ALDEG. THE DESIGN VERIFICATION COMPANY

IT USING COMMON TACTO

ady exists via a previ

ause sfinish is forke

create compose

Active-HDL 15 Overview Presentation

uvn coreservice



Integrated FPGA Design & Simulation

Active-HDL is both a *HDL Design tool* & a *Mixed* Language Simulator in one integrated Multi-FPGA (AMD[®], Altera[®], Lattice[®], Microchip[®]) and design entry and simulation environment

Included Technology

- Team-based Project Management
- > HDL Design Tool Suite
- Mixed Language RTL Simulator
- > VHDL, Verilog, SystemVerilog, SystemC
- Coverage Tools Bundle (Option)
- Assertions & Functional Coverage (Option)
- **DSP Co-Simulation MATLAB & Simulink**
- > PCB Interface
- **Documentation Generator**







Aldec Active-HDL in a nutshell

- **Common-Kernel Mixed Language Simulator**
- Languages: VHDL, Verilog, SystemVerilog, SystemC & EDIF
- **HDL Design Tools:** Design entry, Design Creation, Code2Graphics[™], Block and State Diagram, Waveform editor, stimulus generation, Language templates & auto-complete, scripting.
- **Design Flow Manager:** use popular third-party tools throughout the design flow within the same FPGA environment.
- **Debugging:** Code execution tracing, Waveform/Compare, Memory Viewer, Xtrace, Advanced Dataflow and Profiler.
- **Coverage:** Code Coverage, Toggle & Functional Coverage.
- **Additional Interfaces:** DSP/HDL algorithm MATLAB[®] and Simulink[®] Interfaces & Zuken CADSTAR PCB Design
- **Project Documentation** Entire workspace or single file export to PDF/HTML.
- Assertion and Coverage SystemVerilog and PSL support. Dedicated Assertion viewer, coverage, breakpoint editor.





FPGA Vendor Independent IDE for RTL-User





Design Flow Manager

- Design Flow Manager interfaces to 200+ different 3rd party tools
- Manages the HDL, C and Physical Synthesis Tools
- Runs the implementation for any FPGA vendor



- Runs the simulation at all stages of design
- > Invokes external analysis tools provided by silicon vendors







User-defined Design Management

- Separate HDL files from other types of files, e.g. schematics, text, waveform and scripts
- > Easily maintain and adhere to design structure required by entire team or a company
- > Create directory structure that is compatible to other synthesis and implementation tool





Advanced HDL Editor

- Code analysis and navigation tools
- Action Recorder for often repeated operations
- Keyword and Signal name auto-completion
- Automatic structure generation for enhanced legibility
- > Built-in customizable language assistant
- Source code auto-formatting
- Advanced Find, Find in Files and replace, Column Selection
- Navigation Bookmarks
- Ability to interface to third party text editors
- > OpenAI helper included



Block Diagram Editor

- Multi-page hierarchical block diagrams
- Multidimensional arrays and record signals supported
- > Bottom-up and top-down design methodologies supported
- Allows mixed structural and behavioral elements
- Cross probing with generated code
- > Handles mixed HDL designs
- > Customizable design rules checking
- Imports schematics
- > Customizable symbols







Library: Ipcor	e8051
Pin Name	Connection
len Port0(7:0)	1 PORT0(7:0)
len Port1(7:0)	1 PORT1(7:0)
len Port2(7:0)	1 PORT2(7:0)
len Port3(7:0)	1 PORT3(7:0)
CLK	<u>"] сік</u>
r■ ALE	" <u>]</u> <u>ALE</u>
PSEN	not connected
■ ₁ EA	" <u>]</u> <u>VCC0</u>
RST	¹ <u>RESET</u>



Finite State Machine Editor

- Multiple State Machines on a single diagram
- > Full-Moore machines support
- > Hierarchical states and junctions provided for legibility
- > Delay states simplify control of machine timing
- > Advanced code generation settings
- Automatic testbench generation
- Graphical FSM debug
- > FSM Coverage included

