-bit symbol. Of the thirty-two possible 5-bit symbols, sixteen are chosen to represent the sixteen possible 4-bit nibbles. Some of the others are used, again in pairs, for the encoding of Command inputs (when these are not all 0's). In 9-bit mode, similarly, the Transmitter uses one 4B/5B encoder and one 5B/6B encoder to code nine Data bits into an 11-bit symbol. Finally, in 10-bit mode, two 5B/6B encoders code ten Data bits into a 12-bit symbol.

The Transmitter further performs a NRZI conversion on all symbols, representing bit "1" by effecting a *transition* (LOW-to-HIGH or HIGH-to-LOW) on the serial link's voltage level, and representing bit "0" by maintaining the current voltage.

The following example illustrates this two-stage encoding process. The first stage is dictated by a 4B/5B code table; for the second stage, we assume that the current voltage level (before arrival of the Data) is LOW, or 0:

Data "01" : 0000 0001

4B/5B "01" : 11110 01001

NRZI–4B/5B "01" : 10100 01110

4.2 VHDL Models for TAXIchip

The VHDL models supplied by NSA [3] describe the Transmitter and Receiver chips in terms of a two-level hierarchy: for each chip, the upper level provides a structural description by refinement into components, whereupon the components themselves are described behaviorally at the lower level. In addition, of course, the TAXIchip set as a whole can be modeled at a third, and highest, level by a structural description in which the Transmitter and Receiver themselves are declared and instantiated as components.

The VHDL models do not describe all valid operational modes of the TAXIchip set. In particular, only "8-bit mode" is implemented, wherein the twelve parallel input bits are interpreted as containing eight bits of Data and four bits of Command.

4.2.1 Transmitter Components

The Transmitter model consists of seven components:

- Control : generates internal control signals
- Clock : generates system clock and internal bit rate
- Handshake : manages asynchronous strobing of parallel input
- Input Latch : accepts 12-bit input pattern
- Encoder Latch : stores 12-bit pattern for encoding
- Encoder : encodes 8 Data bits, or 4 Command bits, into 10-bit symbol
- Shifter : converts 10-bit code to serial output bitstream

4.2.2 Receiver Components

Six components comprise the Receiver model:

- Control : generates internal control signals
- Clock : generates system clock and internal bit rate

- Shifter : converts serial bitstream to NRZI-decoded 10-bit patterns
- Decoder Latch : stores 10-bit pattern for 5B/4B decoding
- Decoder : decodes 10 bits into 8 Data bits or 4 Command bits
- Output Latch : outputs parallel Data, parallel Command, or violation flag

4.3 SDVS VHDL Translator

In SDVS, a proof of correctness of a subject VHDL description with respect to a formal specification is accomplished by *symbolic execution* of a representation of that description in the verification system's internal temporal logic, whose characteristic formulas are called *state deltas*.

Accordingly, SDVS has a *VHDL Translator* which parses the subject VHDL description, submits it to static semantic analysis (type-checking, run-time environment construction), and generates the logical formulas representing each language construct occurring in the VHDL source.

The VHDL Translator itself is a formally specified computer program, being a Common Lisp implementation of a denotational semantics for a portion of VHDL [5]. It has been developed incrementally to handle a sequence of increasingly complex VHDL language subsets, attempting to capture the informal VHDL semantics specified in [6].

4.3.1 Current SDVS VHDL Subset: Stage 3

At the close of FY93, the degree to which VHDL had been incorporated into SDVS defined *Stage 3 VHDL*, summarized below as the list of language features processable by the VHDL Translator [5].

Stage 3 VHDL comprises a relatively powerful *behavioral* subset of VHDL. That is to say, Stage 3 VHDL descriptions are restricted to the specification of hardware behavior or data flow, rather than hierarchical structure.

