Serial Vector Format Specification

Copyright © 1997-1999
ASSET InterTech, Inc.

Copyright © 1994
Texas Instruments Inc.

Users of this document are granted the right to copy and use the information in the document at no cost. Users may not make changes to the document, or redistribute the document under another name.

Revision E
8 March 1999
Part Number: ASSET-SVF-DOC

ASSET InterTech, Inc.

Table of Contents

INTRODUCTION....5

SVF COMMAND DETAILS....7

ENDDR, ENDIR....8
FREQUENCY....9
HDR, HIR (Header Data Register, Header Instruction Register)....10
PIO (Parallel Input/Output)....13
PIOMAP (Parallel Input/Output Map)....14
RUNTEST....16
SDR, SIR (Scan Data Register, Scan Instruction Register)....19
STATE....21
TDR, TIR (Trailer Data Register, Trailer Instruction Register)....23
TRST (Test ReSeT)....26

INTRODUCTION

This document describes the syntax for a Serial Vector Format (SVF) file. SVF is the media for exchanging descriptions of high-level IEEE 1149.1 bus operations. In general, IEEE 1149.1 bus operations consist of scan operations and movements between different stable states on the IEEE 1149.1 state diagram. SVF does not explicitly describe the state of the IEEE 1149.1 bus at every Test Clock.

SVF is designed to encourage reuse of serial vectors throughout the life cycle of the product, from its inception in the design phase to its deployment in the field service phase, and all phases in between. Life-cycle portability places restrictions on the design and capabilities of SVF.

The SVF file is defined as an ASCII file that consists of a set of SVF statements. The maximum number of characters allowed on a line is 256, although one SVF statement can span more than one line. Each statement consists of a command and associated parameters. Each SVF statement is terminated by a semicolon. SVF is not case sensitive. Comments can be inserted into a SVF file after an exclamation point '!' or a pair of slashes '//'. Either '//' or '!' will comment out the remainder of the line.

Scan data within a statement is expressed as hexadecimal and is always enclosed in parentheses. The scan data cannot specify a data string that is larger than the specified bit length. Most Significant Bit (MSB) zeroes in the hex string are not considered when determining if the string is too large. The bit order for scan data follows the convention that the least significant bit (rightmost bit) is the first bit scanned into the hardware for TDI and SMASK scan data and is the first bit scanned out for TDO and MASK data. This bit ordering is consistent with the IEEE 1149.1 convention.

An example of an SVF file is:

```
!Begin Test Program
TRST OFF;                                  !Disable Test Reset line
ENDIR IDLE;                                !End IR scans in IDLE
ENDDR IDLE;                                !End DR scans in IDLE
HIR  8  TDI (00);                          !8-bit IR header
HDR 16  TDI (FFFF) TDO (FFFF) MASK (FFFF);!16-bit DR header
TIR 16  TDI (0000);                        !16-bit IR trailer
TDR  8  TDI (12);                          !16-bit DR trailer
SIR  8  TDI (41);                          !8-bit IR scan
SDR 32  TDI (ABCD1234) TDO (11112222);     !32-bit DR scan
STATE   DRPAUSE;                           !Go to stable state DRPAUSE
RUNTEST 100 TCK ENDSTATE IRPAUSE;          !RUNBIST for 100 TCKs
!End Test Program
```

The following set of SVF commands are supported:

ENDDR .....Specifies default end state for DR scan operations.

ENDIR .....Specifies default end state for IR scan operations.

FREQUENCY....Specifies maximum test clock frequency for IEEE 1149.1 bus operations.

HDR .....(*Header Data Register*) Specifies a header pattern that is prepended to the beginning of subsequent DR scan operations.

HIR.....(*Header Instruction Register*) Specifies a header pattern that is prepended to the beginning of subsequent IR scan operations.

PIO.....(*Parallel Input/Output*) Specifies a parallel test pattern.

PIOMAP .....(*Parallel Input/Output Map*) Maps PIO column positions to a logical pin.

RUNTEST.....Forces the IEEE 1149.1 bus to a run state for a specified number of clocks or a specified time period.

SDR.....(*Scan Data Register*) Performs an IEEE 1149.1 Data Register scan.

