

# VTRAN<sup>®</sup> 9.2 Technical Summary

SOURCE III, INC.

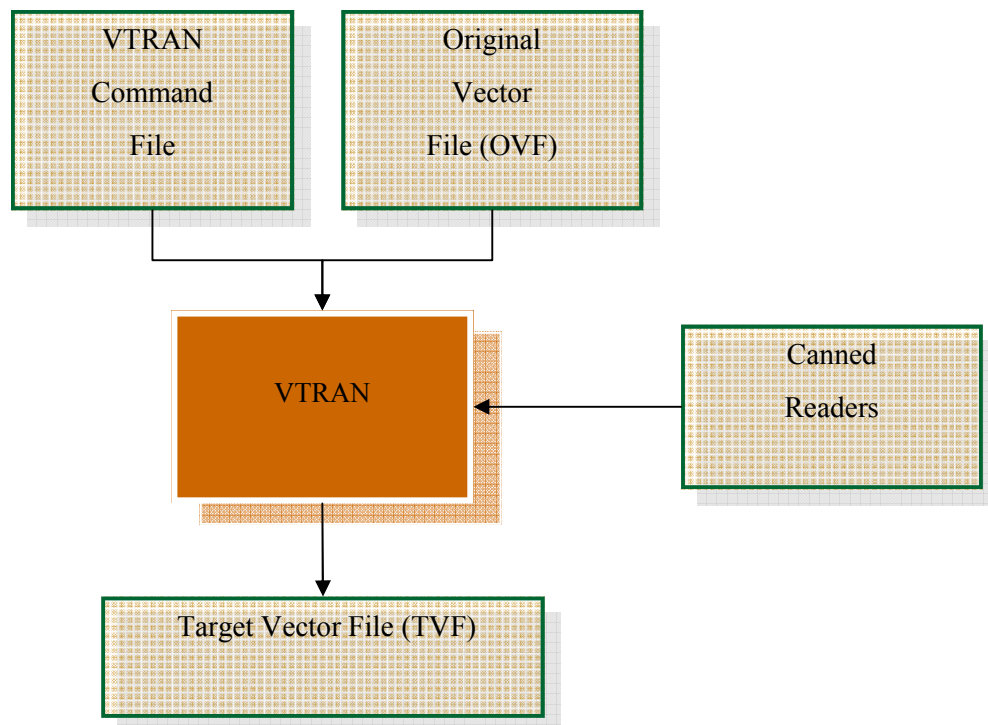
**V**TRAN is a vector translation program which takes vector data from either logic simulators or from ATPG tools, and translates it into test programs for ATE or into formats compatible with other EDA environments. In order to accomplish these conversions, VTRAN breaks the flow down to three main components – Reading the original vector file data, applying optionally Processing the vector data, and finally Writing out the desired vector data for the target environment (simulator, EDA tool or ATE).

VTRAN is driven by commands in a text file which specify the specifics for the translation. There are also a number of command line options that can be specified at invocation of the program. In order to be effective with using VTRAN's features, it is important to understand the command set and

the order of processing the program follows. This document presents a summary of VTRAN's commands, their syntax, and the order in which VTRAN applies them to the translation flow. For a more detailed description of each VTRAN command, see the VTRAN User's Guide.

## Overall VTRAN Flow

The following flow chart indicates the relationship between the different files during a translation. The Original Vector File (OVF) is the input file (WGL, STIL, VCD, EVCD, etc.....) which is to be translated by VTRAN. The Vtran Command File is the user-created text file which contains the instructions for the translation. The Canned Readers are a library of standard readers for handling most of the popular vector formats, and the Target Vector File (TVF) is the output file(s) produced by VTRAN.



