110062

N95-16477

Hardware Description Languages

Jerry H. Tucker

Hardware description languages are special purpose programming languages. They are primarily used to specify the behavior of digital systems, and are rapidly replacing traditional digital system design techniques. This is because they allow the designer to concentrate on how the system should operate rather than on implementation details. Hardware description languages allow a digital system to be described with a wide range of abstraction, and they support top down design techniques. A key feature of any hardware description language environment is its ability to simulate the modeled system.

The two most important hardware description languages are Verilog and VHDL. Verilog has been the dominant language for the design of application specific integrated circuits (ASIC's); however, VHDL is rapidly gaining in popularity. VHDL was developed for the DOD and then transferred to the IEEE in 1986. The language is defined by IEEE standard 1076. Since 1988 the DOD has required all of its digital ASIC's to be supplied with VHDL descriptions.

By describing a digital system in VHDL at a behavioral level, the effect of different architectural decisions can be simulated and evaluated early in the design process. Once an architecture has been selected the various circuits in that architecture can be described using a restricted subset of VHDL. It is then possible to synthesize that VHDL description to obtain the actual implementation of the circuit.

Hardware Description Languages

by

Jerry H. Tucker

Presented at the Workshop on
The Role of Computers in LaRC R&D
June 15-16, 1994

Questions Addressed

- What are HDL's?
- Why use HDL's?
- What HDL's are available?
- How do HDL's differ from other languages?

What are HDL's?

- A special purpose programming language.
- Primarily for specifying behavior and structure of digital systems.
  - Replaces traditional digital design techniques.
  - Supports wide range of system abstraction.
  - Supports top down design.
- Running the HDL program simulates the modeled system.

Why use HDL's?

- Old design methods are inadequate to satisfy demands on digital systems.
  - Increasing complexity.
  - Decreasing development time.
- Automates design process.
  - Requires digital hardware designers to also be programmers.
- HDL synthesized to implement design.

Types of HDL's

- Two dominant HDL's.
- Verilog
- VHDL
- A key component of both is the simulator.

Verilog

- Developed 1983-1984.
- Originally proprietary.
- Now IEEE 1364.
- Dominant language for ASIC's.
  - More "real" designs in Verilog.
- Inherently faster simulation than VHDL.

VHDL

- DOD required common HDL to support designs from different vendors.
- DOD contract awarded in 1983.
- Strong Ada influence.
- Public released 1985.

VHDL (cont.)

- Transferred to IEEE in 1986.
- IEEE standard 1076 in 1987.
- Revised standard IEEE 1076-1993.
- Since 1988 DOD requires all its digital ASIC's to be supplied with VHDL descriptions.

VHDL (cont.)

- VHDL more verbose than Verilog
- Example in VHDL

```
IF ((clk'EVENT) and (clk='1') and
  (clk'LAST_VALUE='0')) then ...
```

- Example in Verilog

```
@(posedge(clk)) ...
```

VHDL (cont.)

- VHDL more flexible than Verilog.
- Momentum seems to be with VHDL.

Levels of Design

- Behavioral
  - Highest level, Most general.
- Register Transfer Level (RTL)
  - Defines registers, counters, I/0 buffers etc.
  - Can be synthesized to specific devices.
- Gate Level
  - Defines design in terms of logic primitives.

VHDL Example mod 3 counter

```
-- MOD 3 counter VHDL example for
-- The role of computers in LaRC R&D
-- workshop June 15-16, 1994.
use work.all;
entity CNT is
  port(CLK: in BIT; Q1, Q0: out BIT);
end CNT;
```

mod 3 counter (cont.)

```
architecture BEHAVIOR of CNT is
begin
  CNT3: process(CLK)
    variable COUNT: INTEGER := 0;
  begin
    if CLK = '1' then
      COUNT := COUNT + 1;
```

mod 3 counter (cont.)

```
if (COUNT > 3) then
        COUNT := 0;
      end if;
      Q0 <= bit'val(COUNT mod 2) after 10 ns;
      Q1 <= bit'val(COUNT/2) after 10 ns;
    end if;
  end process CNT3;
end BEHAVIOR;
```

VHDL example test bench

```
-- Test bench for MOD 3 counter VHDL example for
-- The role of computers in LaRC R&D
-- workshop June 15-16, 1994.
use work.all;

entity TB is end TB;
```

test bench (cont.)

```
architecture TEST of TB is
-- Signal declaration.
  signal CLOCK, Q1, Q0: BIT;
-- Component declaration.
  component CNT
    port(CLK: in BIT; Q1, Q0: out BIT);
  end component;
```

test bench (cont.)

```
for U1: CNT use entity work.CNT(BEHAVIOR);
begin
-- component instantiation statement.
  U1: CNT port map(CLOCK, Q1, Q0);

  CLOCK <= not CLOCK after 50 ns;
end test;
```

Simulation of example