

INTEGRATION OF VHDL INTO AN INTERNAL DESIGN ENVIRONMENT

Venkat Immaneni
Intel Corporation
MSCG Architecture and Development Group
5000 W. Chandler Blvd., M/S C5-217
Chandler, AZ 85226.
Phone No: 602-554-8483
Email: vimmanen@sedona.intel.com

ABSTRACT

VHDL is becoming an industry standard and several IC design houses are looking into integrating VHDL into their design environments. Complete integration of an external hardware description language such as VHDL into an internally developed design flow is a complex task. Most design environments use tools and HDLs which have been developed internally over many years. Tools and design flows based on VHDL need to be developed. Capability to transfer design data between the two HDLs (internal HDL and VHDL) should be provided. Also, the internal tool suite should be enhanced to take advantage of the powerful features offered by VHDL. In addition to providing an internal design flow which uses VHDL, a true and complete VHDL integration should enhance the internal design environment by tapping into the vast potential that VHDL has to offer as a language, and by utilizing the powerful VHDL based tools available in the market today. Only after complete integration is achieved will the main benefits of using VHDL begin to surface. This paper describes how VHDL is being integrated into the internal design environment in three phases at Intel's Multimedia and Supercomputing Components Group (MSCG).

Evaluation and Translation Phase: This first phase of VHDL migration addresses the evaluation of VHDL and internal HDL (iHDL) language features, evaluation of various VHDL simulators available in the market and development of translators between the two HDLs.

VHDL Design Flow Development Phase: The next phase concentrates on developing tools and methodologies to support designing in VHDL. The goal is to provide the designers with an environment similar to iHDL. This phase enables the designers to model chips in VHDL and also utilize all the advantages the internal design environment has to offer.

Design Environment Enhancement Phase: The last phase completes the VHDL integration process by enhancing the internal tool environment to take advantage of the wide variety of VHDL tools available in the market and the powerful features offered by the VHDL language.

A three-phase integration of VHDL into Intel's design environment retains the advantages of the internal environment and enhances the environment by taking advantage of the VHDL language features and the powerful tools available in the market. Significant increases in productivity of the designers can only be noticed after the completion of all three phases.

Integration of VHDL into an Internal Design Environment

October 21st, 1992

Venkat Immaneni
Intel Corporation
5000 W. Chandler Blvd, CH5-217
Chandler, AZ 85226
Tel No: 602-554-8483
email: vimmanen@sedona.intel.com



V. Immaneni

Outline

- **Project Goals**
- **Internal Design Environment**
- **Three-Phased Integration of VHDL**
- **Evaluation and Translation Phase**
- **VHDL Design Flow Development Phase**
- **Design Environment Enhancement Phase**
- **Conclusions**



V. Immaneni

Project Goals

To completely integrate VHDL into the internal design environment and derive productivity increases from it

- NOT just using VHDL in the design flow
- Utilize all the powerful features and tools that VHDL has to offer
- Retain the features and advantages of the internal design environment
- Develop new methodologies for productivity gains