## The VTRAN Command File

The VTRAN Command File is created by the user with a text editor and contains instructions which provide VTRAN with the details required for the translation to be performed. This file has 3 sections corresponding to the 3 components of the flow mentioned above. It has the following structure:

VTRAN Command File

```

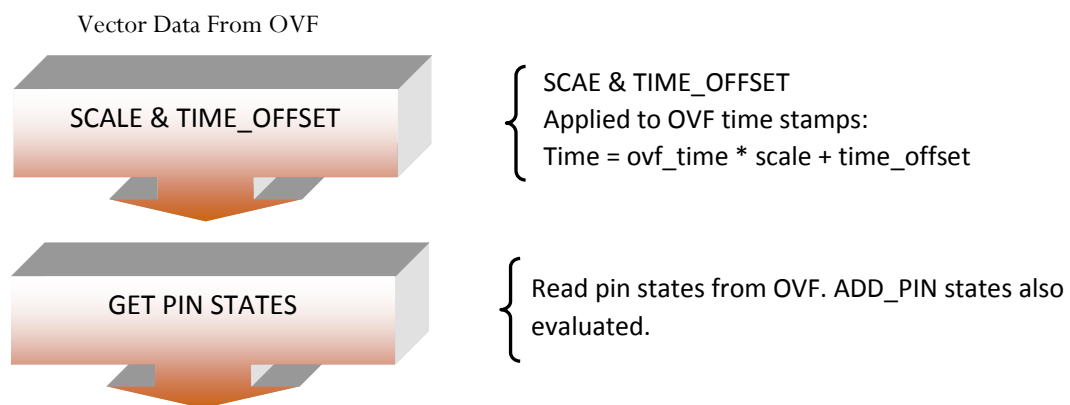
OVF_BLOCK
  BEGIN
    { Read Commands }
  END
PROC_BLOCK
  BEGIN
    { Processing Commands }
  END
TVF_BLOCK
  BEGIN
    { Writer Commands }
  END
END

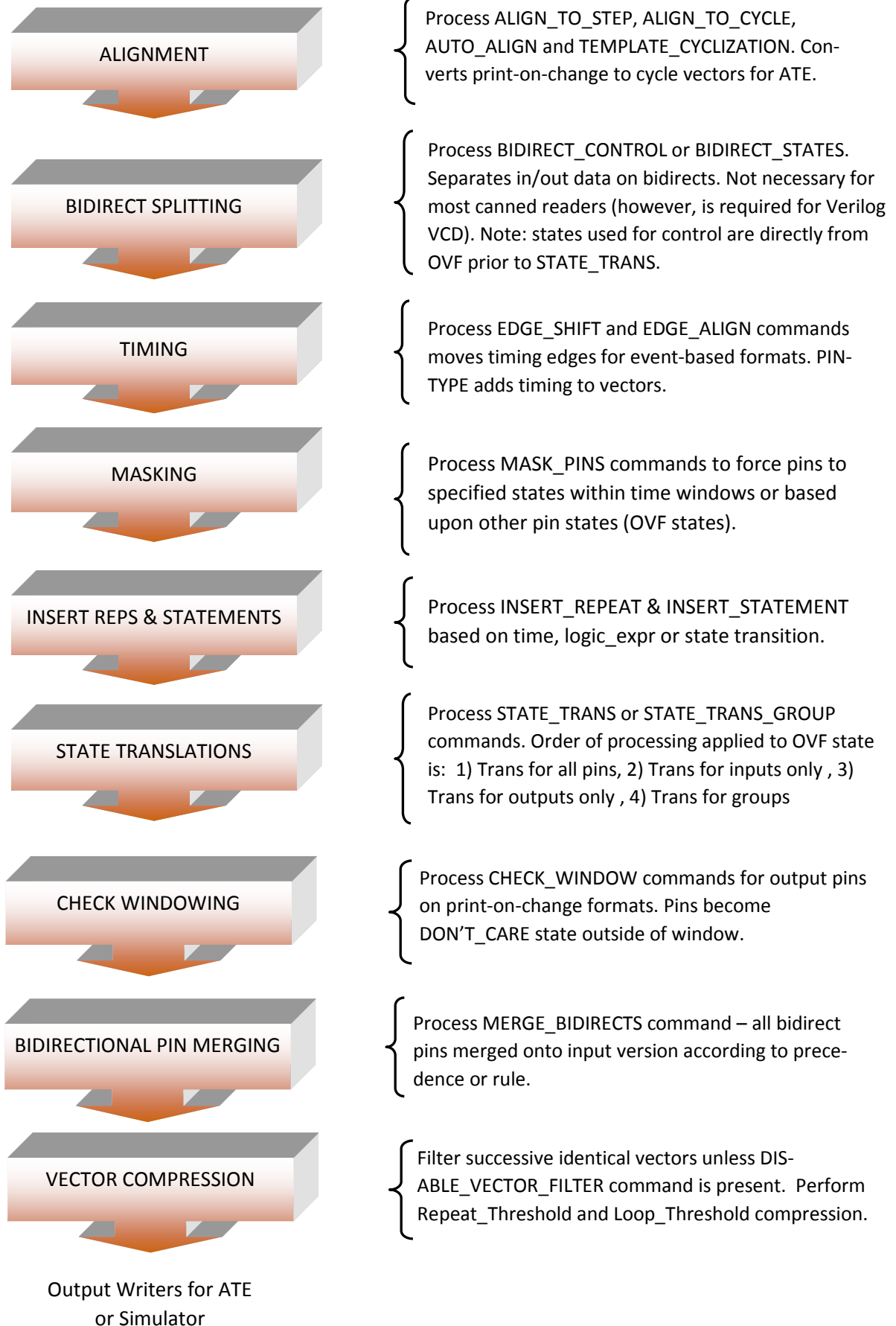
```