Testbench Gene	ration Settings	\times									
Select method											
Strategy1	Minimal test of moving through all vertexes of FSM state transition graph										
Strategy2	Strategy2 Minimal test of moving through all transitions										
Strategy3	Strategy3 Test of getting every FSM state starting from the initial one and next returning to it by using the reset signal										
	OK Cancel										







Project Documentation

- > Entire workspace, individual designs or files can be printed or exported to vector PDF
- > The HTML document closely resembles Active-HDL environment layout
- > Vector graphics and other advanced PDF options available
- Graphical Design Entry documents can be accompanied by generated code
- > Ideal solution for design documentation

 (\leftarrow)





Code2Graphics[™]

- Converts HDL code into graphical representation:
 - > Block Diagrams
 - Finite Stae Machines
 - > Editor files
- > Creates perfect documentation for HDL designs
- > Helps with debugging of large designs







Common Kernel Simulator

- VHDL, Verilog, EDIF, SystemC and SystemVerilog
- > Ultra Fast Verilog Simulation
- Strict IEEE Standards Adherence:
 - > VHDL: IEEE 1076-1993 Standard, IEEE 1076[™]-2002 VHDL Standard, IEEE 1076[™]-2008 Standard and IEEE 1076[™]-2019 Standard.
 - > Verilog: IEEE 1364-1995 Standard, IEEE 1364-2001 Standard, IEEE 1364-2005 Standard IEEE 1364-2005 Standard.
 - > SystemVerilog: IEEE Std. 1800[™]-2012
- > **PSL** embedded in **VHDL**
- > VHPI interfacing to C/C++ code
- > Verilog PLI (Programming Language Interface) and VPI (Verilog Procedural Interface) are supported as described in the IEEE 1364 Standard
- > Performance Optimizations (Verilog and VHDL)
- Extensive debugging while strictly adhering to IEEE Standards.



Spec-TRACER™

Traceability Capture and Reporting

- > Document Types:
 - > Microsoft[®] Word
 - > Microsoft[®] Excel
 - > Text/design files
- Capture methods:
 - > Regular expression
 - > Document style
 - > Spreadsheet/table column
- > Traceability Matrices Report
 - > Upstream and Downstream coverage analysis
 - > Traceability links consistency verification
 - > Microsoft[®] Excel and CSV



FPGA Requ	uirements		Test Scenarios					
Code	Name	File	Code	Name	File			
FPGA-001	Name of fpga-001	fpga requirements.txt	TST-001	Name of test1	test scenarios.txt			
FPGA-002	Name of fpga-002	fpga requirements.txt	TST-003	Name of test3	test scenarios.txt			
FPGA-003	Name of fpga-003	fpga_requirements.txt						



Description

Relations

_	
_	

HDL Source Encryption / IP Protection

- Support for the latest industry standards:
 - > IEEE 1735 Version 1 Recommendations
 - > IEEE 1735-2014 Version 2 using delegated rights
 - Subscription Methods defined in Verilog IEEE Std 1364-2005 and VHDL IEEE 1076-2008 Standards
 - Simulating encrypted files requires no action at Active-HDL user's *side* (given IP vendor encrypted the files for target simulation tool)
 - > Decrypts the source on-the-fly, leaving no traces that could compromise the security of encryption
 - Supports DES, 3DES, AES and Blowfish encryption algorithms
 - > Up to 256 bit encryption keys (AES)



Waveform Viewing and Editing

- Waveform Viewer for fast handling of large simulation databases
- Instant zooming and scrolling
- Multiple cursors and bookmarks
- > Waveform comparison
- Sort signals and browse by differences in two waveform files.
- > Export to VCD (Verilog and VHDL)
- > Embedded List Window displays the object values in an ASCII format
- > User-friendly Waveform Editor for manipulating signal data
- > Hand-drawing stimulus for interactive simulation
- > Preparation of data for automatic testbench generation
- Strobe Comparison command to compare signals probed by different strobe signals

.
top_tb (tb_arch
rarchy
top_tb (TB_A
🗕 🖶 UUT : top (
— ≢ U1 : se
🕀 🖶 🖶 U2 : fife
🖽 🖶 🖶 U3 : co
⊡- = U4 : co
🕀 🖶 U5 : rar
ine_1
CLOCK_CL
vl verilog log
std.textio
ne
CLK1
CLK2
CLR
DONE
Q
r ADDR_INCR
r CNTR_FULL
r EMPTY
r FIFO_READ
r FULL
MEM WR
Files)

