

---

---

# V3 User Manual

LAST UPDATE: 2014/12/05

---

Cheng-Yin Wu

gro070916@yahoo.com.tw

author.v3@gmail.com

Design Verification Lab (DVLab), NTUGIEE

<http://dvlab.ee.ntu.edu.tw>

National Taiwan University (NTU), Taiwan

<http://www.ntu.edu.tw>

A stylized, handwritten-style logo consisting of a large 'V' followed by a subscript '3'.

COPYRIGHT© DESIGN VERIFICATION LAB, 2012-2014

---

---

## ACKNOWLEDGMENT

The author, Cheng-Yin Wu, wants to thank his colleagues in Design Verification Lab (DVLab) and his supervisor Professor Chung-Yang (Ric) Huang for their supports and encouragements during the construction of  $\mathcal{U3}$ , which turns out to be one of the most valuable contributions of his Ph.D. work. Also, he would like to thank friends and anonymous researchers from all over the world who gave precious comments and interests to  $\mathcal{U3}$ . Last but not least, he thanks all organizers and participants of Hardware Model Checking Competition (HWMCC) [1] for their contributions to hardware verification.

## LICENSE OF THIS FRAMEWORK

- \* THIS LICENSE FORBIDS THIS SOFTWARE TO BE USED UNDER ANY COMMERCIAL PURPOSES.
- \* IN ADDITION, USING THIS SOFTWARE WITHOUT EXPLICIT PERMISSIONS IS PROHIBITED.

*$\mathcal{U3}$  : An Extensible Framework for Hardware Verification and Debugging*  
<http://dvlab.ee.ntu.edu.tw/~publication/V3>

Copyright© 2012 - 2014

Design Verification Lab (DVLab), Graduate Institute of Electronics Engineering,  
National Taiwan University (NTU), Taipei, Taiwan. ALL RIGHTS RESERVED.

Permission is hereby granted, free of charge, to use, copy, and modify this software and its documentations for evaluation and research purposes, provided that the above copyright notice and the following content appear in all copies or substantial portions of this software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**Command-line Lexicographic Notations.** Here we give the lexicographic notations of *U3* commands:

- **CAPITAL LETTERS or leading ‘-’:** The leading ‘-’ and capital letters in the command name or parameters are mandatory entries and will be compared *case-insensitively*. The following letters can be partially skipped. However, when entered, they should match the specification *case-insensitively*.
- **Round bracket ():** Meaning it should be replaced by a proper argument as suggested by the (type *variable*) description in the round brackets.
- **Angle brackets < >:** Mandatory parameters; they should appear in the same relative order as specified in the command usage.
- **Square brackets []:** Optional parameters; they can appear anywhere in the command parameters.
- **Or ‘|’:** Or condition; expecting either one of the argument.

**List of Commands** Basically *U3* offers a set of commands that are roughly categorized into the following functions, as listed here:

Functionality	Examples
General	HElP prints command information Quit terminates the program
Design I/O	REAd Aig reads in an AIGER [2, 4] design WRItE Rtl reports design into Verilog [11]
Design Reporting	PRInt NTk reports network information PLOt NTk depicts network topology
Design Synthesis	REWrite NTk performs rule-based rewriting techniques FLAtten NTk flattens a hierarchical network
Design Manipulation	@CD change current network into another
Intent Extraction	EXTract FSM extracts high-level design information from networks
Design Simulation	SIM NTk simulates networks either from given patterns or randomly
Design Verification	SEt PRoPerty creates a property VERify PDR runs PDR [8] algorithm for property checking
<i>U3</i> Model Checking	REAd PRoPerty reads in a set of properties RUN performs multiple property checking under given resource bounds
Design Debugging	SIMplify TRace performs counterexample simplification

The following sections give the descriptions to commands and corresponding parameters. Please note that there is no complete regressions to the current release of *U3*, so good luck! For bug reports or questions to us, please send an e-mail to [author.v3@gmail.com](mailto:author.v3@gmail.com) or report an issue to <https://chengyinwu@bitbucket.org/chengyinwu/v3.git>.

# 1 General Commands

<i>Command:</i>	HElp [<(string cmd) [-Verbose]>]
<i>Synopsis:</i>	Print this help message.
<i>Parameters:</i>	(string cmd): The (partial) name of the command.
	-Verbose: Print usage in more detail.