SIR .....(*Scan Instruction Register*) Performs an IEEE 1149.1 Instruction Register scan.

STATE .....Forces the IEEE 1149.1 bus to a specified stable state.

TDR.....(*Trailer Data Register*) Specifies a trailer pattern that is appended to the end of subsequent DR scan operations.

TIR .....(*Trailer Instruction Register*) Specifies a trailer pattern that is appended to the end of subsequent IR scan operations.

TRST.....(*Test ReSeT*) Controls the optional Test Reset line.

SVF COMMAND DETAILS

Each command and associated parameters are described in the remainder of this document. Parameters are mandatory, unless enclosed by brackets '[ ]', which indicates the enclosed parameter is optional. For the purposes of this document, a scan operation is defined as the execution of a SIR or SDR command and any associated header or trailer commands. Some optional command parameters such as, MASK, SMASK, and TDI are "sticky" (they are remembered from the previous command until changed or invalidated) to minimize SVF file size. The MASK, SMASK, and TDI parameters are "remembered" separately for SIR, SDR, HIR, HDR, TIR, and TDR commands. Some SVF commands reference IEEE 1149.1 TAP states. The following table lists each SVF state name used for each IEEE 1149.1 TAP state name.

IEEE 1149.1 TAP State Name	SVF TAP State Name
Test-Logic-Reset	RESET
Run-Test/Idle	IDLE
Select-DR-Scan	DRSELECT
Capture-DR	DRCAPTURE
Shift-DR	DRSHIFT
Exit1-DR	DREXIT1
Pause-DR	DRPAUSE
Exit2-DR	DREXIT2
Update-DR	DRUPDATE
Select-IR-Scan	IRSELECT
Capture-IR	IRCAPTURE
Shift-IR	IRSHIFT
Exit1-IR	IREXIT1
Pause-IR	IRPAUSE
Exit2-IR	IREXIT2
Update-IR	IRUPDATE

Real numbers in SVF are specified using the syntax:

digits [. *digits*] [E [+|-] *digits*]

The *digits* consist of one or more decimal digits 0-9. This syntax matches the decimal literals of VHDL excluding underlines. Thus, 1, 1E0, 1E+0, 1E-0, 1.0, 1.0E0, 1.0E+0, and 1.0E-0 are all equivalent, valid real numbers; 1., 1.E0, .5, and .5E0 are not valid. The range and precision of real numbers is implementation defined.

ENDDR, ENDIR

Syntax:

ENDDR *stable_state*;
ENDIR *stable_state*;

Purpose:

Establishes the IEEE 1149.1 end state for scan operations.

Parameters:

stable_state ..... The stable IEEE 1149.1 state that the IEEE 1149.1 bus will be forced to at the conclusion of a scan operation. Valid stable states are **IRPAUSE**, **DRPAUSE**, **RESET**, and **IDLE**.

General Information:

The ENDDR and ENDIR commands specify the IEEE 1149.1 stable state that the IEEE 1149.1 bus will be forced to at the conclusion of a DR or IR scan, respectively. Once specified, the ENDDR/ENDIR commands remain in force until overridden by another ENDDR or ENDIR command. At startup, ENDDR and ENDIR are both set to IDLE.

Examples:

```
ENDIR IDLE;
ENDDR DRPAUSE;
```

FREQUENCY

Syntax:

FREQUENCY [*cycles* HZ];

Purpose:

Establishes a maximum IEEE 1149.1 test clock (TCK) frequency for subsequent scans (SDR and SIR), state changes (STATE), and test operations (RUNTEST). The new frequency remains in effect until the next FREQUENCY statement is executed or the end of file is reached. The maximum safe frequency may be a characteristic of the UUT, the fixturing, or any RUNBIST or INTEST designs. Omitting *cycles* removes any restriction on the maximum frequency, enabling the test to return to a higher clock rate after being temporarily slowed down.

Parameters:

cycles.....[Optional] The maximum TCK rate in Hz expressed as a real number greater than zero (0).

General Information:

The value *cycles* is specified as a real number. The range, number, and accuracy of supported frequencies is implementation defined. An implementation that is unable to source its TCK at or below the frequency specified should report an error. If *cycles* is specified, the new frequency takes effect before the next command whose behavior is altered by it; i.e. before the next RUNTEST, SDR, SIR, or STATE command.