The **Read Commands** tell VTRAN the name and format of the input file, the names and directions of signals to be translated (if necessary). If a Canned Reader is not being used, then the user describes the custom format here. The merging of multiple files can also be specified here. The **Processing Commands** are optional and can be used to edit/modify the vector data in various ways. This processing can include adding new signals, masking (conditioning) signals, separating bidirectional data when necessary, modifying timing, filtering, state translations, cyclizing data and many other options. The **Writer Commands** tell VTRAN the desired output format (for simulator or ATE), the output file name(s) and any options associated with the desired output format. In this section you can also specify a new signal order, delete signals, scale the timing and perform other operations on the data.

## Processing Vectors During Translation

In order to be most effective in using VTRAN for desired translations, it is important to know the order in which the different processing steps occur during the translation. The following flow chart shows the order of vector data processing by VTRAN. Note that for any given translation, many of these processes will either be optional or not used at all.





## VTRAN Commands

### OVF BLOCK COMMAND SUMMARY

**AUX\_FILE** [=] "filename";  
Used to specify auxiliary file for some canned readers.

**BEGIN\_LINE** [=] n ;  
Used to define the line number in the OVF file at which VTRAN should begin processing vectors.

**BEGIN\_STRING** [=] "string";  
Used to define a unique text string in the OVF file after which VTRAN should begin processing vectors.

**BIDIRECTS** [=] pin\_list;  
Defines the names and order of pins in the OVF file which are bidirectional.

**BUSFORMAT** radix; or **BUSFORMAT** pin\_list = radix;  
Specifies the radix of busses in the OVF file.

**CASE\_SENSITIVE**;  
Allows there to be more than one signal with the same name spelling but differing only in case of letters in the name.

**COMMENTS** [ON|OFF];  
Enables/Disables the passing of comments from the OVF to the TVF.

**GROUP** n [=] pin\_list;  
Together with the \$gstatesn keyword, it tells VTRAN how the pin states are organized.

**IDDQ** **IDDQfile** = "fname", **keyword** = "kwd", [params]  
Insert IDDQ vectors in STIL or WGL files.

**IGNORE\_UNMATCHED\_NAMES** ;  
Do not terminate when find unmatched names—  
useful when using common command file for multiple file translations.

**INPUTS** [=] pin\_list;  
Defines the names and order of input pins in OVF file.

**MAX\_UNMATCHED** [=] n [verbose];  
Specifies the number of, and information contained in, warnings for lines in the OVF which don't match a format\_string.

**MERGE\_FILE** . . . . . end\_Merge ;

**ORIG\_FILE** [=] "filename";  
Used to specify the OVF file name to be translated.

**OUTPUTS** [=] pin\_list;  
Defines the names and order of output pins in OVF file.

**Q-STAR Product** = "product", [parameters];  
Insert IDDQ vectors specifically for Q-Star hardware.

**SCRIPT\_FORMAT** [=] "format#1" [, . . ."format#n"] ;  
Format descriptors for User-Programmed reader.

**TABULAR\_FORMAT** [=] "format #1" [, . . ."format#n"] ;  
Format descriptors for User-Programmed reader.

**TERMINATE TIME** [=] n; or  
**TERMINATE LINE** [=] m; or  
**TERMINATE STRING** [=] "string";  
Defines where in the OVF to stop processing, at a certain time, line number or when a string is reached.

