1.0 INTRODUCTION

The Equipment Division of Raytheon Company is striving to develop a VHDL based top-down design, bottom-up verification methodology. The goal of this methodology is to support and automate the process of transitioning complex digital systems from top level system specifications to hardware implementation. This paper will discuss the present state of that methodology, the tool suite required to support the methodology, and the successful application of the methodology to a major Department of Defense program. In addition, pathfinding efforts and associated tools for extending the methodology will be described. The first application of this methodology to a major program resulted in a 16 percent cost reduction and 30 percent schedule improvement over previous approaches. The program is a space-based, signal and data processing system which required eleven modules (8"x 8" double sided, surface mount), eight ASICs (10K - 60K gates), seventeen FPGAs, off the shelf complex components, and off the shelf MSI and SSI devices. This accomplishment was largely due to the initial efforts that were invested to pathfind and evaluate CAE tool options and the development of a concrete design methodology for all facets of system design.

1.1 Program Design Challenges

Although not a formal requirement for this program, Raytheon feels that it is only a matter of time before VHDL will be a requirement for specification and implementation of government sponsored digital system designs. Early exposure to the language's capabilities and training of key individuals within the company would be of significant value on future programs.

Historical data from other programs at Raytheon pointed to a major stumbling block to achieving first pass success in the development of complex application specific integrated circuits (ASICs). Design methodologies and simulation strategies developed to date were very successful in implementing and validating an ASIC from a standalone ASIC level specification. However, occasionally an implementation problem occurs because the ASIC level specifications are an interpretation and allocation of system level requirements through the system -> subsystem -> module -> multi-chip module -> ASIC design hierarchy. A methodology and supporting simulation toolset environment is required to validate that the proper functional and
performance requirements have been allocated to each design hierarchy level.

The development of a successful methodology for the above program was complicated by several factors. First, the very high signal processing (>one billion-operations-per-second) throughput requirements, coupled with the low weight, low power and high radiation tolerance requirements, dictated the need for radiation hardened, sub-micron CMOS technology. The desire to implement the system at the lowest cost mandated the need to demonstrate the ability to interface to multiple foundry sources. Second, the selection of radiation hardened, sub-micron, CMOS foundries imposed the reality of a sixteen to twenty week fabrication cycle. With a program demonstration deadline of twelve months, ASIC specification, design, and validation would have to be completed by month seven. Raytheon concluded these challenges could not be met without a top-down design, bottom-up verification, VHDL based, digital system design methodology.

1.2 Raytheon's Top-down VHDL Design Methodology

The VHDL methodology employs a top-down design, bottom-up verification approach. From top level system requirements specifications, a series of functional requirement specifications are generated to define the system at the subsystem, module, and application specific integrated circuit level. A VHDL register transfer level (RTL) model and a behavioral testbench are developed at each specification level. The VHDL testbench is used to exercise the VHDL model at each design level to verify functionality, timing, and traceability of each specification requirement to the design implementation. This design/simulation step verifies that the design definition and allocation process to subsystem, module, and ASIC level have preserved top level requirements. This process can be viewed as the decomposition of system level requirements into both written functional requirements and executable specifications in the form of VHDL register transfer level models. Logic synthesis tools are then used to convert the lower level ASIC RTL executable specifications into a gate level design implementation. The logic synthesis tools are calibrated with the target foundry design library. At the gate level, each ASIC design type is simulated to verify design functionality, detail timing, and fault grading. Once validated at the gate level, each ASIC is exercised and validated at each design level. That is, the ASIC model is inserted in the module level VHDL model and revalidated. The module level VHDL model is, in turn, inserted in the subsystem VHDL model and revalidated. This process forms the bottom-up verification phase of the methodology. The final step, prior to release of the ASICs, is the system verification via ASIC emulation. This step allows final verification of each ASIC design by execution of operating system and application level software. The methodology and associated hardware/software environment is shown in Figure 1.