V. Immanuel

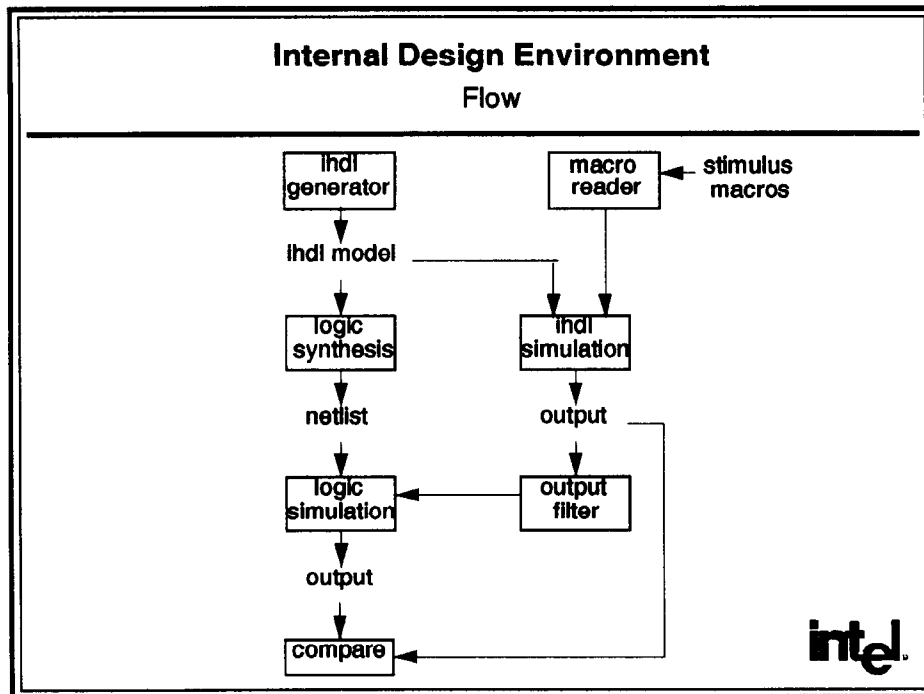
Internal Design Environment

Features

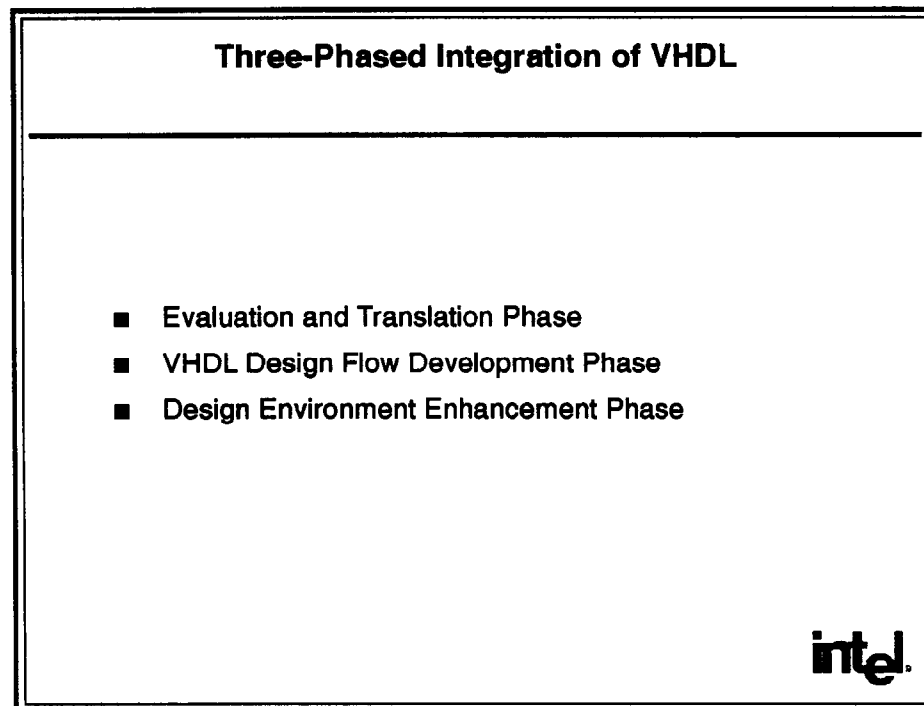
- Tools tailored to meet the needs of internal designs
- Standard products
- Emphasis on high performance and area density
- Internal HDL (iHDL) and internal synthesis tools
- RTL level modeling
- Restrictive design environment (strict design rules)
- Synchronous and race free designs
- Static timing analysis
- Device level design for high performance circuits