**WAVE\_FORMAT** [=] "format #1" [, . . ."format#n"] ;  
Format descriptors for User-Programmed reader.

**WHITESPACE** [=] 'a','b','c', . . . , 'n';  
Defines characters in the OVF file that are to be treated as though they are space, (they are ignored).

### PROC BLOCK COMMAND SUMMARY

**ADD\_PIN** pinname = state1 [WHEN expr=state2  
OTHERWISE state3];  
Tells VTRAN to add a new pin to the TVF, and allows you to define the state of this pin.

**ALIGN\_TO\_CYCLE** [-warnings] cycle pin\_list @ time,  
. . . , pin\_list @ time ;  
Vectors can be mapped to a set of cycle data, the state of each pin in a given cycle is determined by its state at a specified strobe time in the OVF.

**ALIGN\_TO\_SIGNAL** [-novector] ref A->B  
SAMPLE=pinlist @ offset;  
Print-on-change vectors are mapped to cycle-based vectors, with each cycle boundary determined by a reference signal transition, and the state of each pin in a given vector determined at the specified strobe time from the start of a cycle.

**ALIGN\_TO\_STEP** [-warnings] step [offset];  
Forces a minimum time resolution in the TVF.

**AUTO\_ALIGN** [-warnings] cycle;  
Collapses print-on-change data in the OVF to cycle data by computing strobe points from information given in the PINTYPE commands.

**BIDIRECT\_CONTROL** pin\_list = dir WHEN expr = state ;  
Separates input data from output data on bidirects under control of a pin state or logical combination of pin states.

**BIDIRECT\_CONTROL** pin\_list = direction @ time ;  
Separates input data from output data on bidirects based upon when the state transitions occur.

**BIDIRECT\_STATES** INPUT state\_list, OUTPUT state\_list ;  
Separates input data from output data on bidirects where unique state characters identify pin direction.

CHECK\_WINDOW pin\_list @ t1, t2, [,t3, t4];  
 pin\_list@TRANSITION ctl, a->b, t1, t2;

CYCLE [=] n;  
 Used to specify the time step between vectors in the OVF when the format of the vectors does not include a time stamp.

CYCLE\_SHIFT [fill\_character=c] pinlist @ n;  
 DISABLE\_VECTOR\_FILTER;  
 Can be used to disable filtering of redundant vectors.

DONT\_CARE 'X';  
 Defines the character state to which output pins should be set outside of their check windows.

EDGE\_ALIGN pinlist @ rtime [,ftime] [xtime];  
 Can modify pin transition times by snapping them to predefined positions within each cycle.

EDGE\_SHIFT pinlist @ rtime [,ftime] [,xtime];  
 Can modify pin transition times by shifting them by fixed amounts.

INSERT\_REPEAT nn @ CONDITION  
 compound\_logic\_expr;  
 Insert repeat count nn opcode on vector when compound\_logic\_expr is true.

INSERT\_REPEAT "pinlist" @ CONDITION  
 Compound\_logic\_expr;  
 Insert repeat count opcode equal to binary value of "pinlist" on vector when compound\_logic\_expr is true.

INSERT\_REPEAT nn @ TIME mm;  
 Insert repeat count nn opcode on vector at time mm.

INSERT\_STATEMENT "statement" @ CONDITION  
 [-before|after], [count n]  
 logic\_expr;  
 | @ TRANSITION [-before|after], [count n]  
 Sig\_name a->b;  
 | @ TIME [-before|after] time;  
 Insert any "statement" text at (before or after) the vector location identified by one of the three mechanisms.

MASK\_PINS [mask\_character='X']  
 [pin\_list]  
 @ t1, t2 [-CYCLE] ; or  
 @ CONDITION expr = state ; or  
 @TRANSITION A>B [m] [,+n]; or  
 @CONTROL\_TRANSITION ctl\_pin A->B [,-m][,+n];  
 Masks the state of specified pins to the mask\_character within the time range between t1 and t2, or when a specified logic conditions exists on other pins or for a number of cycles surrounding the pin's transition.