2.0 A LOOK AT RAYTHEON'S VHDL CAE TOOL SUITE

After extensive evaluation, Raytheon has adopted three major VHDL tool components to meet its aggressive design and performance goals. The front end component to this environment is a VHDL simulator residing on a workstation platform. The middle component is a logic synthesis tool for both generic and target foundry gate level implementation that also resides on a workstation platform. The back end component is a hardware accelerator for both RTL and gate level simulation and verification. In addition, Raytheon has integrated a final level of verification and added confidence for all ASIC designs by introducing rapid prototyping with ASIC emulators. IKOS Voyager Series was chosen for VHDL software simulation and gate level accelerated simulation. For synthesis and ASIC emulation tools, Racal-
Redac’s SilcSyn and Quickturn were chosen respectively.

2.1 Simulation Phase

At the front end, the IKOS Voyager VHDL simulator is used to support all simulation activities. The IKOS Voyager environment provides a seamless simulation link between the workstation platform and the IKOS hardware accelerator platform. That is, the simulator user interface is transparent. The fact that the simulation is executed on the hardware accelerator or the workstation is virtually invisible to the user. Features like source level debugging are preserved for behavioral level and RTL level VHDL simulations. This ability of the workstation/hardware accelerator environment to support mixed level simulation is very important to the top-down design approach. This allows the same behavioral level VHDL testbench to be used to validate the same design at the various levels of abstraction as the design evolves. For example, the same VHDL testbench that validates the RTL level description of an ASIC can be used to validate the gate level design implementation after completion of the logic synthesis step in the methodology.

2.2 Synthesis Phase

In the middle, Racal-Redac’s synthesis tool SilcSyn is used to perform three distinct synthesis steps.

The first step of logic synthesis (architectural synthesis) process is a unique capability of the IKOS/Racal-Redac toolset. Once an RTL level model is generated using the SilcSyn VHDL coding guidelines, the SilcSyn tool can "synthesize" a generic gate level design implementation. This implementation is then mapped to a set of primitive functions that have been optimized for execution on the IKOS hardware accelerator. This process has two major benefits. The first benefit is that this produces a design representation and model that more closely represents that actual hardware implementation. Problems associated with simulation initialization can be addressed early. The second benefit is in execution speed: simulation is 1 - 3 X faster than the workstation, architectural synthesis is 2 - 4 X faster than gate level synthesis. All these factors contribute to a faster development and validation of a design at the RTL level.

The second step in the synthesis phase consists of gate level logic synthesis. From the VHDL RTL ASIC design description, the SilcSyn tool is used to implement the design in the target foundry technology. This issue was compounded on the above program by the requirement to interface to three, radiation hardened, sub-micron CMOS foundries (Harris Semiconductor, Texas Instruments, and LSI Logic). The designs are iterated at this step until functional, performance, and gate count requirements are met.

The final step in synthesis is started after completion of the logic synthesis step. SilcSyn provides an automatic test vector generation (ATG) capability. This allows the insertion of boundary, partial or full scan logic into the design. The ATG software then derives a set of test vectors that yield a very high fault grade (>90%).

2.3 Acceleration Phase

At the back end, the IKOS hardware accelerator box performs the conventional gate-level simulations, fault grading, back annotating of layout effects from ASIC foundries, and full-timing (dynamic) analysis. It also performs VHDL RTL acceleration that maps VHDL descriptions at the RTL level into IKOS primitives. This RTL structure is then simulated on the IKOS accelerator without the loss of user interface functions such as source level debugging of the VHDL code. Board level simulation consisting of a blend of ASIC VHDL models and off the shelf MSI and SSI models can be accomplished on the IKOS accelerator platform as well.
2.4 Emulation Phase

One of the major problems that face any simulation based design methodology is addressing the practical limitation of how to effectively simulate software. Most modern, complex, digital systems derive a significant portion of their functionality from software. To properly validate the functional and performance requirements of a system may require the execution of both operating system and application software. Even with the assistance of behavioral and RTL model abstraction levels for hardware, coupled to hardware acceleration platforms still yields expansions of $10^4$ - $10^8$ X for subsystem and system level simulations. The expansion factor means that execution of tens of seconds of software (range required to validate practical systems) translates to simulation times of days, weeks, or months. This is not a practical solution. Raytheon has addressed this problem by the introduction of rapid prototyping via ASIC emulation.

