

---

## Quick Reference: *NAPA 4.00* instructions

<b>#</b>	<any comment on one line>	NB: <i>not compatible with the MAC macroprocessor</i>
<b>#*</b>	<any comment on one line>	NB: <i>preferred</i>
<b>alias</b>	<alias_name> <target_node> <alias_name> <target_variable>	
<b>array</b>	(analog) <array_name> ['<array_size>'] [<"file_pathname">] (digital) <array_name> ['<array_size>'] [<"file_pathname">] (pointer) <array_name> ['<array_size>'] <node_name   variable_name   array_name...> (pointer) <array_name> [''] <node_name   variable_name   array_name...>	
<b>assert</b>	<"text_message"> <C_Boolean_expression>	
<b>call</b>	void <C_function_returning_void> <return_value> <C_expression>	
<b>command_line</b>	<variable_name...>   fs   ts   void	
<b>comment</b>	<"text_message">	
<b>data</b>	<"file_name"> <parameters...>	
<b>debug</b>	[<debug_level_number   identifier...>]	
<b>decimate</b>	[fs] <decimation_factor> [<decimation_initial_value>]	
<b>declare</b>	(analog) <identifier...> (digital) <identifier...> (char) <identifier...> ( ) <identifier...> (constant) <identifier...> ( true) <a_function_returning_a_boolean...>	
<b>directive</b>	<C_preprocessor_macro_identifier> [<value>]	
<b>drop</b>	[fs] <C_Boolean_expression>	
<b>dump</b>	<"file_pathname"> [when <C_Boolean_expression_of_events>]	
<b>dvar</b>	<variable_name> [<initial_value>] [&update   &constant] [&export]	
<b>error</b>	<"text_message">	
<b>event</b>	<event_name> [<C_expression_returning_a_boolean>] <event_name> (new) [<C_expression>]	
<b>export</b>	<global_variable_name> <node_name> <variable_name>	
<b>format</b>	(analog) <"C_double_output_format">   S   M   L (digital) <"C_long_long_output_format">   S   M   L (string) <"C_string_output_format">   S   M   L	
<b>ganging</b>	<array_name> ['<array_size>'] <nod_nam   var_nam   number   string   array_nam...>	

	<array_name> ‘[ ]’ <nod_nam   var_nam   number   array_nam...>
<b>fs</b>	[<sampling_frequency>]
<b>gateway</b>	[<count_down>]
<b>header</b>	<“file_pathname”> [(noexpand)] <“file_pathname”> (expand)
<b>init</b>	void <C_function_returning_void> <variable_name> <C_function> <node_name> <C_expression>
<b>inject</b>	<node_name> <C_function>
<b>input</b>	<“file_pathname”> <variable_name...> “stdin“ <variable_name...>
<b>interface</b>	<\$node   \$variable   \$parm...>
<b>cell interface</b>	<\$node   \$variable   \$parm...>
<b>data interface</b>	<\$variable   \$parm...>
<b>interpolate</b>	[fs] <interpolation_factor>
<b>ivar</b>	<variable_name> [<“initial_value”>] [&update   &constant] [&export]
<b>load</b>	<“file_pathname”>
<b>napa_version</b>	<version_id>
<b>node</b>	<node_name> <node_kind> <node_name   variable_name   parameter...> void <node_kind> <node_name   variable_name   parameter...>
<b>nominal</b>	fs
<b>num_initial</b>	<number_of_initial_samples>
<b>opcode</b>	<alu_name> <opcode_number> [<template>]
<b>output</b>	<“path_nam”> <node_nam   var_nam...> [when <boolean_expression_of_events>] <string_var_nam> <node_nam   var_nam...> [when <boolean_expression_of_events>] “stderr“ <node_nam   var_nam...> [when <boolean_expression_of_events>] “stdout“ <node_nam   var_nam...> [when <boolean_expression_of_events>]
<b>ping</b>	[ “stderr” ] <“path_nam”> <string_var_nam>
<b>post</b>	[<label>] <function_id> <file_name> [<parameters>] <label> void <file_name>
<b>random_seed</b>	<[-   +] seed_number>
<b>restart</b>	
<b>string</b>	<variable_name> [<“initial_value”>]
<b>stuck</b>	<node_name> <C_expression_returning_a_number>
<b>synchronize</b>	(yes)   (no)
<b>terminate</b>	[ <C_Boolean_expression> ]
<b>title</b>	<“some one-line text”>
<b>tool</b>	<user_defined_itool> <parameters>