Design Brow

_		-				
¢.		້ງ waves.awc				
cture)		G 🗗 🗸 🗟 🖞	▯ᄰᇒᄜᇔᅖᆓᄪ	# ⊕, ⊖, ©, ⊖, ⊕, ** **	┃끓 踢 �• ▮ ♥ ♥ № 計 삶 숙 ¶	/ ^ K % % S S S B
	^ Mode	Hierarchy	Signal name	Type Value	40	120 160 .
HITECT	n	/top_tb/UUT	FIFO_READ	STD 1		125 ns
ructure)	л	/top_tb/UUT	WDATA ■ WDATA ■	STD 20 to 6E	00 X54X6FX20X62X65X2CX20X6	FX72X20X6EX6FX74X20X74X
er (sender)	л	/top_tb/UUT	ADDR_INCR	STD 1		
pehavior)	л	/top_tb/UUT	ADDR	STD 02 to 03	(<u>00</u>) (01)	<u>02</u> 03 <u>04</u>
ol (beha	л	/top_tb/UUT	MEM_WR	STD 1		
ter	n	/top_tb/UUT		STD 32 to 98		L
	л	/top_tb/UUT	CNTR_FULL	STD 0		
	π	/top_tb	CLK1	STD 0 to 1		
	N.	/top_tb	CLK2	SID 1 to 0		
	JU Curr	riop_to	CLR	SID 0		
	Curr	or 2			<u>[50 ns]</u> 75 ns	
	cuis				1	125 ns
	¥					
		A constitute of a constant				
Value		y untitled.awc *				
0	\mathbf{S}	\mathcal{O} \mathbb{P} \mathcal{O} \mathbb{P} \mathbb{P}	▯◪◚▯▧	≝∣⊕, ⊝, ⊚, ⊝, ⊕, <mark>"*</mark> "*	🐝 📆 [や 📕 🔰 🕅 👭 👬 🔆 [🌾	i 📩 🍾 % 🧏 🔍 🎼 🕴
1	Mode	Hierarchy	Signal name 💌	Type Value	8 16 .	24 32
	л	/top_tb/UUT	WE	STD 0	0 fs	
0	лг	/top_tb/UUT		STD 00	00	X 54
0	л	/top_tb/UUT		STD UU		UU
20	n	/top_tb	₽Q	STD XX		XX
1	n	/top_tb/UUT	MEM_WR	STD U	· -1	
0	n	/top_tb/UUT	FULL	STD 0		
0	J.L.	/top_tb/UUT	FIFO_READ	SID U	· ·	
1	N.	/top_tb/UUT	EMPTY	SID 1		
1	- III	/top_to	CNTR FULL	SID U		
1	10	/top_tb/001		STD X		
0	Curs	or 1			0 fs	
4	v				•	fset
tructure / 🕅 Resources	s /					ffse
/tb/b_strobe	el at	61ns (25ns + 36ns	offset) -	strobe2 at 106ns (90ns + 16ns offs
/th/h strobe	al at	71ns (35ng + 36ng	offect) -	strobe2 at 136ns (120ns + 16ns off
/tb/b strobe	51 at	1115 (- 35H3 + 30H3 - 7Epo + 36p	offect)	atrobal at 256ma	12003 · 1003 01
	ar ar	111ns (s offset) -	strobez at 250hs	
/tb/b strobe	el at	121ns (85ns + 36n	s offset) -	strobe2 at 286ns	(_270ns + 16ns of
/tb/b strobe	el at	151ns (, 115ns + 36	ns offset)	- strobe2 at 376ns	(360ns + 16ns c
/tb/b strobe	el at	161ns (125ns + 36	ns offset)	- strobe2 at 406ns	(390ns + 16ns d
/tb/b strobe	el at	171ns (135ns + 36	ns offset)	- strobe2 at 436ns	(420ns + 16ns c
				-		
<pre>1 signal(s)</pre>	comp	ared. 9	difference(s) found in	l signal(s)	
	~ ~ mp*	2.00, 0	5211010100Q		- 019/00(07	