V. Immanuel



V. Immanent



V. Immanent

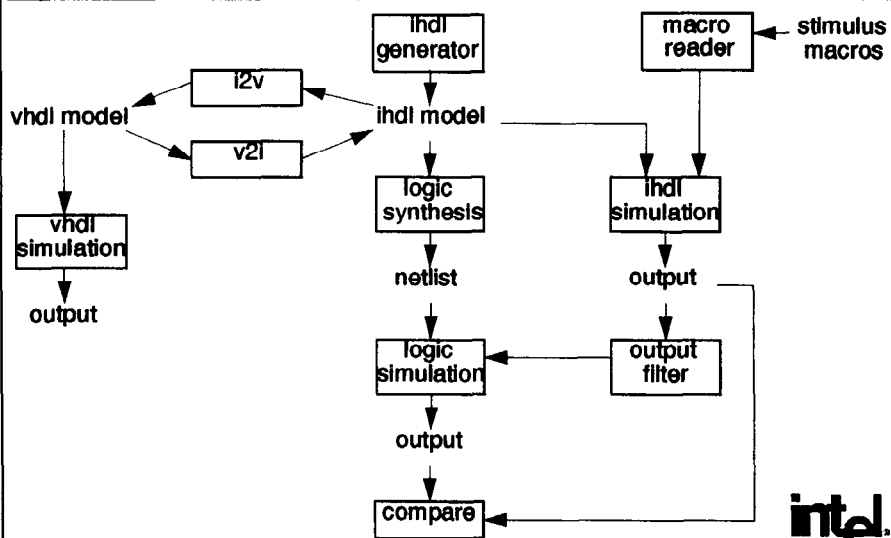
Evaluation and Translation Phase

- Comparison of languages - iHDL vs VHDL (typically most HDL are subsets of VHDL)
- Evaluation of VHDL simulators in the market (selection criteria should be defined)
- Development of translators between the two HDLs (i2v and v2i translators)

intel.

V. Immanuel

Design Flow After the First Phase



V. Immanuel

At the Completion of the Evaluation and Translation Phase...

- VHDL language is evaluated and understood
- VHDL simulator has been selected to meet the requirements
- Models can be translated between the two HDLs
- External designs can be brought into the internal design environment
- VHDL models of components can be provided to external customers/partners
- Designs still cannot start in VHDL and two HDLs need to be supported



