VALID

SCALD Timing Verifier

FEATURES

- Identifies logic-level timing errors including:
  - Set-up and hold
  - Pulse Width
  - Clock Glitch
  - Interface Specification
- Generates timing diagrams or tables automatically
- Displays a concise summary of
  - Any timing errors in the design
  - All signal behavior
- Verifies either partial or complete designs
- Uses estimated or actual wire delays
- Test inputs or microcode are not required

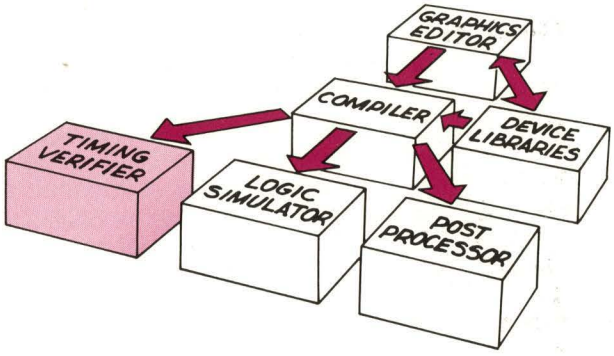
DESCRIPTION

The SCALD Timing Verifier is one of the most important validation tools available for the digital designer. This tool allows you to verify your design as it evolves even on a daily basis, assuring that it does not contain any timing errors when complete.

Identifies All Timing Errors

The powerful SCALD Timing Verifier identifies timing errors before hardware is built. The Timing Verifier uses the design database that has been created by the SCALD Compiler from your own schematics. From this database, the Timing Verifier assures that your partial or complete designs are free of timing errors. Working from the original schematic produced using the SCALD Graphics Editor saves you the task of re-entering your data and avoids potential inconsistency problems.

The Timing Verifier automatically checks to make sure there are no violations of either component specifications or designer specified



interface requirements. Timing Verification uses data that includes: estimated or actual wire delays, designer specified interface requirements and device parameters specified in component libraries. The parameters in these libraries include: propagation delays; set-up and hold times; pulse width constraints; and edge-to-edge constraints

Convenient Output Formats

The Timing Verifier output report consists of a concise listing of the value history of every signal in the design. This numeric representation helps you with the calculations required to correct timing errors. Verification also provides you with a detailed list of timing violations, and interface violations showing the signals involved as well as the type and times of each violation. In conjunction with the SCALD Graphics Editor, the Timing Verifier produces timing diagrams that graphically summarize a circuit's timing behavior (see Figure 1).

Verification Does Not Require Test Inputs

The SCALD Timing Verifier uses an eight-value logic system—zero, one, rising, falling, changing, stable, high impedance and undefined. The Timing Verifier preserves the logic behavior of the system when actual values (0, 1, R, F) are known. In other cases, when actual behavior is not known, the Timing Verifier represents the signal as stable, changing, or undefined. Since verification does not

require actual logic values, microcode and test vectors are unnecessary. The eighth value, high impedance, makes the SCALDsystem™ ideal for working with tri-state buses.

SCALD Timing Models

The Timing Verifier can utilize information contained in the extensive SCALD Component Libraries or in libraries you create. These libraries contain component timing models with specifications of set-up and hold times, pulse widths, edge-to-edge times and min-max propagation delays for both rising and falling edge signals. VALID provides SCALDsystem™ users with component libraries for TTL, STTL, LSTTL, 10K ECL, 100K ECL, CMOS,and memory devices. Since timing models are simply logic diagrams, you can modify timing specifications to suit your needs and easily create your own new models.

The convenient and concise SCALD III signal syntax utilized by the Timing Verifier, allows specification of input and output signal timing behavior. The assertions used by the SCALDsystem™ can specify undriven inputs or check the timing behavior of driven output. Thus the SCALD Timing Verifier is ideal for ensuring that a design meets its interface specifications. The Timing Verifier also understands that clock signals are often special and can correctly verify designs containing tuned or gated clocks.

Wire Delays

No timing verification system is complete without the ability to handle wire delays. With the SCALD Timing Verifier, you can attach a delay to a specific signal on your schematic. Alternately the Timing Verifier accepts a list of wire delays you compute or those from a physical design system. If neither method is used to specify wire delays, you can specify your own default minimum and maximum delays. Wire delays can be separately specified for both rising and falling edges—just as with components—a necessary feature for MOS or CMOS design.