<b>ts</b>	[<sampling_period>]
<b>update</b>	<variable_name> [<C_expression>] [when <boolean_expression_of_events>]
<b>void</b>	<file_name>
<b>warning</b>	<"text_message">

---

## Quick Reference: NAPA 4.00 nodes

<b>adc</b>	<num_level> <input_node> <reference_node>	$R \rightarrow I$
<b>algebra</b>	<C_expression>	<i>Chameleonic</i>
<b>alu</b>	<alu_name> <opcode node> <input_node...>	<i>Chameleonic</i>
<b>and</b>	<input_node> <input_node...>	$I \rightarrow I$
<b>average</b>	<[-   +]input_node> <[-   +]input_node...>	$R \rightarrow R$
<b>bshift</b>	<number> <input_node>	$I \rightarrow I$
	<[-   +]shift_variable> <input_node>	$I \rightarrow I$
	<[-   +]shift_node> <input_node>	$I \rightarrow I$
<b>btoi</b>	<input_node> <input_node...>	$I \rightarrow I$
<b>buffer</b>	<input_node>	$I \rightarrow I$
<b>bwand</b>	<hexavalued_mask> <input_node...>	$I \rightarrow I$
	<input_node> <input_node...>	$I \rightarrow I$
<b>bwbuffer</b>	<input_node>	$I \rightarrow I$
<b>bwinv</b>	<input_node>	$I \rightarrow I$
<b>bwnand</b>	<hexavalued_mask> <input_node...>	$I \rightarrow I$
	<input_node> <input_node...>	$I \rightarrow I$
<b>bwnor</b>	<hexavalued_mask> <input_node...>	$I \rightarrow I$
	<input_node> <input_node...>	$I \rightarrow I$
<b>bwnot</b>	<input_node>	$I \rightarrow I$
<b>bwor</b>	<hexavalued_mask> <input_node...>	$I \rightarrow I$
	<input_node> <input_node...>	$I \rightarrow I$
<b>bwxnor</b>	<hexavalued_mask> <input_node...>	$I \rightarrow I$
	<input_node> <input_node...>	$I \rightarrow I$
<b>bwxor</b>	<hexavalued_mask> <input_node...>	$I \rightarrow I$
	<input_node> <input_node...>	$I \rightarrow I$
<b>cell</b>	<instance_name> <"file_name"> <parameter_list>	<i>N/A</i>
<b>change</b>	<input_node>	$X \rightarrow I$
<b>clip</b>	<[-   +]threshold_low> <[-   +]threshold_high> <input_node>	<i>Chameleonic</i>
<b>clock</b>	<["pattern_descriptor_aperiodic".] "pattern_descriptor_periodic">	$\rightarrow I$
<b>comp</b>	<[-   +]positive_input_node> <[-   +]negative_input_node>	$X \rightarrow I$
	<[-   +]positive_input_node> <[-   +]variable>	$X \rightarrow I$
	<[-   +]variable> <[-   +]negative_input_node>	$X \rightarrow I$
	<[-   +]positive_input_node> <[-   +]number>	$X \rightarrow I$
	<[-   +]number> <[-   +]negative_input_node>	$X \rightarrow I$