ASIC emulation not only provides an effective complement to simulation, it provides an effective means to reduce overall system development cycles and greatly reduces the probability of second pass ASIC requirements. The mitigation of the need for second pass silicon has significant impact on potential program cost and schedule growth risk.

2.5 VHDL Development Environment

One major historical lesson learned at Raytheon is the effect of providing adequate hardware and software resources on the success of any development methodology. Too little capability frustrates the user and leads to limited application of the methodology. Too much resources limits the cost effectiveness of applying the methodology. A major investment was made by Raytheon to perform significant benchmarking and pathfinding to determine not only the best software toolset to support the top-down VHDL development methodology, but evaluate the adequate capability for successfully completing the selected program. The VHDL development environment is shown in Figure 2.

The figure depicts the three tier characteristics of the development/simulation environment. The tier approach allows the cost and capabilities of the hardware/software platforms to match the simulation task.

The first tier is the low-end solution. The low-end environment consists of 15-20 MIPS workstations with IKOS Voyager VHDL simulation software installed. This tier supports the portion of the design team writing low complexity VHDL models and testbenches.

The second tier represents the medium range simulation capability. This tier consists of 25-30 MIPS and 35-40 MIPS workstations with IKOS Voyager VHDL simulation installed. In addition, three of the 35-40 MIPS workstations were used as hardware platforms for support of the Racal-Redac SilcSyn logic synthesis software. This tier layer has three purposes. First, it provides the next level of performance for executing complex ASIC or module VHDL software simulations. Second, it serves as the front end to the high performance IKOS hardware accelerators. Third, as mentioned above, several of the 35-40 MIPS workstations serve as hardware hosts to the SilcSyn logic synthesis software.

The third tier provides the simulation capacity required for subsystem level simulations. This layer consists of an IKOS hardware accelerator tightly integrated with a 25-30 MIPS workstation or a 35-40 MIPS workstation. The execution of the IKOS Voyager software on the workstation provides the transparent interface to the IKOS platform. This capability supports the use of the IKOS as a high performance, mixed level simulator that allows VHDL models, ASICs, FPGAs, and off the shelf components to be simulated together. As indicated in section 2.3, the IKOS hardware accelerators are also used as very
Figure 2 - VHDL Development Environment
high performance, gate level accelerators for gate level simulation tasks.

For completeness, the Quickturn ASIC emulation and MENTOR schematic capture/component placement environments are shown as well.

3.0 PROGRAM METHODOLOGY/ TOOLSET DEVELOPMENT APPROACH

In addition to extensive benchmarking and pathfinding to determine the optimum toolset and hardware platform environment, significant work was done to address design and development issues that have plagued previous programs at Raytheon.

3.1 VHDL Development/Training Aids

For the past few years, Raytheon has been actively monitoring the emerging VHDL based tools in the market and surveying VHDL model resources. It has resorted to several methods for keeping itself up to date with the rapid growing market of VHDL related tools. Extra efforts were invested to evaluate tools that are targeted to ease the process of developing VHDL based design such as VHDL generation tools from graphical entry and VHDL model libraries. One of the tools that Raytheon has acquired to be used corporate wide, is the Std_developersKit® by the VHDL Technology Group. The Std_developersKit was extremely helpful in the development of the testbenches. Designers were able to tap a wide selection of memory models, timing function models, and text I/O related utility models, resulting in a considerable savings in both time and labor. This tool is one of the very few to provide a wide range of VHDL behavioral models.

Another tool that was acquired and proved to be a low cost and efficient means of providing VHDL training is the VHDL SelfStart_Init® by Topdown Solutions. Designers who were brought in later in the design cycle spent a period of less than one week to learn VHDL using this tool.

