
ASIC Design Using VHDL

Djahida Smati
Clive Charlwood
Shir-Shen Chang
Synopsys Inc.
700 East Middlefield Road
Mountain View, CA 94043 -4033
(415)962-5000
smati@synopsys.com
crc@synopsys.com
chang@synopsys.com

Abstract

The VHDL language has been used for a wide range of models. One modeling style used by ASIC designers is to create a structural network of ASIC cells. Each of the cells represents a logic cell in a target ASIC vendor library. The network is usually created either manually or using logic synthesis.

Although VHDL supports this style, there are several issues that need to be addressed to make this modeling strategy practical.

In particular, VHDL currently offers limited and cumbersome support for backannotation of pin-to-pin delays. One solution is to use a delay calculator outside the VHDL environment and back-annotate the timing into the simulation model. The standard delay format (SDF) has been proposed as a standard mechanism to pass timing values to the simulation model.

This paper presents a practical approach for modeling ASIC libraries and supporting timing back-annotation in both synthesis and simulation tools. This approach includes the use of:

- The standard logic value system IEEE-1164.
- The standard delay format SDF.
- A basic simulation package where a set of common lookup tables and simulation routines for scheduling, selecting delays, and processing timing is defined.
- An entity declaration model which defines a port and generic parameter interface. The generics hold wire delays, rising, falling, setup, hold, and other timing constraints. The generics are used to backannotate the pre- and post-layout interconnect delays. The generic parameters comply with the naming convention of EIA 567.C standards.
- A method to backannotate to models in hardware accelerators is presented, since complicated large structural designs with full timing accuracy requirements are often simulated in hardware for faster simulation speed, and greater gate capacity.

ASIC Design Using VHDL

October 20, 1992

Djahida Smati
Clive Charlwood
Shir-Shen Chang

Synopsys Inc.
700 East Middlefield Road
Mountain View, CA 94043-4033
(415)962-5000
smati@synopsys.com
crc@synopsys.com
chang@synopsys.com

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

Presentation Goals

Describe a practical mechanism to:

- model VHDL ASIC libraries containing pin-to-pin timing.
- backannotate timing values.

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

Introduction

ISSUES

- No standard VHDL modeling style used to develop ASIC libraries.
- VHDL does not currently support pin-to-pin timing.
- VHDL does not directly support min/typ/max timing.

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

Introduction - Cont

A practical solution is described below. Based on

- the use of standards.
 - The IEEE logic value system `std_logic`, IEEE-1164
 - OVI's Standard Delay Format (SDF)
 - EIAs 567.C standard
- a VHDL simulation package to support ASIC libraries.
- calculation of timing information outside the simulation environment.
- a VHDL generic timing model in the ASIC library.
- backannotation of timing values into the instanced ASIC cells.

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

IEEE Standard Logic Value-1164

- A multi-value logic system including 9 states: U, X, 0, 1, Z, W, L, H, -.
- The `std_logic` package also includes type conversion functions and resolution functions.
- `std_logic 1164` is sufficient and adequate to describe most, if not all, ASIC libraries.

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

Standard Delay Format (SDF)

- Open Verilog International Standard.
- Specify delays including pin-to-pin delays, interconnect delays, and timing checks.
- SDF can be generated and consumed by different tools such as synthesis tools, timing analyzers, simulators, delay calculators, and layout tools.

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

Example of an SDF File

```
(DELAYFILE
(SDFVERSION "1.1")
(DESIGN "test")
...
(CELLHEADE
  (CELLTYPE "BE")
  (INSTANCE E/B/UUT)
  (DELAY
    (ABSOLUTE
      (DELAY
        (IOPATH R D
          (0.1:0.2:0.3) (0.2:0.3:0.4)
        )
      )
    )
  )
)
(TIMINGCHECK
  (PERIOD (negedge T) (0.1:0.2:0.3))
  (WIDTH (posedge T) (0.1:0.2:0.3))
)
)
)
```

Header

Pin-to-pin
Timing

TimingChecks

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

VHDL Package Used to Model ASIC Libraries

Based on routines and lookup tables defined in the `std_logic` IEEE-1164 system package.