<b>const</b>	[(digital)] <C_expression> (analog) <C_expression>	$\rightarrow I$ $\rightarrow R$
<b>copy</b>	<[-   +]input_node>	<i>Chameleonic</i>
<b>cosine</b>	<[-   +]offset> <amplitude> <frequency> <[-   +]phase>	$X \rightarrow R$
<b>dac</b>	<num_level> <input_node> <reference_node>	$I \rightarrow R$
<b>dalgebra</b>	<C_expression_cast_to_real>	$X \rightarrow R$
<b>dc</b>	[(analog)] <C_expression> (digital) <C_expression>	$\rightarrow R$ $\rightarrow I$
<b>delay</b>	<[-   +]input_node> <number> <[-   +]input_node> <delays_var> <[-   +]input_node>	<i>Chameleonic</i> <i>Chameleonic</i> <i>Chameleonic</i>
<b>differentiator</b>	<[-   +]input_node>	<i>Chameleonic</i>
<b>div</b>	<[-   +]input_node> <[-   +]input_node> <[-   +]input_node> <[-   +]variable> <[-   +]input_node> <[-   +]number>	<i>Chameleonic</i> <i>Chameleonic</i> <i>Chameleonic</i>
<b>dto</b>	<input_node>	$R \rightarrow I$
<b>dtool</b>	<dtool_name> [<parameter_list>]	$X \rightarrow R$
<b>duser</b>	<duser_name> [<parameter_list>]	$X \rightarrow R$
<b>equal</b>	<[-   +]input_node> <[-   +]input_node> <[-   +]input_node> <[-   +]variable> <[-   +]variable> <[-   +]input_node> <[-   +]input_node> <[-   +]number> <[-   +]number> <[-   +]input_node>	$X \rightarrow I$ $X \rightarrow I$ $X \rightarrow I$ $X \rightarrow I$ $X \rightarrow I$
<b>fzand</b>	<input_node> <input_node...>	$R \rightarrow R$
<b>fzbuffer</b>	<input_node> <input_node>	$R \rightarrow R$
<b>fzinv</b>	<input_node>	$R \rightarrow R$
<b>fzor</b>	<input_node> <input_node...>	$R \rightarrow R$
<b>fznand</b>	<input_node> <input_node...>	$R \rightarrow R$
<b>fznor</b>	<input_node> <input_node...>	$R \rightarrow R$
<b>fznot</b>	<input_node>	$R \rightarrow R$
<b>gain</b>	<[-   +]number> <input_node> <[-   +]gain_variable> <input_node>	<i>Chameleonic</i> <i>Chameleonic</i>
<b>generator</b>	<instance_name> <"generator_name"> <parameter_list>	<i>N/A</i>
<b>hold</b>	<control_node> <input_node>	<i>Chameleonic</i>
<b>ialgebra</b>	<C_expression_cast_to_int>	$X \rightarrow I$
<b>integrator</b>	<[-   +]input_node>	<i>Chameleonic</i>

<b>inv</b>	<input_node>	<i>I → I</i>
<b>itob</b>	<bit_rank> <input_node>	<i>I → I</i>
<b>itod</b>	<input_node>	<i>I → R</i>
<b>itool</b>	<itool_name> [<parameter_list>]	<i>X → I</i>
<b>iuser</b>	<iuser_name> [<parameter_list>]	<i>X → I</i>
<b>latch</b>	<set_input_node> <reset_input_node>	<i>I → I</i>
<b>lshift</b>	<number> <input_node>	<i>I → I</i>
	<shift_variable> <input_node>	<i>I → I</i>
	<shift_node> <input_node>	<i>I → I</i>
<b>max</b>	<[-   +]input_node> <[-   +]input_node...>	<i>Chameleonic</i>
<b>merge</b>	<[-   +]input_node_from_seg_a> <[-   +]input_node_from_seg_b...>	<i>Chameleonic</i>
<b>min</b>	<[-   +]input_node> <[-   +]input_node...>	<i>Chameleonic</i>
<b>mod</b>	<[-   +]input_node> <[-   +]input_node>	<i>Chameleonic</i>
	<[-   +]input_node> <[-   +]variable>	<i>Chameleonic</i>
	<[-   +]input_node> <[-   +]number>	<i>Chameleonic</i>
<b>muller</b>	<input_node> <input_node...>	<i>I → I</i>
<b>mux</b>	<control_node> <[-   +]input_node_0> <[-   +]input_node_1...>	<i>Chameleonic</i>
	<control_variable> <[-   +]input_node_0> <[-   +]input_node_1...>	<i>Chameleonic</i>
<b>nand</b>	<input_node> <input_node...>	<i>I → I</i>
<b>noise</b>	<[-   +]DC_level> <noise_density_level>	<i>→ R</i>
<b>nor</b>	<input_node> <input_node...>	<i>I → I</i>
<b>not</b>	<input_node>	<i>I → I</i>
<b>offset</b>	<[-   +]number> <input_node>	<i>Chameleonic</i>
	<[-   +]offset_variable> <input_node>	<i>Chameleonic</i>
<b>or</b>	<input_node> <input_node...>	<i>I → I</i>
<b>osc</b>	<[-   +]offset> <amplitude> <frequency> <[-   +]phase>	<i>X → R</i>
<b>poly</b>	<[-   +]coeff <sub>0</sub> > [ <[-   +]coeff <sub>i</sub> > ... ] <[-   +]input_node>	<i>Chameleonic</i>
<b>prod</b>	<[-   +]input_node> <[-   +]input_node...>	<i>Chameleonic</i>
<b>quant</b>	<input_node> <[-   +]input_node>	<i>Chameleonic</i>
	<variable> <[-   +]input_node>	<i>Chameleonic</i>
	<constant> <[-   +]input_node>	<i>Chameleonic</i>
<b>ram</b>	<name['addr_node']> <CS_node> <control_node> <W_node>	<i>Declared</i>
<b>ram2</b>	<name['addr_node']> <CS_node> <control_node> <W_node>	<i>Declared</i>
<b>rect</b>	<input_node>	<i>Chameleonic</i>
<b>register</b>	<control_node> <input_node>	<i>Chameleonic</i>
<b>relay</b>	<control_node> <input_node>	<i>Chameleonic</i>