<b>Command:</b>	D0file <(string <i>fileName</i> )>
<b>Synopsis:</b>	Execute the commands in the dofile.
<b>Parameters:</b>	(string <i>fileName</i> ): The file name of the script.

<i>Command:</i>	<code>HIStory [(int <i>nPrint</i>)]</code>
<i>Synopsis:</i>	Print command history.
<i>Parameters:</i>	<code>(int <i>nPrint</i>)</code> : The number of the latest commands to be printed. (default = $\infty$ )

<i>Command:</i>	SET LOGfile [  -All   -Cmd   -Error   -Warning   -Info   -Debug] <(string <i>fileName</i> )> [  -File-only   -Both] [-APPend]
<i>Synopsis:</i>	Redirect messages to files.
<i>Parameters:</i>	(string <i>fileName</i> ): The file name for message redirection.
-APPend:	Append messages to the output file.
-All:	Enable any types of messages.
-Cmd:	Enable only executed commands.
-Error:	Enable only ERROR messages.
-Warning:	Enable only WARNING messages.
-Info:	Enable only messages for runtime information.
-Debug:	Enable only debug messages.
-File-only:	Disable message output to standard output.
-Both:	Remain message output to standard output.

<i>Command:</i>	USAGE [-Time-only   -Memory-only]
<i>Synopsis:</i>	Report resource usage.
<i>Parameters:</i>	-Time-only: Disable memory usage reporting.
	-Memory-only: Disable time usage reporting.

<i>Command:</i>	Quit [-Force]
<i>Synopsis:</i>	Quit the execution.
<i>Parameters:</i>	-Force: Quit the program forcedly.

## 2 Design I/O Commands

<i>Command:</i>	REAd Aig <(string <i>fileName</i> )> [-Symbol]
<i>Synopsis:</i>	Read AIGER [2, 4] Designs.
<i>Parameters:</i>	(string <i>fileName</i> ): The file name of the input AIGER design.
	-Symbol: Enable reading of symbolic tables.

<i>Command:</i>	REAd BTOR <(string <i>fileName</i> )> [-Symbol]
<i>Synopsis:</i>	Read BTOR [6] Designs.
<i>Parameters:</i>	(string <i>fileName</i> ): The file name of the input BTOR design.
	-Symbol: Enable the reading of symbols (i.e. name of vars).

<i>Command:</i>	REAd Rtl <(string <i>fileName</i> )> [-Filelist] [-FLAtten] [l -QuteRTL   -Primitive]
<i>Synopsis:</i>	Read RTL (Verilog) [11] Designs.
<i>Parameters:</i>	(string <i>fileName</i> ): The file name of the input Verilog design or a list design files.
	-Filelist: Indicate fileName is a list of design files.
	-QuteRTL: Use QuteRTL [13] RTL front-end for design parsing and synthesis.
	-Primitive: Use V3 Primitive RTL front-end for design parsing and synthesis.
	-FLAtten: Flatten the design after parsing. (only available to - QuteRTL)

<i>Command:</i>	WRItE Aig <(string <i>fileName</i> )> [-Symbol]
<i>Synopsis:</i>	Write AIGER [2, 4] Designs.
<i>Parameters:</i>	(string <i>fileName</i> ): The file name for the AIGER output.
	-Symbol: Enable using signal names specified in the input design.

<i>Command:</i>	WRItE Btor <(string <i>fileName</i> )> [-Symbol]
<i>Synopsis:</i>	Write BTOR [6] Designs.
<i>Parameters:</i>	(string <i>fileName</i> ): The file name for the BTOR output.
	-Symbol: Enable using signal names specified in the input design.

<i>Command:</i>	WRItE Rtl <(string <i>fileName</i> )> [-Symbol] [-Initial]
<i>Synopsis:</i>	Write RTL (Verilog) [11] Designs.
<i>Parameters:</i>	(string <i>fileName</i> ): The file name for the RTL output.
	-Symbol: Enable using signal names specified in the input design.
	-Initial: Enable specifying initial states of a network.

