create_test_protocol

NAME

create_test_protocol

Creates a test protocol based on user specifications.

SYNTAX

```
status create_test_protocol
    [-infer_asynch]
    [-infer_clock]
    [-capture_procedure single_clock | multi_clock]
```

ARGUMENTS

-infer_asynch

Infers asynchronous set and reset signals in the design.

-infer_clock

Infers test clocks in the design.

-capture_procedure single_clock | multi_clock

Specifies the capture procedure type. The **multi_clock** type creates a protocol file that uses generic capture procedures for all capture clocks. The **single_clock** type creates a protocol file that uses the legacy 3-vector capture procedures for all capture clocks.

The default is **multi_clock**.

DESCRIPTION

The **create_test_protocol** command creates a test protocol for the current design based on user specifications issued prior to running this command. The specifications are made using commands such as **set_dft_signal**.

This command removes any protocol that is present in memory due to a previous execution of **create_test_protocol**. However, if the protocol is present in memory due to a previous execution of **read_test_protocol**, it issues a warning and does not create a new test protocol.

This command checks whether the user-specified values are consistent with each other. If they are not, it issues an error and does not generate a protocol.

If **-infer_asynch** is specified, **create_test_protocol** infers asynchronous set and reset signals in the design, and places them at off state during scan shifting.

If **-infer_clock** is specified, **create_test_protocol** infers test clock pins from the design, and pulses them during scan shifting. The timing of the test clock is based on the **test_default_period**, **test_default_delay**, **test_default_strobe**, and **test_default_strobe_width variables**.

Both **-infer_asynch** and **-infer_clock** take previous user specifications into account.

When the default **-capture_procedure multi_clock** is specified, the protocol file contains four capture procedures: multiclock_capture, allclock_capture, allclock_launch, and allclock_launch_capture. This single protocol file can be used for ATPG stuck-at, transition delay, and path delay testing.

The **create_test_protocol** command automatically generates a master_observe procedure for LSSD designs in STIL format. However, to use this feature, you must set the scan style to LSSD design by using the **set_scan_configuration -style** command.

The **create_test_protocol** command should be executed before running the **dft_drc** command because design rule checking requires a test protocol.

If the interface of a design changes, for example, a port is created or removed, after a test protocol is created, there is no need to rerun **create_test_protocol** because the interface change is automatically taken into account in the protocol.

However, if you change any test specification of the design, for example, if you use **set_dft_signal** to specify a test clock, the protocol present in the memory is deleted. In this case, you need to rerun **create_test_protocol** to create a test protocol. The commands that change the test specifications of the existing design include **set_dft_signal** and **set_scan_path**.

The **insert_dft** command automatically updates the protocol after inserting scan circuitry into the design, and the **dft_drc** command can be executed afterward without rerunning **create_test_protocol**.

EXAMPLES

The following example creates a test protocol in memory for the current design:

```
prompt> create_test_protocol
```

The following example creates a test protocol in memory for the current design, inferring both asynchronous set and reset signals, as well as test clocks:

```
prompt> create_test_protocol -infer_clock -infer_asynch
```

SEE ALSO

current_design(2)
read_test_protocol(2)
remove_test_protocol(2)
set_dft_signal(2)
set_scan_configuration(2)
set_scan_path(2)
write_test_protocol(2)