- VHDL design files
  - entity declarations, architecture bodies
  - *restriction*: unique entity and architecture per file
- package `STANDARD`
  - predefined types:
    `BOOLEAN`, `BIT`, `UNIVERSAL_INTEGER`, `INTEGER`, `TIME`, `CHARACTER`, `REAL`, `STRING`, `BIT_VECTOR`

  - various units of type TIME: FS, PS, NS, US, MS, SEC, MIN, HR
  - *restriction*: the implementation of type REAL is preliminary
- user-defined packages
  - package declarations
  - package bodies
- USE clauses for accessing packages
  - *restriction*: packages must be used in their entirety
- entity declarations
  - entity header: port declarations
  - entity declarative part: other declarations
- architecture bodies
- object declarations
  - CONSTANT, VARIABLE, SIGNAL
  - octal and hexadecimal representations of bitstrings
- array type declarations
  - arrays of arbitrary element type
  - bidirectional arrays, unconstrained arrays
- user-defined enumeration types
- subtypes of scalar types
- integer type definitions
- type conversion
- signals of arbitrary types
- subprograms
  - procedures and functions: declarations and bodies
  - *restriction*: excluding parameters of object class SIGNAL
- concurrent statements
  - PROCESS statements
  - conditional signal assignments
  - selected signal assignments

- sequential statements
  - null statement: `NULL`
  - variable assignments (scalar & composite)
  - signal assignments (scalar & composite, inertial or `TRANSPORT` delay)
  - conditionals: `IF`, `CASE`
  - loops: `LOOP`, `WHILE`, `FOR`
  - loop exits: `EXIT`
  - subprogram calls
  - subprogram return: `RETURN`
  - process suspension: `WAIT`
- operators
  - numeric unary operators: `ABS`, `+`, `-`
  - numeric binary operators: `+`, `-`, `*`, `/`, `**` (exponentiation), `MOD` (modulus), `REM` (remainder)
  - boolean and bit operators: `NOT`, `AND`, `NAND`, `OR`, `NOR`, `XOR`
  - relational operators: `=`, `/=`, `<`, `<=`, `>`, and `>=`
  - array concatenation operator: `&`
  - *restriction*: `=`, `/=`, and `&` are the only Stage 3 VHDL operators defined for composite types (i.e., `BIT_VECTOR` and user-defined array types).

4.3.2 Anticipated SDVS VHDL Enhancements: Stage 4

Although Stage 3 VHDL encompasses most aspects of the TAXIchip VHDL models constituting the SDVS/VHDL Application, it falls short of complete coverage in several important respects.

Thus, further enhancements to the VHDL Translator will be required in order for the Application models to be symbolically executable in SDVS. These additional language features will serve to define *Stage 4 VHDL*, and will consist primarily of the following:

- constructs for structural description
  - component declarations
  - component instantiation statements
  - generics
  - configuration specifications/declarations

- type STD_ULOGIC from package IEEE.STD_LOGIC_1164
  - and type STD_ULOGIC_VECTOR (an unconstrained array type)

The necessity of structural constructs is evident, given the multilevel hierarchy of the SDVS/VHDL Application (Section 4.2). The incorporation into SDVS of component declarations and instantiations and of configuration declarations is being guided by a strategy devised in an Aerospace Sponsored Research study [7].

Generics provide a means for an instantiating (parent) component to pass environment values to an instantiated (child) subcomponent, and are typically used in the TAXIchip VHDL models to parameterize timing information.

The enumerated type STD_ULOGIC, defined in IEEE Standard 1164, provides a nine-state logical value system, of which only four states — 0 ("low"), 1 ("high"), X ("unknown"), and Z ("high-impedance") — are actually used in the TAXIchip models.

Other candidates for possible implementation in Stage 4 VHDL, though considerably less essential to the Application models, include:

- certain predefined signal attributes:
  - 'EVENT attribute (function kind)
  - 'STABLE attribute (signal kind)
- OTHERS choice in element associations

The predefined signal attributes 'EVENT and 'STABLE serve, respectively, to indicate whether an event occurred on the referenced signal during the current simulation cycle, and to determine the activity level of the referenced signal during a specified period of time. However, the actual usage of these attributes in the TAXIchip VHDL models appears redundant — and consequently expendable — in every case.