### 3 Design Reporting Commands

**Command:** `PRInt Ntk [| -Summary | -Primary | -Verbose | -Netlist |  
-CombLoops | -Floating | -Unreachable]`

**Synopsis:** Print Network Information.

**Parameters:**

- `-Summary:` Print network summary.
- `-Primary:` Print primary ports.
- `-Verbose:` Print statistics of gates.
- `-Netlist:` Print network topology.
- `-CombLoops:` Print combinational loops.
- `-Floating:` Print floating nets.
- `-Unreachable:` Print unreachable nets.

**Command:** `PRInt Net <(unsigned netId)>`

**Synopsis:** Print Net Information.

**Parameters:** `(unsigned netId):` The index of a net to be reported.

**Command:** `PLot Ntk [| -DOT | -PNG | -PS] <(string fileName)>  
<(-Level | -Depth) (unsigned size)> [-Monochrome]`

**Synopsis:** Plot Network Topology.

**Parameters:**

- `(string fileName):` The file name for output.
- `(unsigned size):` The number of levels or depths to be reported.
- `-Monochrome:` Plot with only black and white colors.
- `-Level:` Enable plotting networks under a specified number of levels.
- `-Depth:` Enable plotting networks under a specified number of time-frames.
- `-DOT:` Output into \*.dot files.
- `-PNG:` Output into \*.png files (an executable dot required).
- `-PS:` Output into \*.ps files (an executable dot required).

### 4 Design Synthesis Commands

**Command:** `REDuce Ntk`

**Synopsis:** Perform COI Reduction on Current Network.

**Command:** `STRash Ntk`

**Synopsis:** Perform Structural Hashing on Current Network.

**Command:** `REWrite Ntk`

**Synopsis:** Perform Rule-based Rewriting on Current Network.

**Command:** SET NTKVerbosity [-All] [-REDuce] [-Strash] [-REWrite]  
[-Fwd-map] [-Bwd-map] [-ON | -OFF | -RESET]

**Synopsis:** Set Verbosities for Network Duplication.

**Parameters:**

- RESET: Reset everything to default.
- ON: Turn specified attributes on.
- OFF: Turn specified attributes off.
- All: Toggle all the following attributes.
- REDuce: Toggle COI Reduction. (default = on)
- Strash: Toggle Structural Hashing. (default = off)
- REWrite: Toggle Rule-based Rewriting. (default = off)
- Fwd-map: Toggle Preservation of Forward (to Sucessor) ID Maps. (default = on)
- Bwd-map: Toggle Preservation of Backward (to Ancestor) ID Maps. (default = on)

**Command:** PRInt NTKVerbosity

**Synopsis:** Print Verbosities for Network Duplication.

**Command:** DUPlicate Ntk

**Synopsis:** Duplicate Current Ntk from Verbosity Settings.

**Command:** BLAst Ntk [-Primary]

**Synopsis:** Bit-blast Word-level Networks into Boolean-level Networks.

**Parameters:**

- Primary: Bit-blast primary inputs, inouts, and latches only.

**Command:** EXPand Ntk <(unsigned *cycle*)> [-Initial]

**Synopsis:** Perform Time-frame Expansion for Networks.

**Parameters:**

- (unsigned *cycle*): The number of time-frames to be considered.
- Initial: Set initial state values at the first timeframe.

**Command:** FLatten Ntk [(unsigned *level*)]

**Synopsis:** Flatten Hierarchical Networks.

**Parameters:**

- (unsigned *level*): The number of hierarchical levels to be flattened. (default =  $\infty$ )

**Command:** MITer Ntk <(unsigned *ntkId1*)> <(unsigned *ntkId2*)> [-Merge]  
[-Name <(string *miterNtkName*)>] [| -SEC | -CEC]

**Synopsis:** Miter Two Networks.

**Parameters:**

- (unsigned *ntkId1*): The index of the first network.
- ((unsigned *ntkId2*): The index of the second network.
- (string *miterNtkName*): The name of the resulting miter.
- Name: Indicate the following token is the name of the miter.
- SEC: Construct miter for sequential equivalence checking.
- CEC: Construct miter for combinational equivalence checking. Please notice that latches are mapped by their names.
- Merge: Merge miter outputs into a representative.