The initial frequency is implementation defined, and is probably specified external to the SVF file by the user based on the characteristics of the fixturing to the UUT. If so, omitting *cycles* returns to this externally-specified frequency. Whether the value of *cycles* is used when it exceeds the externally-specified frequency is implementation defined. Each SVF file ends with an implied FREQUENCY; command so that one SVF file may not affect another SVF file.

An implementation shall report an error on a RUNTEST command if *cycles* is specified, and the RUNTEST command specifies both a TCK *run_count* and a *max_time* that cannot be met given the specified maximum frequency.

Examples:

```
SIR 8 TDI(F3) TDO(01) MASK(03);   ! Set up BIST, full speed
FREQUENCY 90E3 HZ;                ! Decrease to 90 kHz
RUNTEST 100000 TCK;               ! Execute BIST
FREQUENCY 1E5 HZ;                 ! Increase to 100 kHz
RUNTEST 300000 TCK 1 SEC          ! Error! 300000 TCK at 100 kHz is
   MAXIMUM 2 SEC;                 !    3 SEC, but MAXIMUM is 2 SEC
FREQUENCY;                        ! Return to full speed
```

HDR, HIR (Header Data Register, Header Instruction Register)

Syntax:

HDR *length* [TDI (*tdi*)] [TDO (*tdo*)] [MASK (*mask*)] [SMASK (*smask*)];

HIR *length* [TDI (*tdi*)] [TDO (*tdo*)] [MASK (*mask*)] [SMASK (*smask*)];

Purpose:

Specifies a default header pattern that is shifted in before every scan operation. This command is intended to allow a set of scan statements for a specific IEEE 1149.1 component to be easily adapted to an environment where the component is placed in a scan path containing other IEEE 1149.1 devices. The header pattern specifies how to pad the scan statements with a set of leading bits that accommodate the devices located on the scan path beyond the component of interest.

Parameters:

length ..... A 32-bit unsigned decimal integer specifying the number of bits to be scanned. Setting the length to 0 removes the header.

[TDI (*tdi*)] ..... [Optional] The value to be scanned into the target expressed as a hex value. If this parameter is not present, the value of TDI to be scanned into the target will equal the TDI value specified on the last previous HDR/HIR statement. HDR and HIR TDI values are “remembered” independently. The TDI parameter must be explicitly specified for the first command or when the length changes.

[TDO (*tdo*)] ..... [Optional] The values to be compared against the actual values scanned out of the target, expressed as a hex string. If this parameter is not present, no comparison will be performed.

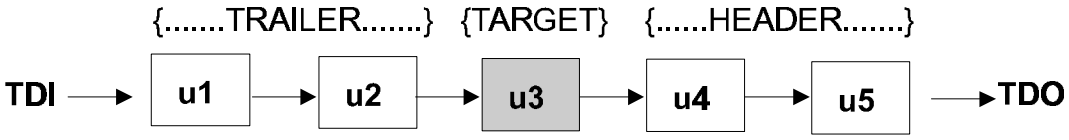
[MASK (*mask*)] ..... [Optional] The mask to be used when comparing TDO values against the actual values scanned out of the target, expressed as a hex string. A ‘1’ in a specific bit position indicates a care for that position. A ‘0’ indicates a don't care. If this parameter is not present, the mask used will equal the last previously specified MASK value specified for the HDR/HIR statement. HDR and HIR MASK values are “remembered” independently. If a header command changes the *length* of the last header of the same type and the MASK parameter is absent, the *mask* pattern used is all cares. If no TDO parameter is present, the MASK will not be used.

[SMASK (*smask*)] .....[Optional] Specifies that TDI data is don't care, expressed as a hex string. A ‘1’ in a specific bit position indicates the TDI data in that bit position is a care. A ‘0’ indicates a don't care. If this parameter is not present, no masking for all bits are assumed. If this parameter is not present, the mask used will equal the last previously specified SMASK value specified for the HDR/HIR statement. HDR and HIR SMASK values are “remembered” independently. If a header command changes the *length* of the last header of the same type and the SMASK parameter is absent, the *smask* pattern used is all cares. The SMASK will be used even if the TDI parameter is not present.

