Douglas E. Ott and Thomas J. Wilderotter,
ITT Avionics,
100 Kingsland Road,
Clifton, New Jersey 07014
Phone (201) 284-2314
FAX (201) 284-2908

Abstract - VHDL International Users' Forum, Fall 1992

Title : Practical Experience with High Level Synthesis of Military ASICs

ITT Avionics has adopted an ASIC design methodology based on high level RTL VHDL synthesis in order to improve the development of today's large complex ASICs. This presentation will compare ITT's VHDL based design flow to more conventional techniques and will discuss the experiences of bringing this process on line, including training, design tools, technical and management problems, and cost and schedule benefits.

Significant improvements in the ASIC development process have already been made due to the use of VHDL and logic synthesis. These include both reduced schedule and cost as well as improved design debug and simulation at the behavioral level very early in the design cycle. Drawing of schematics has been essentially eliminated due to synthesis, and implementing design changes is much more efficient due to VHDL RTL simulation. These benefits have not come without problems however, and a significant amount of work had to be done up front in order to implement this new technique. The major problems faced when moving to a VHDL synthesis design methodology result from the fact that it is such a radically different way to design when compared with traditional schematic-based approaches. Within an organization, key technical leaders must learn the new technology, apply it to the company's system design requirements, and evaluate the complex and expensive simulation and synthesis tools needed for ASIC development. Development of a design methodology involves understanding the VHDL synthesis process thoroughly so that VHDL coding style and synthesis efficiency can be assessed and so that concise design guidelines can be developed for use by digital design engineers. Training and building confidence on the part of the users becomes an important factor in the establishment of a new technology. Fears about getting started with new technology can be overcome with good training and hands on design support in the early stages, and as an experienced user base is established the uncertainty is greatly reduced. In the management area, confidence in the process as well as an understanding of the cost and schedule factors involved is important so that future planning can be done accurately.

Practical Experience with VHDL Synthesis of Military ASICs

October 20, 1992

Doug Ott, Tom Wilderotter

ITT Avionics
100 Kingsland Rd,
Clifton, New Jersey 07014

(201) 284-2314, 5092

ASIC Development at ITT Avionics using VHDL Synthesis

- Typically design 10 new CMOS ASICs/year
  - Today's average : 25k-30k gates, max of 40k
  - LSI Logic 100k Series (.7 um)
  - Real time signal processors, ECM systems
- Seven arrays completed or near completion using VHDL-based logic synthesis
  - 3 arrays of 20k-23k gates, others 10k-15k
- ASICs are being designed, debugged, and reviewed more easily
  - Synthesized gate counts very close to expected results
  - Able to easily control speed vs. gate count parameters
- Detailed design schedule cut by 50% (eg, 4 months to 2 for 20k gates)

ITT VHDL/Synthesis Tools

- Tools used :
  - VHDL Simulation
    - RACAL/REDAC VHDL 2000
    - Synopsys
    - Model Technology PC version (Learning, small designs)
  - Synthesis
    - SILCSYN (RACAL/REDAC)
  - ASIC Verification
    - LSI Logic MDE, TestBuilder
  - Gate and system simulation
    - ZYCAD accelerators (Logic and Fault simulation)

Why use VHDL and Synthesis?

- ASIC complexity continues to grow rapidly
  - 50k gates common, 500k possible (today ...!)
- ASIC development schedules and cost growing

- VHDL and RTL level synthesis help manage the complexity while reducing schedule and cost
  - Schematic generation can be mostly eliminated
    - Recent 25k ASIC designs have needed 250 schematic pages!
    - Logic synthesis produces the "schematics" and net lists
  - Designs are simulated behaviorally early in the design cycle
    - ie, Find and correct design problems before "schematics" are done

What are the problems in moving to a VHDL methodology?

- With VHDL and logic synthesis, ASIC design is done using a language rather than schematics
  - A radically different concept for both design engineers as well as management

Issues to deal with :

- Learning of both VHDL and synthesis
- Evaluation of software tools (simulators, synthesizers)
- Developing confidence in the synthesis process
- Development of methodology and design guidelines
- Training and confidence building for digital design engineers
- Learning how to estimate schedule and cost factors

Making the transition from schematics to VHDL synthesis

- Development activities done by engineers experienced in languages, logic design, and simulation :

Getting started
- Learn VHDL and how to use it to model systems behaviorally
- Evaluate and purchase simulation tools, then synthesis tools

Beginning VHDL synthesis
- Initial phase : Does synthesis work?

- Ongoing effort : Developing faith and experience
  - How well does synthesis work?
  - How much does VHDL style affect results?
  - Identifying and correcting problems and bugs
  - Integration with other tools

Making the transition from schematics to VHDL synthesis

- "Pioneers" must lead the way to establish methodology :
  - What can be synthesized? What can't?
  - How do you translate functional requirements into a synthesizable design?
  - How do you best code VHDL to synthesize a desired function?
- New technologies : overcoming the fear factor
  - Which engineers can handle these tasks?
  - Training
- Dispelling the misconceptions
  - "Just code it in VHDL and the synthesizer will produce the gates!"

Synthesizable logic functions

- Sequential : Counters (Clock generators, timeouts, event counters)
  Gate generators
  Timing waveforms (memory, data bus control)
  Registers, files
  Shift Registers
  Sequencers, state machines
- Combinational
  Decoders
  Multiplexers
  PLAs
  Comparators
  Adders, subtractors, ALUs

VHDL Synthesis Issues

- All of these logic functions can be synthesized but ...
- VHDL and synthesis are not magic!
  - You still must understand logic design in order to use these tools
  - There are many ways to code a design in VHDL, resulting in very different circuits
- Good design guidelines, examples are key to success
- Designers should focus on functions, not gates
- Conversely, be aware of the synthesized gate structure

How do you begin a new design?

- Draw top level block diagram showing I/O, major functions and their relationship
- Identify entities, usually functionally independent logic blocks
- Break down functions into generalized hardware related functions
  - ie, not UART but serial register, clock divider, start bit detector, function decoder
- Identify desired function of each piece and assign to process or concurrent operation

VHDL Synthesis Training

- Overcoming the fear factor
  - Technical overview, lots of examples
  - Initial hands on work with close support
  - Training sessions, textbook reinforcement
- VHDL and synthesis will not eliminate the need for design expertise