## 5 Design Manipulation Commands

<i>Command:</i>	@CD [/   -   .   ..   (Path Format)]
<i>Synopsis:</i>	Change Design for Current Network.
<i>Parameters:</i>	Path Format: [/(unsigned <i>ntkID</i> )] [/(unsigned <i>subModuleIndex</i> )]*

<i>Command:</i>	@LN <(unsigned <i>ntkID</i> )> <(unsigned <i>subModuleIndex</i> )>
<i>Synopsis:</i>	Link a Network with an instance of Current Network.
<i>Parameters:</i>	(unsigned <i>ntkID</i> ): The index of a network. (unsigned <i>subModuleIndex</i> ): The index of an instance of current network.

<i>Command:</i>	@LS [(unsigned <i>level</i> )]
<i>Synopsis:</i>	List Network Instances of Current Network.
<i>Parameters:</i>	(unsigned <i>level</i> ): The number of levels of instances to be printed. (default = 1)

## 6 Intent Extraction Commands

<i>Command:</i>	EXtract FSM [-Name <(string <i>fsmName</i> )>] [-Output <(string <i>outputIndex</i> )>] [-Time <(unsigned <i>maxTime</i> )>] [-SCC   -NONE] [-CONFIRM]
<i>Synopsis:</i>	Extract Finite State Machines from Current Network.
<i>Parameters:</i>	(string <i>fsmName</i> ): The name of FSMs to be extracted. (string <i>outputIndex</i> ): The index of output that represents the bad signal of a safety property. (unsigned <i>maxTime</i> ): The runtime limit for extraction. -Time: Indicate that the following token is the maximum run-time limit. -Output: Indicate that the following token is the index of an output. -Name: Indicate that the following token is the name for FSMs. -SCC: Cluster variables by strongly connected components. -NONE: Disable clustering of variables. -CONFIRM: Self checking if the extraction is successful.



<i>Command:</i>	ELaborate FSM [-Input (string <i>inputName</i> )] [-Name <(string <i>fsmName</i> )>] [-Output <(string <i>outputIndex</i> )>] [-CONFIRM]
<i>Synopsis:</i>	Elaborate Network and Construct FSM from Input Specification.
<i>Parameters:</i>	(string <i>inputName</i> ): The name of FSM specification input file. (string <i>fsmName</i> ): The name of FSMs to be extracted. (string <i>outputIndex</i> ): The index of output that represents the bad signal of a safety property. -Input: Indicate that the following token is the name for the input file. -Output: Indicate that the following token is the index of an output. -Name: Indicate that the following token is the name for FSMs. -CONFIRM: Self checking if the extraction is successful.

<i>Command:</i>	PLOT FSM <(string <i>fsmName</i> )> <(string <i>fsmDirName</i> )>
<i>Synopsis:</i>	Plot Finite State Machines into *.png files.
<i>Parameters:</i>	(string <i>fsmName</i> ): The name of FSMs to be plotted. (string <i>fsmDirName</i> ): The name of a directory for FSM outputs.

<i>Command:</i>	WRite FSM <(string <i>fsmName</i> )> <(string <i>outputFile</i> )>
<i>Synopsis:</i>	Output Finite State Machines Specifications.
<i>Parameters:</i>	(string <i>fsmName</i> ): The name of FSMs to be plotted. (string <i>outputFile</i> ): The name of a file for FSM outputs.

## 7 Design Simulation Commands

<i>Command:</i>	SIM NTk <(-Input <(string <i>fileName</i> )>)   (-Random <(unsigned <i>patterns</i> )>)> [-Output <(string <i>outFileName</i> )>] [-Event]
<i>Synopsis:</i>	Plot simulation or counterexample traces.
<i>Parameters:</i>	(string <i>fileName</i> ): The file name of the input pattern file. (unsigned <i>patterns</i> ): The number of patterns for random simulation. (string <i>outFileName</i> ): The file name for simulation result output. -Event: Enable event-driven simulation. -Input: Enable simulation from input patterns. -Random: Enable random simulation. -Output: Enable dumping simulation results into a file.

<i>Command:</i>	<code>PLot TRace &lt;(string <i>inputPatternFileName</i>)&gt;</code> <code>&lt;(string <i>outputFileName</i>)&gt;</code>
<i>Synopsis:</i>	Plot simulation or counterexample traces.
<i>Parameters:</i>	<code>(string <i>inputPatternFileName</i>):</code> The file name of the input pattern file. <code>(string <i>outputFileName</i>):</code> The file name for simulation result output.