	<control_variable> <input_node>	<i>Chameleonic</i>
	<control_node> <input_node> <[- +] setting_variable>	<i>Chameleonic</i>
	<control_variable> <input_node> <[- +] setting_variable>	<i>Chameleonic</i>
	<control_node> <input_node> <[- +] setting_constant>	<i>Chameleonic</i>
	<control_variable> <input_node> <[- +] setting_constant>	<i>Chameleonic</i>
<b>rip</b>	<hexavalued_mask> <input_node>	<i>I → I</i>
<b>rom</b>	<name>[‘addr_node’]> <CS_node>	<i>Declared</i>
<b>rom2</b>	<name>[‘addr_node’]> <CS_node>	<i>Declared</i>
<b>rshift</b>	<number> <input_node>	<i>I → I</i>
	<shift_variable> <input_node>	<i>I → I</i>
	<shift_node> <input_node>	<i>I → I</i>
<b>rshift1</b>	<number> <input_node>	<i>I → I</i>
	<shift_variable> <input_node>	<i>I → I</i>
	<shift_node> <input_node>	<i>I → I</i>
<b>rshift2</b>	<number> <input_node>	<i>I → I</i>
	<shift_variable> <input_node>	<i>I → I</i>
	<shift_node> <input_node>	<i>I → I</i>
<b>sign</b>	<input_node>	<i>X → I</i>
<b>sine</b>	<[- +]offset> <amplitude> <frequency> <[- +]phase>	<i>X → R</i>
<b>square</b>	<[- +]offset> <amplitude> <frequency> <delay> [<duty_cycle>]	<i>→ R</i>
<b>step</b>	<level1> <level2> <transition_time> [<transition_time>]	<i>→ R</i>
<b>sub</b>	<[- +]input_node> <[- +]input_node>	<i>Chameleonic</i>
	<[- +]input_node> <[- +]number>	<i>Chameleonic</i>
<b>sum</b>	<[- +]input_node> <[- +]input_node...>	<i>Chameleonic</i>
<b>toggle</b>	<input_node>	<i>I → I</i>
<b>track</b>	<control_node> <input_node>	<i>Chameleonic</i>
<b>triangle</b>	<[- +]offset> <amplitude> <frequency> <delay> [<duty_cycle>]	<i>→ R</i>
<b>trig</b>	<number> <input_node> [(dual)]	<i>X → I</i>
	<number> <input_node> (positive)	<i>X → I</i>
	<number> <input_node> (negative)	<i>X → I</i>
<b>uadc</b>	<num_level> <input_node> <reference_node>	<i>R → I</i>
<b>udac</b>	<num_level> <input_node> <reference_node>	<i>I → R</i>
<b>wsum</b>	<weight> <input_node> ... <weight> <input_node>	<i>Chameleonic</i>
<b>xnor</b>	<input_node> <input_node...>	<i>I → I</i>
<b>xor</b>	<input_node> <input_node...>	<i>I → I</i>
<b>zero</b>	<decimation_factor> <decimation_offset> <input_node>	<i>Chameleonic</i>