Finally, the Transmitter model employs the OTHERS choice for array initializations. However, its actual mode of use appears uniformly equivalent to initialization by means of string literals — a capability present in the Stage 3 VHDL subset.

5 Expected Results

The SDVS/VHDL Application task is projected to continue for the duration of FY94. It will consist of investigating the feasibility of applying SDVS to the formal verification of the TAXIchip system VHDL descriptions and, where feasible and as time permits, initiating the development of the correctness proofs themselves. This effort subsumes the writing, review, and editing of the deliverables (Section 5.2 below).

5.1 Projected Accomplishments

Originally, we had intended for the SDVS/VHDL Application task to comprise a full-scale hardware verification effort, encompassing: (i) SDVS VHDL translator enhancements, (ii) formal specifications of VHDL subject code, and (iii) development of formal correctness proofs using SDVS. However, funding constraints have necessitated that the task be descoped.

Thus, we currently project four principal accomplishments:

1. Definition of Stage 4 VHDL, a more expressive VHDL language subset, which is to include those features necessary to support the formal verification of the TAXIchip VHDL models.

2. Implementation of SDVS VHDL Translator enhancements to enable the representation of Stage 4 VHDL hardware descriptions in the language of the state delta logic.

3. Formulation of state delta specifications for the TAXIchip VHDL models. These are anticipated to encompass several levels:

   - component specifications
   - intermediate, chip-level specifications (Transmitter and Receiver)
   - high-level, front-to-back specifications of the TAXIchip system

   The front-to-back specifications are expected to assert delay properties of the system, and not merely input-output behavior.

4. Analysis of extensions, if any, to the SDVS inference engine (Simplifier domains, proof rules) required to carry out correctness proofs relative to the specifications referred to in item 3.

Time and availability of resources permitting, we will initiate the process of carrying out the formal verification of the TAXIchip models with respect to some of their specifications.

5.2 Deliverables

The deliverables will consist of the following:

1. Report on the definition and detailed state delta semantics of Stage 4 VHDL.

2. Sections of the SDVS Verification Project Quarterly Progress Reports devoted to the SDVS/VHDL Application task.

3. Section of the SDVS FY94 Final Report giving a detailed account of the SDVS/VHDL Application task, including accomplishments, problems encountered, and the further research and development agenda that the task defines.

4. Briefing to NSA on results of the SDVS/VHDL Application task.

5.3 Milestones

The effort will be accomplished in ten stages:

1. Identification of the Application hardware descriptions (October 1993).

2. Examination of the VHDL code and documentation, and analysis of the limitations of SDVS' Stage 3 VHDL capability with respect to performing the verification (November - December 1993).

3. Modification of the VHDL code to conform to the features projected for Stage 4 VHDL (November - December 1993).

4. Implementation of Stage 4 VHDL in the SDVS VHDL Translator (January - July 1994).

5. Development of state delta specifications of properties of the Application hardware descriptions (March - June 1994).

6. Study of SDVS inference engine extensions potentially entailed by the specifications (May - July 1994).

7. Preparation of the report on the definition and detailed state delta semantics of Stage 4 VHDL, as specified in Section 5.2 (July - August 1994).

8. Preparation of the SDVS/VHDL Application section of the SDVS FY94 Final Report, as specified in Section 5.2 (July - August 1994).

9. Review, editing, and approval of the deliverables indicated in items 7 and 8 (September 1994).

10. Briefing to NSA on results of the SDVS/VHDL Application task (September/October 1994).

6 Program Management

6.1 Resources

We plan for a 1.5 MTS level of effort for FY94 on the SDVS/VHDL Application task. Aerospace will provide, as required, system support, reports processing support, and managerial support for the coordination of the effort.