## 8 Design Verification Commands

<i>Command:</i>	<code>SET SAFETy [-Name &lt;(string <i>propertyName</i>)&gt;]</code> <code>[(unsigned <i>outputIndex</i>)]</code> <code>[-INVARIANT &lt;(string <i>invName</i>)*&gt;]</code> <code>[-INVConstraint &lt;(string <i>constrName</i>)*&gt;]</code>
<i>Synopsis:</i>	Set Safety Properties on Current Network.
<i>Parameters:</i>	<code>(string <i>propertyName</i>):</code> The name of a property to be set. <code>(unsigned <i>outputIndex</i>):</code> The index of a primary output serving as a bad signal. <code>(string <i>invName</i>)*:</code> List of names of invariants. <code>(string <i>constrName</i>)*:</code> List of names of (either invariant or fairness) constraints. <code>-Name:</code> Indicate the following token is the name of a property. <code>-INVARIANT:</code> Indicate the starting of a list of invariants. <code>-INVConstraint:</code> Indicate the starting of a list of invariant constraints.

<i>Command:</i>	<code>SET LIVEness [-Name &lt;(string <i>propertyName</i>)&gt;]</code> <code>[-INVARIANT &lt;(string <i>invName</i>)*&gt;]</code> <code>[-INVConstraint &lt;(string <i>constrName</i>)*&gt;]</code> <code>[-FAIRnessConstraint &lt;(string <i>constrName</i>)*&gt;]</code>
<i>Synopsis:</i>	Set Liveness Properties on Current Network.
<i>Parameters:</i>	<code>(string <i>propertyName</i>):</code> The name of a property to be set. <code>(string <i>invName</i>)*:</code> List of names of invariants. <code>(string <i>constrName</i>)*:</code> List of names of (either invariant or fairness) constraints. <code>-Name:</code> Indicate the following token is the name of a property. <code>-INVARIANT:</code> Indicate the starting of a list of invariants. <code>-INVConstraint:</code> Indicate the starting of a list of invariant constraints. <code>-FAIRnessConstraint:</code> Indicate the starting of a list of fairness constraints.

<i>Command:</i>	<code>ELaborate PROperty [(string <i>propertyName</i>)]*</code>
<i>Synopsis:</i>	Elaborate Properties on a Duplicated Network.
<i>Parameters:</i>	<code>(string <i>propertyName</i>):</code> The name of a property.

<i>Command:</i>	SET REport [-All] [-RESUlt] [-Endline] [-Solver] [-Usage] [-ON   -OFF   -RESET]
<i>Synopsis:</i>	Set Verbosities for Verification Report.
<i>Parameters:</i>	-RESET: Reset everything to default. -ON Turn specified attributes on. -OFF Turn specified attributes off. -All Toggle all the following attributes. -RESUlt Toggle interactive verification status. (default = on) -Endline Toggle endline or carriage return. (default = off) -Solver Toggle solver information. (default = off) -Usage Toggle verification time usage. (default = on) -Profile Toggle checker specific profiling. (default = off)

<i>Command:</i>	PRInt REport
<i>Synopsis:</i>	Print Verbosities for Verification Report.

<i>Command:</i>	SET SOLver [  -Default   -Minisat   -Boolector]
<i>Synopsis:</i>	Set Active Solver for Verification.
<i>Parameters:</i>	-Default: Enable default solver. (default = minisat) -Minisat Enable MiniSat as the active solver. -Boolector Enable Boolector as the active solver.

<i>Command:</i>	PRInt SOLver
<i>Synopsis:</i>	Print Active Solver for Verification.

<i>Command:</i>	VERify SIM [(string <i>propertyName</i> )] [<-Time (unsigned <i>MaxTime</i> )>] [<-Cycle (unsigned <i>MaxCycle</i> )>]
<i>Synopsis:</i>	Perform (Constrained) Random Simulation.
<i>Parameters:</i>	(string <i>propertyName</i> ): The name of a property to be verified. (unsigned <i>MaxTime</i> ): The upper bound of simulation runtime. (unsigned <i>MaxCycle</i> ): The upper bound of simulation cycle. -Time: Enable setting of runtime limit. -Cycle: Enable setting of cycle limit.