Debug





Debugging Tools

> View simulation results in

- > Waveform Viewer
- > List Viewer
- Memory View
- > Watch
- > Advanced Dataflow
- Trace code execution with
 - > Processes
 - > Call Stack
- Graphical Debug
 - > FSM
 - **BDE**

										ि untitled	awc *			– 🗆 🗙
										File Edit	Search	View Workspace	Design Simulatio	n Waveform
										Tools Wir	ndow He	elp		
1	M	emory	View 7					—		i 🗀 🖪 🖄	200		2 🔍 🖳 🔛 🐺 🌐	∰ ⊕ ୍ ୍ ୍ ୍ , ×
F	ile E	Edit S	Search	View ۱	Norkspac	e Desig	jn Sim	ulation	Memory	Time	Delta	/top_tb/UUT/WE	/top_tb/UUT/EMPTY	/top_tb/UUT/FULL
T I	ools	Wind	low He	lp		-				0 ps	0	U	U	U ^
4 4 4 4	<u>~</u> E		- ** 🤇	• 🗞 🖬	Addres	ss Start:	0			0 ps	1	0	1	0
0	bject	: <no< td=""><td>t availabl</td><td>e></td><td></td><td></td><td></td><td></td><td></td><td>25000 ps 35000 ps</td><td>2</td><td>1</td><td>1</td><td>0</td></no<>	t availabl	e>						25000 ps 35000 ps	2	1	1	0
		0	1	2	3	4	5	6	7	285000 ps	4	1	0	1
	00	00	01	02	03	04	05	06	07	300000 ps	5	1	0	0
		08	00	00	08	00	00		FF	305000 ps 325000 ps	4	1	0	0
`	- 80	00	09	UA	00	00	00			335000 ps	4	1	0	1
	10	10	11	12	13	14	15	16	1/	350000 ps	5	1	0	0
1	18	18	19	1A	1B	1C	1D	1E	1F	355000 ps 375000 ps	4	1	0	1
2	20	20	21	22	23	24	25	26	27	385000 ps	4	1	0	1
2	28	28	29	2A	2B	2C	2D	2E	2F	400000 ps	5	1	0	0
	30	Proces	ses			-				x ps	4	1	0	1
		Label			His	rarchy path			Chature) ps	4	1	0	1
		CTIMUL			1110	raicity paul			Deadu) ps	5	1	0	0
4	40	CLOCK			/pre	ss_top_to			Ready) ps	4	1	0	1
4	48	CLOCK_			/pre	ss_top_tb			Ready) ps	4	1	0	1
1	50 CLOCK_CLK_T /press_top_tb					Ready) ps	5	1	0	0		
tled.awc	*	WRITE_	IO_FILE		/pre	ss_top_tb			Ready) ps	4	1	0	1
lit Sear	rch Vi	MDS_m	achine		/pre	ss_top_tb/l	JUT/U101/I	U201	Ready) ps	4	1	0	1
1	$\odot \mathcal{O}$	OUTPU	T_assignme	nt	/pre	ss_top_tb/l	JUT/U101/I	U201	Ready) ps	5	1	0	0
lierarchy	/	CEL_ass	ignment		/pre	ss_top_tb/l	JUT/U101/U	U201	Ready) ps	4	1	0	1
top_tb/U	UT	CET_ass	ignment		/pre	ss_top_tb/l	JUT/U101/I	U201	Ready	↓) ps 585000 ps	5	1	0	0
top_tb/U top_tb/U		EMI			STD 0 STD 1					600000 ps	5	1	0	0
top_tb/U	UT	FIF	O_READ		STD 1					605000 ps	4	1	0	1
top_tb/U	IUT	⊕ WD	ATA		STD 7	4	<u> </u>	00000	00000000	625000 ps 635000 ps	5	1	0	0
top_tb/U top_tb/U	UT				STD 1	4				00000 p3		-	, , , , , , , , , , , , , , , , , , ,	wave acdb SIM
top_tb/U	UT	ME	M_WR		STD 1	•			~=~~					wave.asub Silvi
top_tb/U	UT	🗆 RD	ATA		STD 2	0	UL	J (54)	6F 20 62	65 X 2C X 20	χ <u>6</u> Γ χ 72	<u>20 (6E 6F 74</u>	<u>20 74 6F 20</u>	<u>62</u>
top_tb/U			RDATA[7]		STD 0									
top_tb/U	UT		RDATA[5]		STD 1									<u> </u>
top_tb/U	UT		RDATA[4]		STD 0	I						1		
top_tb/U	UT		RDATA[3]		STD 0									
top_tb/U	UT		RDATA[2]		STD 0					<u> </u>				
top_tb/U	UT		RDATA[0]		STD 0						<u> </u>			
top_tb/U	UT	CN	TR_FULL		STD 0									
1 	atch													×
VV.	atch			I _		. .						x	wave	asdb SIM
	Name		T (FLU)	1	ype TD LOCIC	Va	lue		Last Value	Last E	vent Time			
	лг/te	op_tb/UU		S		1			1	1455n	IS			
	JU /to	op_tb/UU		S		1			0	35hs	DC.			
	лт /н	op_tb/UU		د ۱ د		1			x	57500 1nc	hz			
	3470	op_00/00	", can in the		10_0000	v			~	1115				