Used to:

- schedule pin-to-pin timing events.
- select appropriate pin-to-pin delays.
- model checker mechanism for timing hazards and forbidden timing events.

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

VHDL Package for ASIC Libraries Delay Handling

Handling Propagation Delay

- **Determine the delay of a signal transition that does not involve a high impedance state.**
- **Accept as inputs: the current value of a signal, the next value of the signal, and the rise and fall times.**
- **Return the appropriate propagation delay time for the output transition.**
- **For high impedance, accept as inputs: the current value of a signal, the next value of the signal, and the enable and disable rise and fall times.**

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

VHDL Package for ASIC Libraries Hazard/Glitch Handling

Hazard Handling

- **if a circuit hazard caused by an input change is detected, schedule an event to reflect the propagation delay of the input change.**

Glitch Handling

- **Once a hazard is detected, schedule an uncertain output X immediately upon the arrival of a second input change.**

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

VHDL ASIC Cell Model

- Entity declaration defines ports and generics.
- Generics are used to backannotate timing information such as rise, fall, setup, and hold.
- Generics follow the naming convention of the EIA 567.C standards.

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

Example of a VHDL Cell

```
entity AND2 is
generic (
    tpdA_O_R      : TIME := 0.417 ns; -- Timing generics:
    tpdA_O_F      : TIME := 0.557 ns; -- for example
    tpdB_O_R      : TIME := 0.417 ns; -- tpdA_O_R:
    tpdB_O_F      : TIME := 0.557 ns; -- Rise time for
    twdA_R        : TIME := 0.337 ns; -- propagation from
    twdA_F        : TIME := 0.337 ns; -- input A to output O
    twdB_R        : TIME := 0.337 ns;
    twdB_F        : TIME := 0.337 ns);
port (
    A      : IN    std_ulogic;
    B      : IN    std_ulogic;
    O      : OUT   std_ulogic);
end AND2;
```

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

Backannotation Using Hardware Accelerators

ISSUE:

Some hardware accelerators do not support pin to pin timing.

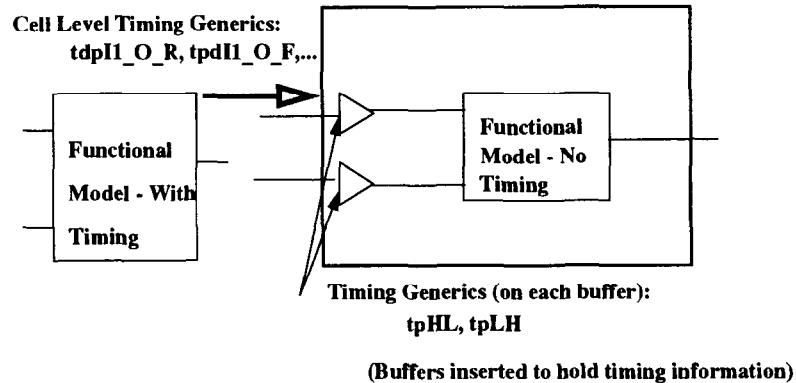
- Hardware acceleration requires special “macro” ASIC cells.
- Backannotate structural “macro” cells.
- Model pin-to-pin timing at the primitive level using “delay buffers”.

VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

“Macro” Cell for Backannotation in HW Accel.



VIUF, October 92: ASIC Design Using VHDL
D. Smati, C. Charlwood, S. Chang

Copyright © Synopsys 1992

synopsys[®]

Conclusions

- **Practical approach to model ASIC libraries.**
- **Practical approach to backannotate timing values.**
- **Some hardware accelerators require specific ASIC libraries.**
- **Feasibility of VHDL backannotation without adding to the language.**
 - **Question: Would pin-to-pin timing be a useful addition to VHDL?**
- **Methodology has been successfully used to model VHDL ASIC libraries.**