<i>Command:</i>	VERify BMC [(string <i>propertyName</i> )] [-Max-depth (unsigned <i>MaxDepth</i> )] [-Pre-depth (unsigned <i>PreDepth</i> )] [-Inc-depth (unsigned <i>IncDepth</i> )]
<i>Synopsis:</i>	Perform Bounded Model Checking [3].
<i>Parameters:</i>	(string <i>propertyName</i> ): The name of a property to be verified. (unsigned <i>MaxDepth</i> ): The upper bound of time-frames to be reached. (default = 100) (unsigned <i>PreDepth</i> ): The number of frames at initial. (default = 0) (unsigned <i>IncDepth</i> ): The number of frames to be increased in each iteration. (default = 1) -Max-depth: Indicate the following token is the time-frame limit. -Pre-depth: Indicate the following token is the number of frames at initial. -Inc-depth: Indicate the following token is the number of frames to be increased.

<i>Command:</i>	VERify UMC [(string <i>propertyName</i> )] [-Max-depth (unsigned <i>MaxDepth</i> )] [-Pre-depth (unsigned <i>PreDepth</i> )] [-Inc-depth (unsigned <i>IncDepth</i> )] [-NOProve   -NOFire] [-Uniqueness]
<i>Synopsis:</i>	Perform Unbounded Model Checking [9].
<i>Parameters:</i>	(string <i>propertyName</i> ): The name of a property to be verified. (unsigned <i>MaxDepth</i> ): The upper bound of time-frames to be reached. (default = 100) (unsigned <i>PreDepth</i> ): The number of frames at initial. (default = 0) (unsigned <i>IncDepth</i> ): The number of frames to be increased in each iteration. (default = 1) -Max-depth: Indicate the following token is the time-frame limit. -Pre-depth: Indicate the following token is the number of frames at initial. -Inc-depth: Indicate the following token is the number of frames to be increased. -NOProve: Disable running k-induction in UMC. -NOFire: Disable performing bounded model checking (BMC) in UMC. -Uniqueness: Enable adding uniqueness constraints.

<i>Command:</i>	VERify ITP [(string <i>propertyName</i> )] [-Max-depth (unsigned <i>MaxDepth</i> )] [-Reverse] [-Increment] [-Force] [-RECYcle] [-Block (unsigned <i>badCount</i> )]
<i>Synopsis:</i>	Perform Interpolation-based Model Checking Algorithm <i>NewITP</i> [12].
<i>Parameters:</i>	(string <i>propertyName</i> ): The name of a property to be verified. (unsigned <i>MaxDepth</i> ): The upper bound of time-frames to be reached. (default = 100) (unsigned <i>badCount</i> ): The maximum number of spurious cex for refinement. (default = 1) -Max-depth: Indicate the following token is the limit of time-frames. -Block: Indicate the following token is the limit to cex analysis. -Reverse: Enables the reversed implementation of <i>NewITP</i> . -Increment: Enables incrementing BMC depth dynamically. -Force: Enables considering 1 k frames (instead of the k-th frame) in the BMC part. -RECYcle: Enables cube recycling for interpolant reuse.

<i>Command:</i>	VERify PDR [(string <i>propertyName</i> )] [-Max-depth (unsigned <i>MaxDepth</i> )] [-Recycle (unsigned <i>MaxCount</i> )] [-Incremental]
<i>Synopsis:</i>	Perform Property Directed Reachability [5, 8].
<i>Parameters:</i>	(string <i>propertyName</i> ): The name of a property to be verified. (unsigned <i>MaxDepth</i> ): The upper bound of time-frames to be reached. (default = 100) (unsigned <i>MaxCount</i> ): The upper bound of temporary assumption literals in solvers. (default = $\infty$ ) -Max-depth: Indicate the following token is the limit of time-frames. -Recycle: Enable setting the limit of assumption literals for recycle. -Incremental: Implement with multiple solvers. (c.f. Monolithic)

<b>Command:</b>	VERify SEC [(string <i>propertyName</i> )] [-Max-depth (unsigned <i>MaxDepth</i> )] [-UMC   -IPDR   -MPDR] [-CEC] [-SEC]
<b>Synopsis:</b>	Perform Sequential Equivalence Checking [10].
<b>Parameters:</b>	(string <i>propertyName</i> ): The name of a property to be verified. (unsigned <i>MaxDepth</i> ): The upper bound of time-frames to be reached. (default = 100) -Max-depth: Indicate the following token is the time-frame limit. -UMC: Enable UMC as a safety checker. -MPDR: Enable Monolithic PDR as a safety checker. -IPDR: Enable Incremental PDR as a safety checker. -CEC: Assume that the Network could be a CEC Miter. -SEC: Assume that the Network could be a SEC Miter.