0

62500ps



Active-HDL 15 Overview Presentation

Click here to add new item

STD_LOGIC

Stimulus

Graphical Stimulus Creation the waveform viewer.

Force command

custom and text.

- > When a text file stimulus is used, the force command reads the values of signals to be forced from a plain text file.
- > The custom stimulus can read test vectors from a number of file types:
 - > Aldec Simulation Database (.ASDB)
 - Custom text stimulus (.TXT)
 - > Value Change Dump (.VCD)
 - MATLAB Binary Data Files (.MAT)
- Force Capabilities Inside HDL Code

The functionality of the force command is available not only in macros but

Convenient GUI window (Stimuli Window) for creation of force stimulus for use in

The syntax of force command provides two kinds of stimuli based on file contents -

also directly in VHDL or Verilog code (VHDL procedure force and Verilog task \$force).



Advanced Dataflow and XTrace

- Allows exploration of design in graphical representation
- Shows the drivers-readers connection and simulation values
- The XWay feature finds the source of unknown value
- Gray mode provided for better legibility
- > The XTrace feature allows detection of irregular values when they happen
- Combined with advanced dataflow creates powerful debugging tool



	The	XTrace R	eport	
		Parameters		
Switch name		Value		
-value		X U - x		
-escape_init -stop_condition		1		
		Statistics		
Time	Signal name	Signal value	Driver name	Drive value





Coverage

- Code coverage checks how many times each statement is executed
- > Branch/expression/condition coverage provide detailed statistics on execution of conditional statements and assignments
- > Path Coverage collects information about the program execution and program sequences (program paths)
- > Toggle coverage reports design activity in terms of changes in signal values
- > FSM coverage allows to identify not visited states, transitions and sequences of states
- Functional coverage allows to collect coverage data from OSVVM code and SystemVerilog (Verification)

44		end
45		endcase
46		endgenerate
47		
48		<pre>base64_enc dut(.clk(clk),.reset(reset),.fd_in(fd_in),.fd_out(fd_out),.complete(dc</pre>
49		
50		// clock generator
51		always
52	1	$#25 \ clk = ~clk;$
53		
54		// main
55		initial begin
56	1	#100 reset = 1'b1; // reset done
57		fork
58	1	@(posedge done);
59	1	#100000;
60		join_any
61	1	if (done)
62	1	<pre>\$info("Encryption done");</pre>
63		else
64	0	<pre>\$warning("Encryption failed - timeout");</pre>
65	1	<pre>\$fclose(fd_in);</pre>
66	1	<pre>\$fclose(fd_out);</pre>
67	1	#10 \$finish;
68		end
69		endmodule