General Information:

The HDR (Header Data Register) specifies a default header pattern that will be prepended to the beginning of all subsequent SDR commands. HIR (Header Instruction Register) specifies a default header pattern that will be prepended to the beginning of all subsequent SIR commands. The header commands have a set of counterpart trailer commands (TIR, TDR) that are described later. A header can be removed by setting *length* to 0.

For example, assume a SVF file is developed for an ASIC. The ASIC is then placed in a board as u3, shown below:



The set of SVF statements originally developed for the ASIC can be reused with minimal modification if the appropriate header and trailer statements are defined to accommodate the devices in front of and behind u3. In this example, a header pattern would be defined for devices u4 and u5, and a trailer pattern would be defined for u2 and u1.

The optional parameters can be specified in any order. Each optional parameter can only be specified once. Hex strings specified for TDI, TDO, MASK, or SMASK cannot be a value larger than the maximum implied by the *length* parameter. Leading zeroes are assumed for a hex string if not explicitly specified.

Examples:

```
HDR 32 TDI(00000010) TDO(81818181) MASK(FFFFFFFF) SMASK(0);
HIR 16 TDI(ABCD);
...
HDR 0;   ! Removes the previous DR scan header.
```

PIO (Parallel Input/Output)

Syntax:

PIO (*vector_string*);

Purpose:

Specifies a parallel test vector. A PIOMAP statement must have been previously defined and the number of vector characters in a PIO statement must equal the number of logical names in the previously specified PIOMAP statement.

Parameters:

vector_string ..... A column-oriented set of one or more vector characters. Each character specifies the direction and state for a specific pin for one test vector. The following characters are defined:

H..... Drive Logical 1
L..... Drive Logical 0
Z..... Drive High Impedance
U..... Detect Logical 1
D..... Detect Logical 0
X..... Detect Unknown

Note: For bidirectional channels a “Z” implies no drive/no detect.

Examples:

```
PIO (HLUDXZHHLL);
```

PIOMAP (Parallel Input/Output Map)

Syntax:

PIOMAP (*direction*$_1$ *logical_name*$_1$…[*direction*$_n$ *logical_name*$_n$]);

PIOMAP (*column*$_1$ *logical_name*$_1$…[*column*$_n$ *logical_name*$_n$]);

Purpose:

Defines the I/O direction and logical name for each column in a PIO statement. The PIOMAP statement is required if the SVF file uses the PIO statement. If PIO is used, the PIOMAP statement must be placed in the SVF file prior to any SIR, SDR, STATE, RUNTEST, or PIO statement. Only one PIOMAP statement is allowed per file.

The first form of PIOMAP defines the I/O direction and logical name for each column, in the order the vector characters appear in the PIO statement. The first logical_name corresponds to the first character of the vector_string in PIO statements; the second logical_name corresponds to the second character of the vector_string; and so forth.

The second form of PIOMAP defines the mapping between a specific column in a PIO statement and the logical name associated with that column. The I/O direction is not defined, so any vector character may be used in any column. This second form is not recommended for current usage because the I/O direction cannot be validated.

Parameters:

***direction*$_{1-n}$** ..... Specifies the I/O direction of the logical name. The *direction* may be **IN**, **OUT**, or **INOUT**. A column with a direction of **IN** is an input to the UUT, and that column in the PIO statement may use the "drive" characters. An **OUT** column is an output from the UUT, and may use the "detect" characters. An **INOUT** column is a bidirectional signal, and may use any vector character.

***logical_name*$_{1-n}$** ..... A character string identifying the logical name of the pin associated with the column. The same string cannot be used more than once in the PIOMAP statement.

***column*$_{1-n}$** ..... Specifies the column number of the PIO statement. The first character in the PIO statement is column 1, the second character column 2, etc. The column number is decimal. A specific column number cannot be used more than once. This form of PIOMAP is not recommended for current usage.