<b>Command:</b>	VERify KLIVE [(string <i>propertyName</i> )] [-Max-depth (unsigned <i>MaxDepth</i> )] [-Inc-depth (unsigned <i>IncDepth</i> )] [ -UMC   -IPDR   -MPDR]
<b>Synopsis:</b>	Perform K-Liveness [7] for Liveness Checking.
<b>Parameters:</b>	(string <i>propertyName</i> ): The name of a property to be verified. (unsigned <i>MaxDepth</i> ): The upper bound of time-frames to be reached. (default = 100) (unsigned <i>IncDepth</i> ): The number of frames to be increased in each iteration. (default = 1) -Max-depth: Indicate the following token is the time-frame limit. -Inc-depth: Indicate the following token is the number of frames to be increased. -UMC: Enable UMC as a safety checker. -MPDR: Enable Monolithic PDR as a safety checker. -IPDR: Enable Incremental PDR as a safety checker.

<b>Command:</b>	CHEck REsult <(string <i>propertyName</i> )> [-Simulation   -Formal] [[-Trace   -Invariant] <(string <i>resultFileName</i> )>]
<b>Synopsis:</b>	Verify Verification Result. Note: Confirmation of Inductive Invariants is Not Available Yet !!
<b>Parameters:</b>	(string <i>propertyName</i> ): The name of a verified property. (string <i>resultFileName</i> ): The file name of a verification result. -Simulation: Enable simulation in verifying the result. -Formal: Enable formal in verifying the result. -Trace: Indicate resultFileName is a file of counterexample. -Invariant: Indicate resultFileName is a file of inductive invariant.

<b>Command:</b>	<code>PLOt REsult &lt;(string <i>propertyName</i>)&gt; &lt;(string <i>resultFileName</i>)&gt;</code>
<b>Synopsis:</b>	Elaborate Properties on a Duplicated Network.
<b>Parameters:</b>	<p><code>(string <i>propertyName</i>):</code>      The name of a verified property.</p> <p><code>(string <i>resultFileName</i>):</code>    The file name for the output of verification results.</p>

<b>Command:</b>	<code>WRite REsult &lt;(string <i>propertyName</i>)&gt; &lt;(string <i>resultFileName</i>)&gt;</code>
<b>Synopsis:</b>	Write Verification Results.
<b>Parameters:</b>	<p><code>(string <i>propertyName</i>):</code>      The name of a verified property.</p> <p><code>(string <i>resultFileName</i>):</code>    The file name for the output of verification results.</p>

## 9 V3 Model Checking Commands

<b>Command:</b>	<code>READ PROperty &lt;(string <i>fileName</i>)&gt; &lt;-Aiger   -Prop&gt;</code>
<b>Synopsis:</b>	Read property specification from external file.
<b>Parameters:</b>	<p><code>(string <i>fileName</i>):</code>      The file name of the property input.</p> <p><code>-Aiger:</code>                      Indicate <i>fileName</i> is an AIGER input.</p> <p><code>-Prop:</code>                        Indicate <i>fileName</i> is a PROP input (build on top of current ntk).</p>

<b>Command:</b>	<code>WRITE PROperty &lt;(string <i>fileName</i>)&gt; &lt;-Aiger   -Prop&gt;</code>
<b>Synopsis:</b>	Write property specification into file.
<b>Parameters:</b>	<p><code>(string <i>fileName</i>):</code>      The file name of the property output.</p> <p><code>-Aiger:</code>                      Output network and properties into AIGER format.</p> <p><code>-Prop:</code>                        Output properties into PROP format (in terms of current network).</p>