MASK\_PINS pinlist @ SEQUENCE "input\_sequence",

"output\_sequence" ;

MASK\_PINS pinlist @ SEQUENCE\_BLOCK  
 "input\_sequence", "output\_sequence" ;  
 Modifies the state sequence of pins in pinlist to be "output\_sequence" whenever "input\_sequence" is detected.

MASK\_PINS [mask\_character=x] pinlist @ [NOT]  
 TIMESET <timeset\_name> ;  
 <Incorporate into existing MASK\_PINS command.>

PINTYPE pintype pin\_list @ start1 end1 [start2, end2] ;  
 Defines the behavior and timing to be applied to input and/or output pins during translation.

POIC;  
 Specifies that vectors in the OVF file should be translated to the TVF only when at least 1 input pin has changed in the vector.

REGISTER name [init=1], set @ CONDITION logic\_expr  
 | @ TIME time\_set;  
 Clear @CONDITION logic\_expr  
 | @ TIME time\_clear;  
 Defines a memory element that is set/cleared by a set of logic\_expr conditions. These registers can then be used in compound\_logic\_expr for other statements.

SCALE [=] nn;  
 Used to linearly expand or reduce the time line of the OVF. Happens prior to any timing modifications.

SEPARATE\_TIMING;  
 Tells VTRAN not to incorporate pin timing and behavior into the vectors themselves.

STATE\_TRANS [=] [dir] 'from1'->'to1', . . . ;  
 Defines a mapping from pin states in the OVF file to states in the TVF file.

STATE\_TRANS\_GROUP pin\_list = 'from1'->'to1', . . . ;  
 Supplements the STATE\_TRANS command by providing state translations on an individual pin or group basis.

TEMPLATE\_CYCLIZATION [params] ;

TIME\_OFFSET [=] n ;  
 When reading the vectors from the OVF, the time stamp can be offset by an arbitrary amount.

TIMESET name [default] . . . ENDTIMESET ;  
 Used with TEMPLATE\_CYCLIZATION to specify the timing characteristics of each timeset.

## **TVF BLOCK COMMAND SUMMARY**

ADD\_VTB\_TEXT = "text", Before|After, <Location>;  
 ALIAS ovf\_name = tvf\_name, . . . ; , or  
 "ovf\_string"="tvf\_string";