CUMULATIVE SUMMARY

Coverage Type	Weight	Hits/Total
Statement Coverage	1	73.972%
Statements		108 / 146
Subprograms		2 / 4
Branch Coverage	1	70.247%
Branch paths		85 / 121
Branches		20 / 25
Covergroup Coverage	1	87.500%
Types		0/1
Expression Coverage	1	100.000%
Expression bins		2 / 2
Expressions		1/1
Condition Coverage	1	92.000%
Condition bins		23 / 25
Conditions		6 / 8
Toggle Coverage	1	100.000%
Toggle bins		26 / 26
Signal bits		13 / 13
Assertion Coverage	1	100.000%
Assertions		5 / 5
Covers		3/3

FSM Hits/Total 75.0001 fsm partse * States - S1 + 52 - 53 Transition - S1-> S2 - <u>\$1->53</u> - <u>52->51</u> + <u>\$2->53</u> - <u>53->51</u> - <u>S3->S2</u> 66.6669 fsm_concat * States +54H 25 H 56 - ST Transition



CUMULATIVE INSTANCE-BASED COVERAGE: 89.102% COVERED INSTANCES: 1 / 4 FILES: 3

Verification Plan

- Enables validating the verification process with a test plan
- It allows users to link the **requirements** from test plan to coverage results to see if verification goals have been reached or not
- Allows user to **rank** tests to select > minimal testsuit for **regression tests**

	6	Aldec Cov	verage Report	:								
	<u>F</u> i	le <u>E</u> dit S	ea <u>r</u> ch <u>V</u> iew	W <u>o</u> rkspace	<u>D</u> esign <u>S</u> imulation <u>T</u> ools <u>W</u>	indo	w <u>H</u> elp					
		$\Leftrightarrow \Rightarrow \textcircled{\bullet}$	C 🖶									
		Design				<	Verification Plan - Cumulative	Summary				
		Verificatio	on Plan Ian				CUMULATIVE SUMMARY					
		4 😥 1 Co	ode Coverag 1.1 FPGA-01	e			Section	Local Coverage	Local Hits/Total	Local Link Status	Recursive Coverage	Recu Hits/1
		😥 1	.2 FPGA-02				🝷 0 testplan	-	-	-	98.958%	674
			1.3 FPGA-03				▼ <u>1 Code Coverage</u>	No bins	0/0	No links	96.875%	636
		😥 1	.5 FPGA-05				- <u>1.1 FPGA-01</u>	100.000%	176 / 176	All	100.000%	176
RANKIN		· · _			-		- <u>1.2 FPGA-02</u>	100.000%	129 / 129	All	100.000%	129
		_	_				- <u>1.3 FPGA-03</u>	100.000%	3/3	All	100.000%	
Id	Cumulative Coverage	Test Coverage	Coverage Increment	Contributing Items	Test Name		₩ <u>1.4 FPGA-04</u>	100.000%	30 / 30	All	100.000%	30
▼ Cont	ribution tests		1	1			- <u>1.5 FPGA-05</u>	100.000%	16 / 16	All	100.000%	16
ŀ- 1	<u>95.961%</u>	<u>95.961%</u>	<u>95.961%</u>	<u>384</u>	./output/results.acdb:testCFG4		₩ <u>1.6 FPGA-06</u>	75.000%	6/8	All	75.000%	
⊨ <u>2</u>	<u>96.702%</u>	<u>80.507%</u>	<u>0.741%</u>	<u>4</u>	./output/results.acdb:testCFG3		₩ <u>1.7 FPGA-07</u>	100.000%	138 / 138	All	100.000%	138
··· 3	<u>97.222%</u>	<u>90.515%</u>	<u>0.519%</u>	3	./output/results.acdb:testCFG1		ⁱ <u>1.8 FPGA-08</u>	100.000%	138 / 138	All	100.000%	138
▼ Redu	Indant tests			1			 2 Functional Coverage 	No bins	0/0	No links	100.000%	22
··· 4	97.222%	87.381%	0.000%	<u>0</u>	./output/results.acdb:testCFG2		- 2.1 FPGA-09	100.000%	1/1	All	100.000%	
- 5	<u>97.222%</u>	<u>95.961%</u>	0.000%	<u>0</u>	./output/results.acdb:testCFG5		<					-
···· 6	97.222%	82.448%	0.000%	<u>0</u>	./output/results.acdb:testCFG6							





Extensive list of Verification Libraries Support

Active-HDL compiles and simulates HDL verification constructs. This provides the ability to simulate using advanced verification methods like functional coverage and constrained randomization.