A Convenient Tool

The Timing Verifier is easy to use—only your schematics are required as inputs. The timing models are complete including wire and component delays (both rising and falling) and timing constraints. Finally, verification of partial designs is simple. These features add up to a designer's tool that can be used from beginning to end of the design process, offering you continuous feedback on the timing performance and correctness of your digital system design.

Timing Verification can be used either with a standalone SCALDsystem™ or can be run on a host computer. Timing Verification is a very important part of the design validation process. The SCALD Timing Verification concept is a unique feature that was invented by VALID engineers.

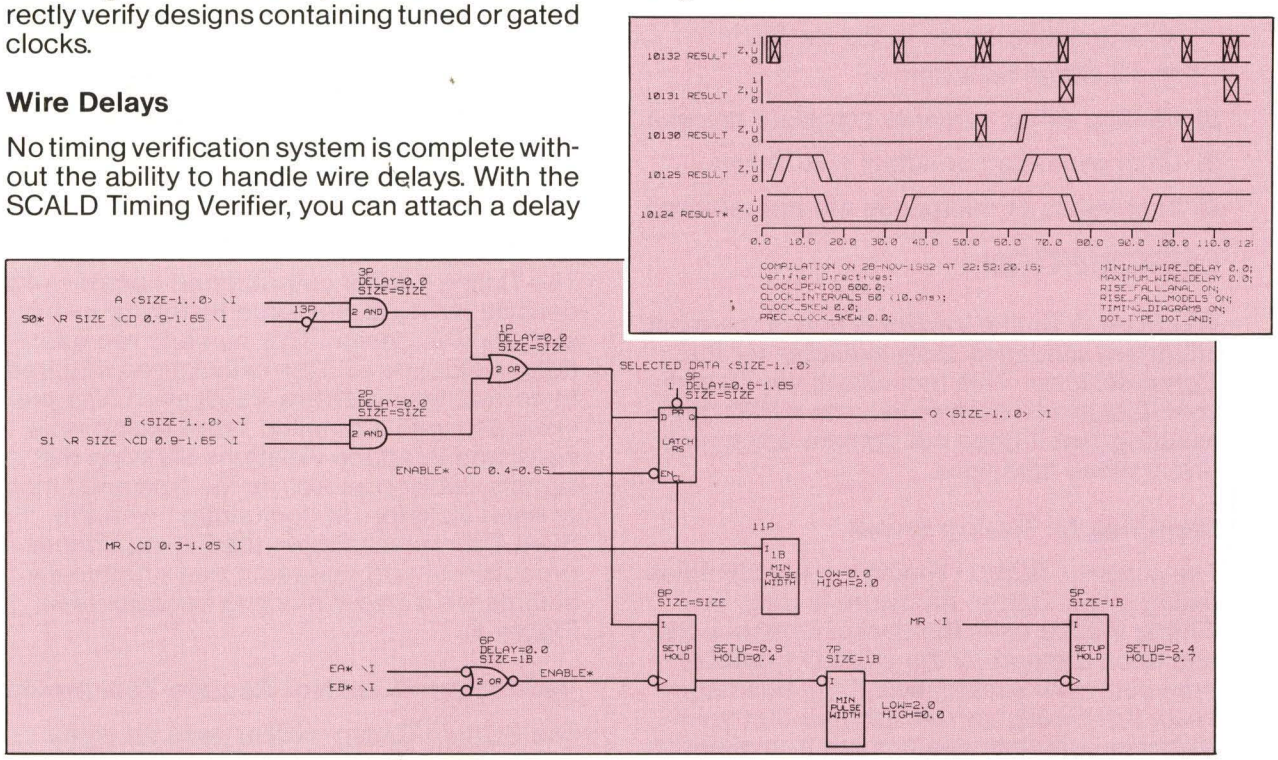


Figure 1. This is an example of a common timing model contained in the device libraries. Inset is a typical timing diagram provided by the SCALD Timing Verifier.

Valid Logic Systems Incorporated
650 North Mary Avenue
Sunnyvale, CA 94086
(408) 773-1300, TWX 910-339-9618

DS110 12/82 10K

Printed in U.S.A.