<p>Provides a way to change the names of pins listed in the OVF, for listing in the TVF.</p> <p>ALIAS_TSET ovf_tsetname=tvf_tsetname, . . . ;</p> <p>Provides a way to change the name of timesets in the OVF when generating the TVF.</p> <p>BIDIRECTS [=] pin_list;</p> <p>Defines the names and order of pins to be listed in the TVF file which are bidirectional.</p> <p>BUSFORMAT radix; or BUSFORMAT pin_list = radix;</p> <p>Specifies the radix of busses in the TVF file. command file for the target simulator, in addition to the vector data file.</p> <p>COMMENTS [PREFACE = “p-string”] [SUFFIX = “s_string”];</p> <p>Used with the User Defined Format writer to specify preface for comments.</p> <p>CREATE_STATISTICS “filename” pinlist [=] [logic0 state_list] , [logic1 state_list] [start nn] [end mm] ;</p> <p>Causes vtran to create a file containing statistics on the states and state transitions of signals in the pinlist between start and end times.</p> <p>DEFINE_HEADER [=] “text string”;</p> <p>Allows you to inhibit the automatic generation of headers and replace it with a custom text string.</p> <p>DELETE_PINS pinlist;</p> <p>Removes the signals in pinlist from the TVF.</p> <p>FORCE_SEQUENTIAL_BUSSES [+   -] ;</p> <p>HEADER [=] n;</p> <p>Causes a vertical list of the pin names to appear as comments in the TVF every n vector lines.</p> <p>INPUTS [=] pin_list ;</p> <p>Defines the names and order of pins to be listed in the TVF file which are inputs.</p> <p>INPUTS_ONLY;</p> <p>Causes only input and the input versions of bidirectional pins to be listed in the TVF.</p> <p>LOWERCASE;</p> <p>Forces all pin names in the TVF to use lower-case letters.</p> <p>MERGE_BIDIRECTS rules = n;</p> <p>Provides for a way to merge the states on bidirectional signals when outputting to the TVF.</p> <p>MERGE_BIDIRECTS state_list;</p> <p>Merges the input and output state information of a bidirectional pin to a single pin after it has been split and processed.</p> <p>MERGE_TSETS TSET_list -&gt; merged_tset_name:</p> <p>Pin1 ‘1’ -&gt; ‘P’, ‘O’ -&gt; ‘N’, @TSET1,</p>	<p>Pin2 ‘1’ -&gt; ‘A’, ‘O’ -&gt; ‘B’ @ TSET2</p> <p>. . .</p> <p>;</p> <p>Can be used to merge several timesets to a single timeset for output to ATE formats which support waveform character timing. Currently these include both STIL and the Verigy 93000.</p> <p>OUTPUTS [=] pin_list ;</p> <p>Defines the names and order of pins to be listed in the TVF file which are outputs.</p> <p>OUTPUTS_ONLY;</p> <p>Causes only output and the output versions of bidirectional pins to be listed in the TVF.</p> <p>PIN_INFO_FILE [=] “filename”, “file_format”;</p> <p>Provides a mechanism to specify signal re-naming and channel number for output formats which use this information.</p> <p>RENAME_BUS_PINS buslist = format;</p> <p>Provides a way of modifying specific bus names in the TVF.</p> <p>RESOLUTION [=] n;</p> <p>Specifies the resolution of time stamps in the output vector file (n = 1.0, 0.1, 0.01 or 0.001).</p> <p>SCALE [=] nn ;</p> <p>Linearly scales all times to TVF.</p> <p>SIMULATOR [=] name [param_list];</p> <p>Defines the target vector file format to be compatible with the simulator named.</p> <p>STOBE_WIDTH [=] n;</p> <p>Used with several of the simulator interfaces to define the width of an output strobe window.</p> <p>SYSTEM_CALL “. . .text . . . “;</p> <p>Upon completion of translating vectors from the OVF to the TVF, VTRAN sends this text string to the system just prior to termination.</p> <p>TARGET_FILE [=] “filename”;</p> <p>Specifies the name of the output Target Vector File.</p> <p>TESTER_FORMAT [=] tester [params];</p> <p>Specifies target tester format and parameters.</p> <p>TITLE [=] “title”;</p> <p>Specifies a special character string to be placed in the header of certain simulator vector files.</p> <p>UPPERCASE;</p> <p>Forces all pin names in the TVF to uppercase letters.</p>
---	---

## **VTRAN Command File Boilerplates**

The following VTRAN command file boilerplates indicate the VTRAN commands which are applicable to the four main translation flows between Cycle-based and Event-based vector formats. Highlighted commands are required or highly recommended, others are optional. Note that the order of processing is dictated by the previous flow charts, not by the order of statements in command file blocks. These boilerplates do not include use of the User-Programmed Reader, only canned readers for VCD/EVCD, WGL, STIL and ATE.

### **Cycle-Based (WGL/STIL) to Cycle-Based (ATE, WGL, STIL) Translations:**

```

OVF_BLOCK
  BEGIN
  ORIG_FILE ...
  COMMENTS ...
  IDDQ ...
  MERGE_FILE ...
  TABULAR_FORMAT ...
  END
PROC_BLOCK
  BEGIN
  ADD_PIN ...
  INSERT_REPEAT ...
  INSERT_STATEMENT ...
  MASK_PINS ...
  REGISTER ...
  STATE_TRANS ...
  STATE_TRANS_GROUP ...
  END
TVF_BLOCK
  BEGIN
  ALIAS ...
  ALIAS_TSET ...
  CREATE_STATISTICS ...
  DELETE_PINS ...
  INPUTS / OUTPUTS / BIDIRECTS ... (if re-
  ordering or deleting)
  HEADER ...
  MERGE_BIDIRECTS ... (to change default)
  MERGE_TSETS ... (Verigy 93000 only)
  RENAME_BUS_PINS ...
  RESOLUTION ...
  SCALE ...
  TARGET_FILE ...
  TESTER_FORMAT ...
  END
END

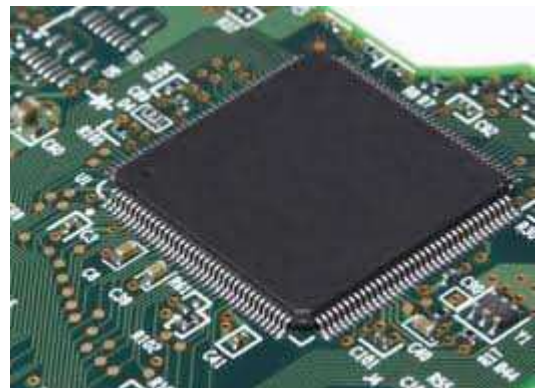
```