3.2 Team Organization

Due to the large scope of this project, it was necessary to appoint individuals to coordinate the needed activities between the design team and the rest of the world. Figure 3 illustrates how the key personnel were organized in relationship to the design team. The two key individuals were an ASIC/VLSI design coordinator and a system manager. The Topdown Design Solutions consulting group, working with the lead engineers and the VHDL experienced engineers, guided the design team initially in the VHDL development. The design team was partitioned into three major groups, each group consisted of a design subteam and a test subteam. The three groups were: ASIC/FPGA group, module group, and subsystem/system group. It is important to stress that within each group, the subteams were completely separate and worked from the same specification requirements associated with their respective design.

ASIC/FPGA Group

- ASIC designers were responsible for generating the VHDL RTL code and performing the synthesis activities such as optimizing for silicon area or performance, static timing analysis, fanout considerations, etc.

- Testbench developers were responsible for testing the RTL and gate level models. Dynamic timing was also performed on the gate level models.
Module Group

- Module designers were responsible for generating the Mentor schematics for the module and translating it to a VHDL structural netlist via our in-house developed translator. These netlists served two purposes: 1) Validation of module netlist for proper PWB release, but could be at either gate or RTL level when functional verification of the module was performed.

Subsystem/System Group

- Subsystem designers were responsible for architecting major functional blocks of the overall system.
- Testbench developers were responsible to

![Diagram](image)

Figure 3 - Team Organization

and 2) A backbone to interconnect IKOS models for off-the-shelf components with gate level representations of the ASIC/FPGAs. This backbone was essential in verifying the ASIC's integrity in its respective system environment.

- Testbench developers were responsible for testing the module model which incorporated the ASIC models. ASIC models had to be at the gate level when dynamic timing analysis was performed validate the subsystem functionality and timing. Once again, the ASIC models were at the gate level to perform dynamic timing analysis and at either level to perform functional simulation. A subsystem consisted of one to four modules.

ASIC/VLSI Coordinator

It was this ASIC/VLSI coordinator's primary responsibility to ensure that silicon would work on a first pass basis.
Thus, this individual had to have the breadth of experience and overall knowledge necessary in order to act as the central, technical communication link between the lead ASIC designers and the multitude of vendors -- the three ASIC foundries, the EDA vendors, the modeling houses, etc. Furthermore, the coordinator had to function as a design consultant to the ASIC modellers and testbench developers, not only to lend his expertise in solving technical problems, but also to ensure common design practices on all designs.

It was important to relieve designers from the day to day tasks of interfacing to the EDA vendors, allowing them to fully concentrate on developing/debugging their individual models. The reporting and tracking of bugs in the software platforms was also best left to one person to manage. This provided a focal point for both the vendors and the design team. It provided a prioritizing mechanism for the vendors and served as a human bulletin board, allowing designers to see if anyone else experienced the same problem. It was also imperative that one person disseminate the information coming in from the foundries and pass on the appropriate data to the team, and vice-versa. Specifics on clock distribution and buffering schemes, scan design and implementation, and foundry specific data are just some of the examples of this information.

**System Manager**

Due to the size and complexity of the VHDL development environment (described in section 2.5), it was essential to establish a full time system manager. Knowledgeable in the design process and trained in VHDL, the system manager was responsible for coordinating all activities related to the development environment. Primary responsibilities included: maintenance of hardware platforms, tracking and coordinating versions of EDA software, tracking and coordinating of IKOS and VHDL library development, daily backups of design and simulation data bases.