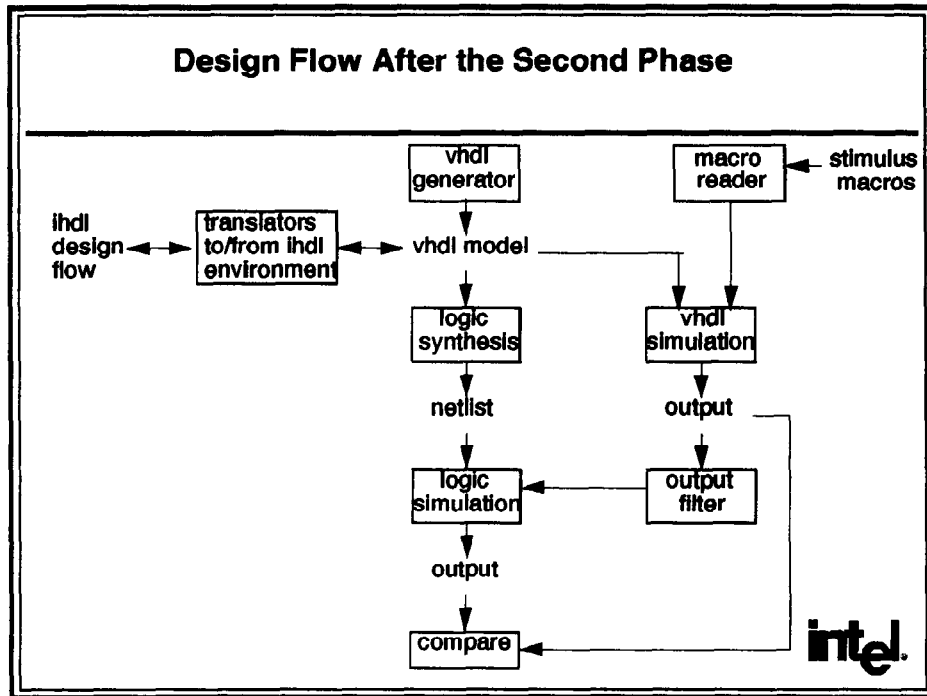
V. Immanent

VHDL Design Flow Development Phase

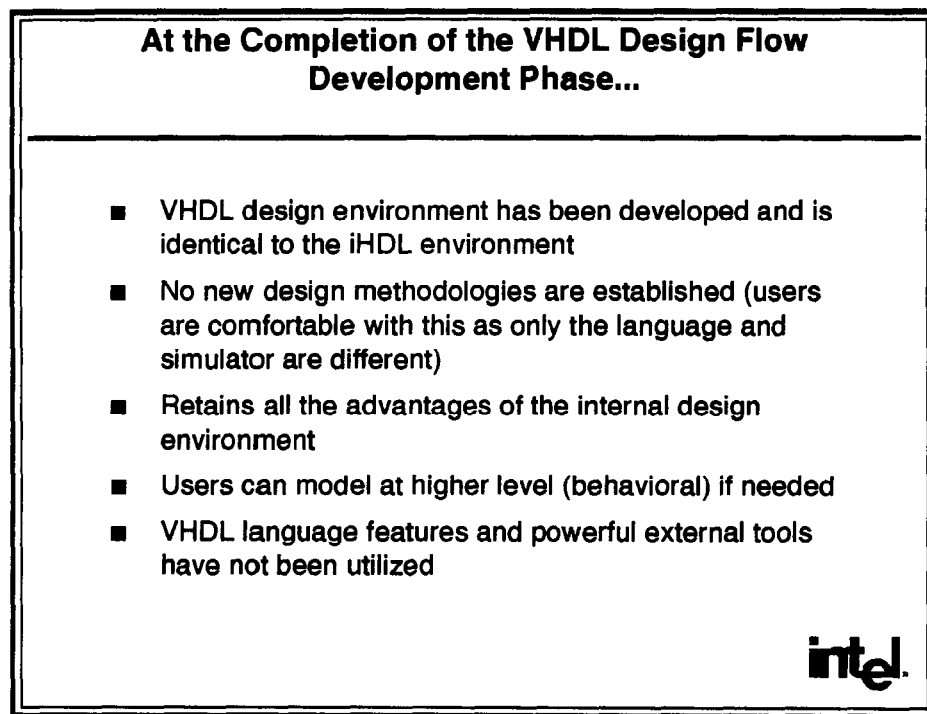
- To provide a design environment starting with VHDL and similar to the iHDL environment
- Developed VHDL generation tool similar to the one in iHDL environment (reads the same input format)
- Developed a stimulus macro front end reader for VHDL
- Provided a bridge to internal synthesis tool from VHDL
- Implemented a stimulus filter from VHDL simulator to the logic simulator/hardware accelerator
- Developed simulation output comparison tool (between VHDL and logic simulator/hardware accelerator)



V. Immanent



V. Immanent



V. Immanent

Design Environment Enhancement Phase

- To enhance the VHDL design environment by using the language features and powerful external tools
- Develop top-down methodology from a architectural/behavioral VHDL description
- Establish guidelines to incorporate timing into behavioral and RTL models
- Develop methodologies, tools and libraries for dynamic timing simulation at behavioral, RTL and logic level
- Establish back-annotation guidelines and tools to feed timing data back to VHDL models



V. Immanuel

Design Environment Enhancement Phase (cont...)

- Integrate VHDL generation tools (from state machine descriptions/diagrams, block diagrams, etc)
- Develop/ integrate syntax checkers, rule checkers, library management tools, etc
- Develop test program generation tools for testers from VHDL test benches
- Develop/ integrate formal verification tools for VHDL
- Integrate VHDL hardware accelerator interface for mixed-level simulation
- Integrate the VHDL Instruction Processor



V. Immanuel

At the Completion of the Design Environment Enhancement Phase...

- VHDL will be completely integrated!
- VHDL language features will be fully utilized
- Useful VHDL tools in the market will be integrated
- Several tools will be developed internally
- Major productivity gains are expected



V. Immanent

Conclusions

- Completely integrating external HDL (VHDL) into an internal design environment is a major task
- Need to complete several evaluations (tools and languages), development of translators and tools, integration of external and internally developed tools and development design methodologies and guidelines
- A three-phased and a smooth migration path has been outlined and is under implementation (the last phase has started)
- Increases in productivity are already surfacing!



V. Immanent