### **Cycle-Based (WGL, STIL, ATE) to Event-Based (testbench) Translations:**

```

OVF_BLOCK
  BEGIN
  ORIG_FILE ...
  AUX_FILE ... (for some ATE)
  COMMENTS ...
  MERGE_FILE ...
  TABULAR_FORMAT ...
  END
PROC_BLOCK
  BEGIN
  ADD_PIN ...
  PINTYPE ... (with ADD_PIN only)
  STATE_TRANS ...
  STATE_TRANS_GROUP ...
  END
TVF_BLOCK
  BEGIN
  ALIAS ...
  ADD_VTB_TEXT ...
  DELETE_PINS ...
  RENAME_BUS_PINS ...
  TARGET_FILE ...
  SIMULATOR ...
  END
END

```



**Event-Based (VCD / EVCD) to Cycle-Based (ATE, WGL, STIL) Translations:**

```

OVF_BLOCK
  BEGIN
  INPUTS / OUTPUTS / BIDIRECTS ...
  ORIG_FILE ...
  COMMENTS ...
  MERGE_FILE ...
  SCRIPT_FORMAT ...
  END
PROC_BLOCK
  BEGIN
  ALIGN_TO_CYCLE / ALIGN_TO_STEP /
  AUTO_ALIGN ...
  TEMPLATE_CYCLIZATION ... TIME-
  SET ... (instead of ALIGN..)
  BIDIRECT_CONTROL ... (only if VCD)
  ADD_PIN ...
  TIME_OFFSET ...
  INSERT_REPEAT ...
  INSERT_STATEMENT ...
  MASK_PINS ...
  REGISTER ...
  PINTYPE ...
  STATE_TRANS ...
  STATE_TRANS_GROUP ...
  END
TVF_BLOCK
  BEGIN
  ALIAS ...
  DELETE_PINS ...
  MERGE_BIDIRECTS ... (to change default)
  RENAME_BUS_PINS ...
  SCALE ...
  TARGET_FILE ...
  TESTER_FORMAT ...
  END
END
    
```

**Event-Based (VCD / EVCD) to Event-Based (testbench) Translations:**

```

OVF_BLOCK
  BEGIN
  INPUTS / OUTPUTS / BIDIRECTS ...
  ORIG_FILE ...
  MERGE_FILE ...
  SCRIPT_FORMAT ...
  END
PROC_BLOCK
  BEGIN
  BIDIRECT_CONTROL ... (only if VCD)
  ADD_PIN ...
  PINTYPE ... (for ADD_PIN only)
  STATE_TRANS ...
  STATE_TRANS_GROUP ...
  CHECK_WINDOW ...
  CYCLE ... (if CHECK_WINDOW used)
  EDGE_SHIFT / EDGE_ALIGN ...
  END
TVF_BLOCK
  BEGIN
  ALIAS ...
  ADD_VTB_TEXT ...
  DELETE_PINS ...
  RENAME_BUS_PINS ...
  SCALE ...
  TARGET_FILE ...
  SIMULATOR ...
  END
END
    
```

VTRAN® is a registered trademark of Source III, Inc.



**SOURCE III**

**Source III**

3941 Park Dr. #20-342  
 El Dorado Hills, CA 95762  
 Phone: 916.941.9403  
 FAX: 916.941.9404  
 corp@sourceiii.com



**On the web at  
 Sourceiii.com**