4.0 A DETAILED LOOK AT RAYTHEON'S DEVELOPMENT METHODOLOGY

4.1 ASIC VHDL Model Development and Verification

ASIC VHDL models are created at the RTL level in compliance with the SilcSyn synthesis guidelines using manual text entry. This RTL code is used as a common denominator between synthesis and simulation of all the ASIC designs. Like all synthesis tools, SilcSyn imposes a VHDL coding style that is a subset of the VHDL language constructs. This style is better known as Register Transfer Level (RTL) which can be thought of as an intermediate step between the highly abstract behavioral level and the detailed structural level. In parallel to this effort, VHDL testbenches are generated from the same specification by other designers. The testbenches are modeled at the behavioral level making extensive use of the Std_developers Kit packages, a product of the VHDL Technology Group. These testbenches are used to test the functional requirements of the ASICS as identified in the specifications both before and after the synthesis process. IKOS provided the capability of preserving the VHDL testbenches throughout the ASIC validation phases. As such, these testbenches exercise the ASIC as a VHDL model as well as its equivalent gate netlist. Engineers generating the testbenches for subsystem and ASIC levels are well acquainted with the circuit's behavior and use that knowledge to generate high-quality test vectors.

4.2 ASIC Validation Through RTL Acceleration

RTL accelerated simulation played an important role in the development phase of the ASICS as well as augmenting the speed of module and system level simulation. The principle objective when coding for synthesis is to achieve gate-level designs with optimum performance and minimum gate count. In achieving this goal, several problems surfaced: 1) Different VHDL
representations may result in functionally equivalent yet structurally different designs, and (2) the code may behave differently in gates than in the software simulation world. RTL accelerated simulation offered a unique capability by qualifying the VHDL code for hardware structure before the synthesis process, thus optimizing debug iteration cycles and reducing the amount of gate-level simulation required by identifying problems earlier in the design process. Moreover, it results in significant simulation speed improvement, thereby allowing real-world testing of the ASIC design while the design is still language-based and easy to change. The latter provides the capability to support several minutes of real time system simulation.

4.3 ASIC Detailed Design Through Synthesis

Having produced acceptable code, after validating the ASIC design at the RTL level, the design is then transferred to the SilcSyn synthesis tool for gate level generation. When running the synthesizer, there are many compile options. Timing or area optimization can be chosen, or both. Maximum area and minimum timing requirements can be entered where the synthesizer attempts to meet both requirements, based upon a cost function. Its success in this attempt is dependent upon the target library technology. Furthermore, boundary scan, partial or full scan test insertion options are available using SilcSyn Test Synthesis option. Additionally, static timing analysis is performed at this level as part of the constraint driven timing optimization.

4.4 ASIC Validation Through Gate Level Simulation

Once the gate level netlist is generated, a second round of validation is performed on the IKOS accelerator by simulating the gate level equivalent of the ASIC using the same testbench that validated its RTL equivalent. At this level, dynamic timing analysis and back annotation of foundry layout effects are all performed. Problems resulting from asynchronous logic or post-layout clock distribution can be addressed at this time. In addition, testbenches are instrumental in supplying efficient and concise functional test vectors for foundry release. Since the gates are now targeted toward a particular technology, detail timing issues (i.e., setup and hold times) are also checked.

4.5 Subsystem And Module Simulation And Verification

In parallel to the ASIC development efforts, module/subsystem designs are conducted. Module designs are created through the traditional schematic-capture methods on Mentor. The schematic entry functions as a liaison for proper PWB release and fabrication. The schematics are then translated to VHDL skeleton netlists through an in-house developed utility software tool. These VHDL netlists are used as a backbone to interconnect IKOS models for off the shelf components with the gate level representations of the ASICs. Module/Subsystem simulations and validations are likewise performed on the IKOS accelerator stimulated by VHDL behavioral testbenches. Once all modules or subsystems are modeled and verified with their respective ASICs at the gate level, a system level simulation can be accomplished. System simulation is heavily dependent on the amount of time consumed to simulate thoroughly. One of the main goals to accomplish during the methodology development and tools evaluation was the capability of performing real time system simulation for a complete validation of system design integrity. Simulation at higher level definition such as RTL level offers a significant simulation speed improvement than the lower level gate netlist. At the system level, individual components are brought together as a mixture of RTL and gate level representations. ASICs are kept at the RTL level and the rest of the glue logic such as off the shelf components are at the gate level. By simulating a complete system, designers are effectively performing a virtual system integration. It is the first opportunity for engineers to
bring individual components of a system together and observe subtle design errors caused by misinterpreted interface specifications or errors impossible to find by verifying individual components in isolation. Perhaps more significant, designers verify the system-level operation of the design by executing Operating System and/or application software code.