<b>Command:</b>	<code>RUN &lt;-TIMEout (unsigned <i>maxTime</i>)&gt;</code> <code>          &lt;-MEMoryout (unsigned <i>maxMemory</i>)&gt;</code> <code>          &lt;-THReadout (unsigned <i>maxThread</i>)&gt;</code>
<b>Synopsis:</b>	Run <i>U3</i> Model Checker. (see also my PhD thesis for detailed descriptions)
<b>Parameters:</b>	<p><code>(unsigned <i>maxTime</i>):</code>      Wall Timeout limit in seconds.</p> <p><code>(unsigned <i>maxMemory</i>):</code>    Memoryout limit in Mega Bytes.</p> <p><code>(unsigned <i>maxThread</i>):</code>    The number of available CPU cores.</p> <p><code>-TIMEout:</code>                      Indicate the next number is the timeout limit.</p> <p><code>-MEMoryout:</code>                    Indicate the next number is the memoryout limit.</p> <p><code>-THReadout:</code>                    Indicate the next number is the CPU core limit.</p>

## 10 Design Debugging Commands

<i>Command:</i>	OPTimize TRace <(string propertyName)> [-NOReduce   -NOGeneralize]
<i>Synopsis:</i>	Optimize a Counterexample Trace.
<i>Parameters:</i>	(string <i>propertyName</i> ): The name of a failing property. -NOReduce: Disable counterexample reduction. -NOGeneralize: Disable counterexample generalization.

<i>Command:</i>	SIMplify TRace <(string propertyName)> [(unsigned maxNoFrames)] [  -Care   -Transition]
<i>Synopsis:</i>	Simplify Counterexample Traces.
<i>Parameters:</i>	(string <i>propertyName</i> ): The name of a fired property. (unsigned maxNoFrames): Upper bound frame numbers in a sub-problem. (default = $\infty$ ) (string <i>constrName</i> )*: List of names of (either invariant or fairness) constraints. -Care: Start the configuration of minimizing care signals. -Transition: Start the configuration of minimizing transitions.



## References

- [1] Hardware Model Checking Competition. <http://fmv.jku.at/hwmcc/>.
- [2] A. Biere. The AIGER And-Inverter Graph (AIG) Format. Available at [fmv.jku.at/aiger](http://fmv.jku.at/aiger/), 2007.
- [3] A. Biere, A. Cimatti, E. Clarke, and Y. Zhu. Symbolic Model Checking without BDDs. *Tools and Algorithms for the Construction and Analysis of Systems*, pages 193–207, 1999.
- [4] A. Biere, K. Heljanko, and S. Wieringa. AIGER 1.9 and Beyond. Available at [fmv.jku.at/hwmcc11/beyond1.pdf](http://fmv.jku.at/hwmcc11/beyond1.pdf), 2011.
- [5] A. Bradley. SAT-based Model Checking without Unrolling. In *Verification, Model Checking, and Abstract Interpretation*, pages 70–87. Springer, 2011.
- [6] R. Brummayer, A. Biere, and F. Lonsing. BTOR: Bit-precise Modelling of Word-level Problems for Model Checking. In *Proceedings of the Joint Workshops of the 6th International Workshop on Satisfiability Modulo Theories and 1st International Workshop on Bit-Precise Reasoning*, pages 33–38. ACM, 2008.
- [7] K. Claessen and N. Sörensson. A Liveness Checking Algorithm that Counts. In *FMCAD*, pages 52–59, 2012.
- [8] N. Een, A. Mishchenko, and R. Brayton. Efficient Implementation of Property Directed Reachability. In *Formal Methods in Computer-Aided Design (FMCAD), 2011*, pages 125–134. IEEE, 2011.
- [9] N. Eén and N. Sörensson. Temporal Induction by Incremental SAT Solving. *Electronic Notes in Theoretical Computer Science*, 89(4):543–560, 2003.
- [10] A. Mishchenko, M. Case, R. Brayton, and S. Jang. Scalable and Scalably-verifiable Sequential Synthesis. In *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, pages 234–241. IEEE, 2008.
- [11] D. Thomas and P. Moorby. *The Verilog® Hardware Description Language*, volume 2. Springer, 2002.
- [12] C.-Y. Wu, C.-A. Wu, C.-Y. Lai, and C.-Y. Huang. A Counterexample-guided Interpolant Generation Algorithm for SAT-based Model Checking. In *Design Automation Conference (DAC), 2013 50th ACM/EDAC/IEEE*, pages 1–6. IEEE, 2013.
- [13] H.-H. Yeh, C.-Y. Wu, and C.-Y. Huang. QuteRTL: Towards an Open Source Framework for RTL Design Synthesis and Verification. *Tools and Algorithms for the Construction and Analysis of Systems*, pages 377–391, 2012.