Examples:

```
!PIOMAP must be placed before PIO statement
PIOMAP (IN  STROBE
        IN  ALE
        OUT DISABLE
        OUT ENABLE
        OUT CLEAR
        IN  SET);

PIO (HLUDXZ);

!Vector is:
!  STROBE  <- H
!  ALE     <- L
!  DISABLE <- U
!  ENABLE  <- D
!  CLEAR   <- X
!  SET     <- Z
```

RUNTEST

Syntax:

RUNTEST [*run_state*] *run_count run_clk* [*min_time* SEC [MAXIMUM *max_time* SEC]] [ENDSTATE *end_state*];

RUNTEST [*run_state*] *min_time* SEC [MAXIMUM *max_time* SEC] [ENDSTATE *end_state*];

Purpose:

Forces the target IEEE 1149.1 bus to the specified run state for a specified number of clocks (either Test Clocks or System Clocks), a specified length of time, or both, then moves the target bus to the specified end state. This can be used to control RUNBIST operations in the target.

The first form of the RUNTEST command executes the test in the specified run state for the specified number of clocks. Optionally, a minimum and maximum time delay in seconds can be specified. The second form of the RUNTEST command executes the test for the specified minimum and maximum number of seconds. Since RUNTEST is a command that generates vectors, TCK is clocked even if *run_count* is not specified.

Either *run_count* or *min_time* must be specified. If both *run_count* and *min_time* are specified, both conditions must be met before the RUNTEST command finishes execution. If *max_time* is exceeded, RUNTEST stops before the *run_count* is reached.

Both *min_time* and *max_time* are specified as real numbers. The resolution and range of the time delay is implementation defined.

The translator shall issue a warning if it cannot guarantee the maximum time when the current state before execution of the RUNTEST command is the same as the *run_state*, and a *max_time* is specified. The translator shall also issue a portability warning that the maximum time may be violated in some implementations under these conditions. An implementation that uses a free-running TCK could be in the run state and clocking longer than *max_time*, since it would have entered the *run_state* at the end of the previous command, not at the start of the RUNTEST command.

Parameters:

run_state.....[Optional] The stable IEEE 1149.1 state that the IEEE 1149.1 bus will be forced to during the RUNTEST command. Valid run states are **IRPAUSE**, **DRPAUSE**, **RESET**, and **IDLE**. If the test bus is already in the run state, no state transitions occur. Once a *run_state* is specified, subsequent RUNTEST commands will default to the same run state if one is not specified. The initial default is **IDLE**. For information on the default path taken when moving from one IEEE 1149.1 state to another, refer to the table on page 22.

run_count .....[Optional] The number of clocks that the IEEE 1149.1 bus will remain in the run state, expressed as a 32-bit unsigned decimal integer greater than 0.

run_clk .....[Optional] Specifies the clock used, either TCK (Test Clock) or SCK (System Clock). System Clock refers to a clock on the UUT that is asynchronous with respect to the TCK. The System Clock is implementation dependent.

[*min_time* SEC] .....[Optional] The minimum amount of time in seconds that the RUNTEST command must execute before finishing.

[MAXIMUM *max_time* SEC] ..[Optional] The maximum amount of time in seconds that the RUNTEST command may execute before it must finish. The maximum time must be greater than the minimum time. If both *run_count* and *max_time* are specified, and not all clocking has completed when the *max_time* is reached, the command finishes even though not all the requested clocking has been performed. Whether or not a maximum time can be guaranteed is implementation defined.

[ENDSTATE *end_state*] .....[Optional] The stable IEEE 1149.1 state that the IEEE 1149.1 bus will be forced to after executing the specified number of clocks, waiting the specified length of time, or both. Valid end states are **IRPAUSE**, **DRPAUSE**, **RESET**, and **IDLE**. If the test bus is already in the end state, no state transitions occur. If the *end_state* is not specified, the default end state is used. When an *end_state* is specified, it becomes the default. When a *run_state* is

specified, the new *run_state* becomes the default *end_state*. When a *run_state* is not specified, the default *end_state* remains in effect. The initial default for *end_state* is **IDLE**.

