ABSTRACT FOR VHDL INTERNATIONAL USER'S FORUM

JEFFREY R. FOX
VICE PRESIDENT OF ENGINEERING
RACAL-REDAC, INC.
238 LITTLETON RD
WESTFORD, MA 01886
(508)-692-4900

VHDL Based Synthesis for ASIC and System Design

This presentation will provide an overview of synthesis technology, from combinational optimization designed for optimizing a detailed gate-level logic design for timing and area, to High-level synthesis including resource sharing, system partitioning and scheduling. The presentation will also distinguish the three domains of representation -- behavioral, structural, and physical -- and discuss the tools that are applicable to each domain at each level of abstraction.

A section will be devoted to the applicability of VHDL for the various levels of synthesis, by first exploring the semantic design of the language, and how it may need to be interpreted and adapted for synthesis purposes.

The presentation will be aimed at the practicing ASIC designer. It will describe typical ASIC design styles, and explain how the various classes of synthesis technology can be profitably employed by each.
VHDL-Based Synthesis for ASIC and System Design

Jeffrey R. Fox

RACAL-REDA
238 Littleton Road
Westford, MA 01886
508-692-4900

Motivation for Using VHDL

"Very Important" Reasons for Using VHDL

- Design at higher level of abstraction (53%)
- Transferability from vendor to vendor (50%)
- Faster design cycles (46%)
- Standard language for synthesis, simulation (44%)
- Engineering management plays an important role in the decision to use VHDL tools.

30% currently use a different IIDL.

Within two years, 48 of the 52 sites will use synthesis.

Biggest issues in vendor selection:
- Customer support
- Pricing

Mixed-Level Design

Structure

Behavior

RAM

Macro Modules

Test Controller

Synthesis Challenges

VHDL designed with event-driven, simulation semantics.

- Events require zero width pulses
- No inherent idea of clocks or resets
- Supports arbitrary delays
- Supports general recursion
- Supports file objects
- Abstract data types present representation problems
**Architectural Synthesis**

Tasks:
- Resource allocation
- Resource sharing, including communications costs

Constraints:
- Timing constraints:
  - propagation delays
- Resource constraints:
  - available resource requirements
  - required and prohibited resource sharings
Benefits of Architectural Synthesis

- More natural style of design entry
- Design space exploration on a higher level of abstraction
- Timing constraints on a higher level of abstraction
- Block diagram generation
- Automatic resource allocation and sharing

RTL & Architectural Style VHDL Descriptions

- Architectural Level
  - Implicit State Machine Style
    - 1 Process
    - 22 Lines
- RTL Level
  - Incremental State Machine Style
    - 2 Processes
    - 49 Lines
RTL VHDL Description

```vhdl
case current_state is
  when state1 =>
    temp_count <= count1;
    temp_val <= incr;
    if (count_enable = '1') then
      next_count1 <= temp_count + temp_val;
    else
      next_state <= state2;
    end if;
    else
      next_state <= state1;
    end if;
  when state2 =>
    temp_count <= count1;
    temp_val <= incr;
    if (temp_count < temp_limit) then
      next_count1 <= temp_count + temp_val;
    else
      next_state <= state2;
    end if;
    next_state <= state1;
  when others =>
    next_state <= state1;
end case;
```

Architectural Level

```vhdl
do_cnt : process
begin
  wait until clk;
  while count_enable loop
    while count1_reg < limit1 loop
      count1_reg <= count1_reg + incr;
      wait until clk;
    end loop;

    while count2_reg < limit2 loop
      count2_reg <= count2_reg + inc2;
      wait until clk;
    end loop;
  end loop;
end process;
```

RACAL-REDAC
High-Level Synthesis

Abstract Behavioral Specification → Register-Transfer Level Structure

Scheduling Resource Allocation (including resource sharing)

High-Level Synthesis Definitions

**Scheduling** - The task of assigning program operations to control steps given certain limits on the available hardware resources and speed requirements

**Allocation** - The task of assigning program operations to hardware resources, given certain time requirements and limits on the available resources
Why Hasn't
High-Level Synthesis Caught On?

1. No logic optimization
2. High-Level systems were technology independent
3. Allocation and scheduling poorly integrated
4. Real costs and timing constraints not considered
5. User control wasn't required
6. No standard language for partial constraints

Conclusion

- VHDL synthesis is here today
- Mixed-level capability required of synthesis tools
- Microarchitectural level is highest practical level of abstraction
- Active research and multi-vendor support makes VHDL the obvious winner for future ASIC and system design