4.6 ASIC Simulation and Verification By Emulation

Emulation is the converse of a hardware modeler. While the hardware modeler plugs a physical part in a virtual system, the Quickturn tool plugs a virtual device into a physical system. A final level of ASIC validation is performed before release to foundry. Designers can load the optimized ASIC netlist generated by SilcSyn synthesis in the Quickturn tool, which programs FPGAs to emulate the function of the target ASIC. Emulating ASICs can provide a benefit in accelerating the development process as a whole. ASIC emulation allows operating system and application software to be executed on the virtual ASICs significantly increasing system level design confidence. In addition, software integration does not have to wait for ASIC fabrication to complete, reducing system test and integration time.

1) Methodology dictates development of both VHDL executable specification model and testbench at each design level. This ensures that design function and performance requirements are properly flowed down and tested.

2) Methodology dictates bottom-up verification approach. Lowest level design implementations are verified in context of higher level design assemblies. This ensures design implementation compliance.

3) Design implementation via logic synthesis from verified RTL level designs greatly enhances gate level design implementation productivity.

4) Methodology allows rapid prototyping with ASIC emulation. This greatly enhances confidence in ASIC design by allowing execution of operating system and application level software on virtual ASIC implementations. This approach also allows early test and integration of both system diagnostics and operating system and/or application software. This reduces the over-all system test and integration schedule.

5.0 ELEMENTS OF COST/SCHEDULE SAVINGS

Raytheon has successfully applied this methodology to the development of: eleven double sided, 8" x 8", surface mount modules, 277K gates of logic contained in eight ASIC types, 50K gates of logic contained in seventeen FPGAs, and a significant amount of off the shelf MSI and SSI components. The end result was a sixteen percent reduction in cost and a thirty percent reduction in schedule.

The methodology is successful for several major reasons:

6.0 RAYTHEON'S METHODOLOGY EXTENSION PATHFINDING

The methodology and toolset described in this paper makes significant progress towards establishment of a digital system design methodology that ensures first pass design implementation success. To further enhance its cost effectiveness requires the development of additional strategies to reduce software development and system test and integration cycles. Towards this end, Raytheon is currently evaluating the viability of "virtual prototyping" with the Quickturn ASIC emulation systems.

The concept is to use the netlist output from the architectural synthesis step in the subsystem VHDL modelling methodology as the source data for the Quickturn FPGA/Memory/Hardware Modeller design.
implementation as opposed to using the synthesized gate level model, as is currently done. Once validated, the Quickturn virtual prototype could be utilized by the software group as the first target hardware platform. A flow diagram of this process is shown in Figure 4.

If successful, this capability could reduce a programs over-all development cycle by one to two months.

7.0 CONCLUSION.

The Equipment Division of Raytheon has developed and implemented a top-down design, bottom-up verification with application of the methodology, yielded a 16 percent cost and 30 percent schedule improvement over previous approaches.

8.0 ACKNOWLEDGEMENT

The authors wish to thank the program design team and EDA vendor support team. Their efforts made the application of VHDL a success.

Figure 4 - Quickturn "Virtual Prototype" Methodology Extension

methodology using the emerging IEEE 1076 VHDL. The viability of both the methodology and the ability of VHDL to support it, was demonstrated by the successful application of the methodology to a major DoD program's signal and data processing system design development. This accomplishment was made possible by significant initial efforts that were invested in, such as: pathfinding/benchmarking of each major component in the proposed methodology, installation of adequate hardware and software resources to avoid methodology compromises, and the development of formal tool set training and methodology documentation. These factors, combined