Examples:

```
! Run in Run-Test/Idle for 1000 TCKs, then go to Pause-DR.
RUNTEST 1000 TCK ENDSTATE DRPAUSE;
! Go back to Run-Test/Idle for 20 SCKs, then go to Pause-DR.
RUNTEST 20 SCK;
! Run in Run-Test/Idle for 1000000 TCKs or at least one second,
! then go to Pause-DR.
RUNTEST 1000000 TCK 1 SEC;
! Run in Run-Test/Idle for at least one millisecond and at most
! 50 milliseconds, then remain in Run-Test/Idle.
RUNTEST 10.0E-3 SEC MAXIMUM 50.0E-3 SEC ENDSTATE IDLE;
! Run in Pause-DR for at least 50 ms, then go to Run-Test/Idle.
RUNTEST DRPAUSE 50E-3 SEC ENDSTATE IDLE;
! Run in Pause-DR for at least one second, then go to Run-Test/Idle.
RUNTEST 1 SEC;
! Run in Run-Test/Idle for at least 10 ms, then remain in
! Run-Test/Idle.
RUNTEST IDLE 1E-2 SEC;
```

SDR, SIR (Scan Data Register, Scan Instruction Register)

Syntax:

SDR *length* [TDI (*tdi*)] [TDO (*tdo*)][MASK (*mask*)] [SMASK (*smask*)];

SIR *length* [TDI (*tdi*)] [TDO (*tdo*)][MASK (*mask*)] [SMASK (*smask*)];

Purpose:

Specifies a scan pattern to be applied to the target scan register(s).

Parameters:

length ..... A 32-bit unsigned decimal integer greater than 0 specifying the number of bits to be scanned.

[TDI (*tdi*)] ..... [Optional] The value to be scanned into the target, expressed as a hex value. If this parameter is not present, the value of TDI to be scanned into the target will equal the TDI value specified on the last previous SDR/SIR statement. SDR and SIR TDI values are “remembered” independently. The TDI parameter must be explicitly specified for the first command or when the length changes.

[TDO (*tdo*)] ..... [Optional] The values to be compared against the actual values scanned out of the target, expressed as a hex string. If this parameter is not present, no comparison will be performed. If no TDO parameter is present, the MASK will not be used.

[MASK (*mask*)] ..... [Optional] The mask to be used when comparing TDO values against the actual values scanned out of the target, expressed as a hex string. A ‘1’ in a specific bit position indicates a care for that position. A ‘0’ indicates a don't care. If this parameter is not present, the mask used will equal the last previously specified MASK value specified for the SDR/SIR statement. SDR and SIR MASK values are “remembered” independently. If a scan command changes the *length* of the last scan of the same type and the MASK parameter is absent, the *mask* pattern used is all cares. If no TDO parameter is present, the MASK will not be used.

[SMASK (smask)] ..... [Optional] Specifies that TDI data is don't care, expressed as a hex string. A ‘1’ in a specific bit position indicates the TDI data in that bit position is a care. A ‘0’ indicates a don't care. If this parameter is not present, the mask used will equal the last

previously specified SMASK value specified for the SDR/SIR statement. SDR and SIR SMASK values are “remembered” independently. If a scan command changes the *length* of the last scan of the same type and the SMASK parameter is absent, the *smask* pattern used is all cares. The SMASK will be used even if the TDI parameter is not present.

General Information:

The SDR (Scan Data Register) specifies a scan pattern to be scanned to the target Data Register. The SIR (Scan Instruction Register) specifies a scan pattern to be scanned to the target Instruction Register.

Prior to scanning the values specified in the SDR or SIR command, the last defined header command (HDR or HIR, respectively) will be prepended to the beginning of the SDR or SIR data pattern and the last defined trailer command (TDR or TIR, respectively) will be appended to the end of the SDR or SIR data pattern.

The optional parameters can be specified in any order. Each optional parameter can only be specified once. Hex strings specified for TDI, TDO, MASK, or SMASK cannot be a value larger than the maximum implied by the *length* parameter. Leading zeroes are assumed for a hex string if not explicitly specified.

Examples:

```
SDR 24 TDI(000010) TDO(818181) MASK(FFFFFF) SMASK(0);
SIR 16 TDI(ABCD);
```

STATE

Syntax:

STATE *[pathstate$_1$ [pathstate$_2$ ...[pathstate$_n$]]] stable_state* ;

Purpose:

Forces the target IEEE 1149.1 bus to a stable IEEE 1149.1 state.

