Clock - Reset VIP

Application Note

Revision 1.0

Table of Contents

1 Overview

The VerificationWorks™ Clock-Reset VIP allows the rapid implementation of an ideal clock and reset with the flexibility to introduce variations at a later stage of verification with no disruption to the existing structure.

The Clock-Reset VIP encapsulates a configurable number of clock agents and a configurable number of reset agents.

Each agent can be independently configured as active or passive. This is of benefit for vertical integration, allowing higher level testbenches to deconfigure the clock agent, but leave the reset agent intact as a passive monitoring VIP. Such usage preserves any prediction code relative to reset assertion at the block level.

For example, a unit level testbench may use scoreboarding methods to check the flow of transactions through the DUT. Warm reset testing may require that the scoreboard be flushed out, as anything still in flight will be lost. In this case, transactions from the reset agent's monitor component are passed to the environment's prediction logic via TLM. The prediction logic retrieves these transactions, notes a reset assertion based on transaction content, and issues reset to the scoreboard component. If this unit level testbench is brought into a system testbench, the need to reset the scoreboard remains. Configuring the reset agent as passive allows the VIP to be attached to the reset signal arriving at the original unit's boundary, preserving the scoreboard reset mechanism without further intervention from the system level testbench.

Multiple agent instances can be configured independently to provide clocks of related frequency, or of wholly asynchronous nature.

1.1 Clock Agent

Each Clock Agent includes the following features:

- Can be configured as active or passive agent. Active agent includes the clk_driver UVM component. Passive agent does not, and is effectively removed completely. Agents are typically configured as active in the top level testbench, and as passive when the containing testbench is itself a component of a higher level testbench.
- clk_driver component generates a free-running clock signal according to the configured parameters noted below.
  - Independently configurable nominal clock period
  - Independently configurable clock jitter, allowing cycle-to-cycle variation in clock period
  - Independently configurable duty cycle
  - Independently configurable starting offset, allowing for skewing of edges between different clock agents
  - Independently configurable starting signal level
- No sequences required to drive clock

1.2 Reset Agent

Each Reset Agent includes the following features:

- Can be configured as an active or passive agent. Active agent includes the rst_driver and rst_sequencer UVM components. Passive agent does not. Agents are typically configured as active in the top level testbench and as passive when the containing testbench is itself a component of a higher level testbench. A passive agent will typically retain its rst_monitor component, allowing the testbench to be notified of changes on the reset signal where necessary. Typical uses for this would be to reset any register shadow state, and to flush out any scoreboarded predictions, as well as resetting any other variables used in prediction code.

- rst_sequencer component generates 'canned' sequences to control what is driven by the rst_driver component.
  - canned sequence provided to generate a simple change of signal level, and ensure it is maintained for a desired minimum number of cycles.
  - canned sequence provided to produce a pulse of configurable duration.
- rst_driver component manages driving of its associated reset signal according to parameters of a sequence received from the rst_sequencer component.
- rst_monitor component can be separately deconfigured if not required. This would be the case for a testbench that does not need to reset CSR state, flush scoreboards, etc, due to assertion of reset. Deconfiguring the rst_monitor under such circumstances eliminates unnecessary components from the testbench, reducing image size for simulation.
- rst_monitor component provides testbench with visibility to reset assertion and negation via analysis port TLM
- Independently configurable reset polarity
- Independently configurable selection of synchronous or asynchronous behavior
- Independently configurable selection of relative clock edge
- Independently configurable skew range from clock edge (for asynchronous assertion)

2 Company Background

Paradigm Works is a leading chip design and verification services company. The company is recognized for engineering excellence, integrity in business, and overall productivity and cost effectiveness.

We provide expert consultants and contractors both on site and offshore to assist in complex chip developments. We offer world class domain expertise (PCI Express, USB, Ethernet), application knowledge (Networking, Computing, Storage, Wireless), and leverage Paradigm Works suite of productivity accelerator software (VerificationWorks™, ReleaseWorks®, and SystemVerilog FrameWorks™) to help clients bring their innovations to market as quickly as possible.

For more information on Paradigm Works products and services call 978-824-1400 or see our web site at www.paradigm-works.com.