---

## Quick Reference: *NAPA 4.00* macros functions

**ABS(x)**  
**SIGN(x)**  
**MIN(x,y)**  
**MAX(x,y)**  
**CLIP(x,l,h)**

**ISINSIDE(x,l,h)**  
**ISOUTSIDE(x,l,h)**  
**ISEQUAL(x,y)**  
**ISNOTEQUAL(x,y)**  
**ISSMALL(x)**  
**ISNOTSMALL(x)**  
**ISEVEN(x)**  
**ISODD(x)**  
**ISINTEGER(x)**  
**ISTIME(t)**

**POWEROF2(n)**  
**MODULO(x,y)**  
**SIN(x)**  
**COS(x)**  
**SQR(x)**  
**SQRT(x)**  
**LOG(x)**  
**POW(x,y)**  
**ROOT(x,y)**  
**LOG10(x)**  
**POW10(y)**  
**D2I(x)**  
**I2D(n)**  
**DB2LIN(x,r)**  
**LIN2DB(x,r)**  
**DB2POW(x,r)**  
**POW2DB(x,r)**  
**RAD2DEG(x)**  
**DEG2RAD(x)**  
**LENGTH(s)**

**LINDOMAIN(c,b,e)**  
**LOGDOMAIN(c,b,e)**  
**LINSWEEP(c,b,e,n)**  
**LOGSWEEP(c,b,e,n)**

**RAND\_01()**  
**RAND\_01\_X()**

**FSS(n)**  
**STS(n)**  
**NIS(n)**  
**PS(n)**  
**SEGMENT\_CONDITION(n)**  
**TIMER(n)**  
**IO\_MANAGER(c,f,n,s,t)**  
**ISOPTION(f,i,o)**  
**ISNOTOPTION(f,i)**  
**ISDELAYED(f,i)**



---

## Quick Reference: The *NAPA* file system

Absolute reference	" / "
Reference to a <b>generic</b> library	< >
Reference to the <b>root</b> directory	" "
Reference to the <b>main</b> directory	"~/"
Reference to the <b>current cell</b> directory	". /"

A *generic library* is one of three particular libraries: header, cell and generator libraries

The *root directory* is the working directory from where *NAPA* compiler has been called.

The *main directory* is the directory containing the *NAPA* main netlist.

The *current cell directory* is the directory containing the cell currently processed.

---

## Quick Reference: *file naming recommendations*

<i>:NAPA</i> main netlist file	<b>xxxx.nap</b>	
<i>NAPA</i> netlist of a cell	<b>xxxx.net</b>	
<i>MAC</i> output	<b>xxxx.tmp</b>	(NAPA preprocessor)
<i>MAXIMA</i> package	<b>xxxx.mac</b>	
C code generated by <i>NAPA</i>	<b>xxxx.c</b>	
Executable binary code	no suffix	(UNIX platform)
	<b>xxxx.exe</b>	(DOS platform )
<i>NAPA</i> header file and user's profile	<b>xxxx.hdr</b>	
<i>NAPA</i> data cells	<b>xxxx.dat</b>	
Simulation output file	<b>xxxx.out</b>	
<i>NAPA</i> log file	<b>xxxx.log</b>	
<i>NAPA</i> dump file	<b>xxxx.dmp</b>	
<i>NAPA</i> ping file	<b>xxxx.png</b>	
<i>NAPA</i> load file	<b>xxxx.ini</b>	
<i>NAPA</i> RAM initialization file	<b>xxxx.ram</b>	
<i>NAPA</i> ROM description file	<b>xxxx.rom</b>	
<i>NAPA</i> generator (executable)	no suffix	(UNIX platform)
	<b>xxxx.exe</b>	(DOS platform )
<i>NAPA</i> generator (output cell)	<b>xxxx.gen</b>	
Graphics front end directives	<b>xxxx.plt</b>	(Gnuplot... )