Parameters:

pathstate$_{1-n}$ ..... An optional list of one or more IEEE 1149.1 states describing the explicit path through the Test Access Port (TAP) state diagram that is taken in order to reach the final stable state. Valid states are **RESET**, **IDLE**, **DRSELECT**, **DRCAPTURE**, **DRSHIFT, DRPAUSE**, **DREXIT1**, **DREXIT2**, **DRUPDATE**, **IRSELECT**, **IRCAPTURE**, **IRSHIFT, IRPAUSE**, **IREXIT1, IREXIT2,** and **IRUPDATE**. The states must be listed in an order that complies with the TAP state diagram. If the *pathstate* parameters are not listed, a default path is assumed based on the current state and the final stable state, as listed in the following table.

stable_state ..... The stable IEEE 1149.1 state that the IEEE 1149.1 bus will be forced to. Valid stable states are **IRPAUSE**, **DRPAUSE**, **RESET**, and **IDLE**.

Examples:

```
!Force bus to DRPAUSE from current state it is in
STATE DRPAUSE;

!Dictate explicit path bus will take moving from
! DRPAUSE to IRPAUSE
STATE DREXIT2 DRUPDATE DRSELECT IRSELECT IRCAPTURE IREXIT1 IRPAUSE;
```

General Information:

The STATE command is used to move the IEEE 1149.1 bus from one stable state to another for UUT test initialization, completion or TAP controller testing. TDI and TDO values are undefined when using the STATE command.

If no *pathstate* parameters are specified, the STATE command always follows the same path through the IEEE 1149.1 state diagram when moving from one stable state to another. The following table indicates the path taken for each stable state. For every state path, it is assumed that at least one Test Clock will be executed.

Current State	New State	State Path
RESET	RESET	RESET
RESET	IDLE	RESET-IDLE
RESET	DRPAUSE	RESET-IDLE-DRSELECT-DRCAPTURE-DREXIT1-DRPAUSE
RESET	IRPAUSE	RESET-IDLE-DRSELECT-IRSELECT-IRCAPTURE-IREXIT1-IRPAUSE
IDLE	RESET	IDLE-DRSELECT-IRSELECT-RESET
IDLE	IDLE	IDLE
IDLE	DRPAUSE	IDLE-DRSELECT-DRCAPTURE-DREXIT1-DRPAUSE
IDLE	IRPAUSE	IDLE-DRSELECT-IRSELECT-IRCAPTURE-IREXIT1-IRPAUSE
DRPAUSE	RESET	DRPAUSE-DREXIT2-DRUPDATE-DRSELECT-IRSELECT-RESET
DRPAUSE	IDLE	DRPAUSE-DREXIT2-DRUPDATE-IDLE
DRPAUSE	DRPAUSE	DRPAUSE-DREXIT2-DRUPDATE-DRSELECT-DRCAPTURE-DREXIT1-DRPAUSE
DRPAUSE	IRPAUSE	DRPAUSE-DREXIT2-DRUPDATE-DRSELECT-IRSELECT-IRCAPTURE-IREXIT1-IRPAUSE
IRPAUSE	RESET	IRPAUSE-IREXIT2-IRUPDATE-DRSELECT-IRSELECT-RESET
IRPAUSE	IDLE	IRPAUSE-IREXIT2-IRUPDATE-IDLE
IRPAUSE	DRPAUSE	IRPAUSE-IREXIT2-IRUPDATE-DRSELECT-DRCAPTURE-DREXIT1-DRPAUSE
IRPAUSE	IRPAUSE	IRPAUSE-IREXIT2-IRUPDATE-DRSELECT-IRSELECT-IRCAPTURE-IREXIT1-IRPAUSE

TDR, TIR (Trailer Data Register, Trailer Instruction Register)

Syntax:

TDR *length* [TDI (*tdi*)] [TDO (*tdo*)][MASK (*mask*)] [SMASK (*smask*)];

TIR *length* [TDI (*tdi*)] [TDO (*tdo*)][MASK (*mask*)] [SMASK (*smask*)];

Purpose:

Specifies a default trailer pattern that is shifted in after all subsequent scan operations. This command is intended to allow a set of scan statements for a specific IEEE 1149.1 component to be easily adapted to an environment where the component is placed in a scan path containing other IEEE 1149.1 devices. The trailer pattern specifies how to pad the scan statements with a set of trailing bits that accommodate the devices located on the scan path after the component of interest.