We will investigate the possibility of acquiring commercial VHDL design tools, such as a VHDL Simulator, to bring additional leverage to bear on the task. In a like vein, it was our experience that the FY93 MSX Ada verification effort [8, 9] benefitted greatly from the use of the Verdix Ada compiler. Tools under current consideration include the Vantage Spreadsheet and the Model Technology V-System.

6.2 Coordination

Aerospace will maintain configuration management of the VHDL hardware descriptions for the SDVS/VHDL Application task, requesting code updates and assistance from NSA points-of-contact (Section 6.3.2), as needed.

Most of the communication and file transfers between Aerospace and NSA will be handled via electronic mail and FTP (File Transfer Protocol). These will be supplemented by phone calls, FAX transmissions, and regular U.S. Mail when appropriate.

None of the communications or files will involve classified material.

6.3 Points of Contact

We identify the Aerospace and NSA Points of Contact (POCs) for the SDVS/VHDL Application task, at both the managerial and technical levels.

6.3.1 Aerospace POCs

- Manager, Computer Assurance Section
  - Ranwa Haddad
    phone: (310) 336-5288
    e-mail: haddad@aero.org
- Principal Investigator, SDVS Project
  - Leo Marcus
    phone: (310) 336-4541
    e-mail: marcus@aero.org
- Technical Lead, SDVS/VHDL Application
  - Ivan Filippenko
    phone: (310) 336-1808
    e-mail: ivan@aero.org

6.3.2 NSA POCs

- Chief, Formal Methods Special Projects Office, R2
  - C. Terrence Ireland
    phone: (301) 688-0840
    e-mail: cti@tycho.ncsc.mil
- Technical Lead, TAXIchip VHDL Model Development
  - Steve Lobeck
    phone: (301) 688-0275
    e-mail: salobec@alpha.ncsc.mil

References

[1] B. Levy, I. Filippenko, L. Marcus, and T. Menas, "Using the State Delta Verification System (SDVS) for Hardware Verification," in *Proceedings of the IFIP TC10/WG 10.2 International Conference on Theorem Provers in Circuit Design: Theory, Practice and Experience: Nijmegen, The Netherlands* (ed. V. Stavridou, T. F. Melham, and R. T. Boute), pp. 337–360, North-Holland, June 1992.

[2] I. V. Filippenko, "Some Examples of Verifying Stage 3 VHDL Hardware Descriptions Using SDVS," Technical Report ATR-93(3778)-1, The Aerospace Corporation, September 1993.

[3] S. A. Lobeck, "Am7968 Transmitter/Am7969 Receiver TAXIchip VHDL Models." National Security Agency, December 1993.

[4] Advanced Micro Devices, Inc., *Am7968/Am7969 TAXIchip Integrated Circuits Technical Manual*, October 1992.

[5] I. V. Filippenko, "A Formal Description of the Incremental Translation of Stage 3 VHDL into State Deltas in SDVS," Technical Report ATR-93(3778)-2, The Aerospace Corporation, September 1993.

[6] IEEE, *Standard VHDL Language Reference Manual*, 1988. IEEE Std. 1076-1987.

[7] I. V. Filippenko and L. G. Marcus, "Integrating Structural VHDL Hardware Descriptions into the State Delta Verification System (SDVS)," Technical Report ATR-92(8180)-1, The Aerospace Corporation, September 1992.

[8] T. K. Menas, J. M. Bouler, J. E. Doner, I. V. Filippenko, B. H. Levy, and L. G. Marcus, "Overview of the MSX Verification Experiment using SDVS," Technical Report ATR-93(3778)-6, The Aerospace Corporation, September 1993.

[9] T. K. Menas, J. M. Bouler, and J. E. Doner, "Specifications and Correctness Proofs for Portions of the MSX Ada Software," Technical Report ATR-93(3778)-5, The Aerospace Corporation, September 1993.