Parameters:

length ..... A 32-bit unsigned decimal integer specifying the number of bits to be scanned. Setting the length to 0 removes the trailer.

[TDI (*tdi*)] ..... [Optional] The value to be scanned into the target, expressed as a hex value. If this parameter is not present, the value of TDI to be scanned into the target will equal the TDI value specified on the last previous TDR/TIR statement. TDR and TIR TDI values are “remembered” independently. The TDI parameter must be explicitly specified for the first command or when the length changes.

[TDO (*tdo*)] ..... [Optional] The values to be compared against the actual values scanned out of the target, expressed as a hex string. If this parameter is not present, no comparison will be performed.

[MASK (*mask*)] ..... [Optional] The mask to be used when comparing TDO values against the actual values scanned out of the target, expressed as a hex string. A ‘1’ in a specific bit position indicates a care for that position. A ‘0’ indicates a don't care. If this parameter is not present, the mask used will equal the last previously specified MASK value specified for the TDR/TIR statement. TDR and TIR MASK values are “remembered” independently. If a trailer command changes the *length* of the last trailer of the same type and the MASK parameter is absent, the *mask* pattern used is

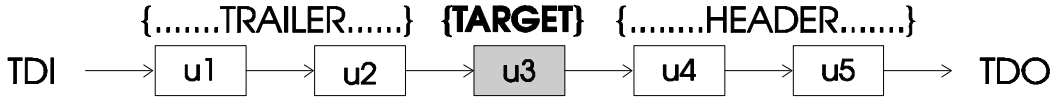
all cares. If no TDO parameter is present, the MASK will not be used.

[SMASK (*smask*)] .....[Optional] Specifies which TDI data is don't care, expressed as a hex string. A ‘1’ in a specific bit position indicates the TDI data in that bit position is a care. A ‘0’ indicates a don't care. If this parameter is not present, the mask used will equal the last previously specified SMASK value specified for the TDR/TIR statement. TDR and TIR SMASK values are “remembered” independently. If a trailer command changes the *length* of the last trailer of the same type and the SMASK parameter is absent, the *smask* pattern used is all cares. The SMASK will be used even if the TDI parameter is not present.

General Information:

The TDR (Trailer Data Register) specifies a trailer pattern that will be appended to the end of all subsequent SDR commands. TIR (Trailer Instruction Register) specifies a default trailer pattern that will be appended to the end of all subsequent SIR commands A trailer can be removed by setting *length* to 0.

For example, assume a SVF file is developed for an ASIC. The ASIC is then placed in a board as u3, shown below:



The set of SVF statements originally developed for the ASIC can be reused with minimal modification if the appropriate header and trailer statements are defined to accommodate the devices in front of and behind u3. In this example, a header pattern would be defined for devices u4 and u5, and a trailer pattern would be defined for u1 and u2.

The optional parameters can be specified in any order. Each optional parameter can only be specified once. Hex strings specified for TDI, TDO, MASK, or SMASK cannot be a value larger than the maximum implied by the *length* parameter. Leading zeroes are assumed for a hex string if not explicitly specified.

Examples:

```
TDR  32 TDI(00000010) TDO(81818181) MASK(FFFFFFFF) SMASK(0);
TIR  16 TDI(ABCD);
...
TDR  0;  ! Removes the previous DR scan trailer.
```

TRST (Test ReSeT)

Syntax:

TRST *trst_mode*;

Purpose:

The TRST command describes the operation of the optional Test Reset signal.

Parameters:

trst_mode..... Specifies whether the TRST line is active, inactive, is high impedance or even exists. Valid *trst_mode* states are:

ON.....Active (Logic 0)
OFF.....Inactive (Logic 1)
Z.....High Impedance
ABSENT.....Not present

If TRST with a parameter of ABSENT is specified, it must be placed at the beginning of the SVF file before any SIR, SDR, STATE, RUNTEST, or PIO statements, and no further TRST statements are allowed. The TRST statement is independent of the state of the IEEE 1149.1 TAP. This allows testing of a target using either the TRST line or the state of the IEEE 1149.1 TAP.

Examples:

```
TRST ON;
TRST OFF;
```