- Active-HDL supports followings verification libraries: > UVM 1.1d, 1.2, 1800.2-2017, 1800.2-2020
- > OSVVM
- > UVVM
- > VUnit











Profiler

Profiler measures information on statement execution time and helps to identify performance bottlenecks

· · · · · · · · · · · · · · · · · · ·	⊽ AAL t = ≣)ee a	#E 🔺 🧐	- 12 X 41	bb			
lierarchy	CPU Ticks	Share [%]	Time [us]	CPU Ticks with C	Share	REG.vhd	Detai
- = Root : datapath.Dummy top-leve	-	-	-	61 617 703	~	Line	CPUtick
+ solve > : < other code > : < other code >	59 494 534	96. 55	17 457. 32	59 494 534		10	None
other code >	59 494 534	96. 55	17 457. 32	59 494 534		20	None
- top_testbench : datapath.top	1 167 977	1.90	342. 72	2 123 169		21	None
🕀 🏠 WRITE_TO_FILE	1 054 038	1.71	309.28	1 054 038		22	None
– 🔂 CLOCK_CLK	60 765	0. 10	17.83	60 765		23	None
– Ъ STIMULUS	53 174	0.09	15.60	53 174		24	None
占 扣 UUT : datapath.top (top)	0	0.00	0.00	955 192		25	None
🔄 👥 U0 : datapath.datapat	0	0.00	0.00	686 879		20	None
🖃 🚛 U0 : datapath.ALU	11 408	0. 02	3.35	435 922		28	None
🕂 🛨 🛨 U0 : datapath.A	331 149	0.54	97.17	331 149		29	None
庄 🤹 U1 : datapath.re	93 365	0.15	27.40	93 365		30	None
— 🄂 line109	10 963	0. 02	3. 22	10 963		31	None
🗆 🍌 line_104	445	0.00	0. 13	445		32	None
🔄 🛟 U1 : datapath.REG	3 562	0. 01	1.05	165 890		33	None
🕂 💶 U0 : datapath.re	84 129	0.14	24.69	84 129		34	None
🕀 🚮: U1 : datapath.re	78 199	0.13	22. 95	78 199		36	None
- 🔂 line 67	2 484	0.00	0.73	2 484		37	None
- 5 line 68	1 078	0.00	0.32	1 078		38	None
U2 : datapath.COM	1 448	0.00	0.42	85 067		39	None
🕂 📲 U1 : datapath.re	75 632	0. 12	22. 19	75 632		40	None
⊕ : U0 : datapath.C	5 374	0. 01	1. 58	5 374		41	None
+ J: U2 : datapath.M	2 613	0.00	0.77	2 613		42	None
T 🔂 line 88	1 448	0.00	0.42	1 448		43	None
- It U1 : datapath.CONTR	0	0.00	0.00	268 313		45	10 899
- 1: U0 : datapath.contr	160 588	0.26	47, 12	160 588		46	None
CONTROL mac	87 658	0.14	25.72	87 658		47	7 798
- LOAD COUNT.	14 810	0.02	4.35	14 810		48	1 350*
	14 108	0.02	4 14	14 108		49	None
	12 606	0.02	3 70	12 606		50	10 700
	11 264	0.02	3 31	11 264		52	240*
	10 568	0.02	3 10	10.568		53	3 544
	9 574	0.02	2 81	9 574		54	6 958
Et al. U1 : datapath COU	107 725	0.17	31 61	107 725		55	7 302
	103 733	0.17	30 44	103 733		56	7 636
2 mio_10	100 700	0.17	00.11	100 700	~	57	None
					>	59	None

	_		Х	
-{{ Section below this comment is automatically maintaineo - and may be overwritten -{entity {reg} architecture {reg}} library IEEE; use IEEE.STD_LOGIC_1164.all; entity reg is generic(S : NATURAL := 15				
); port(CLK : in std_logic; DIN : in std_logic_vector(S downto 0); LOAD : in std_logic; RESET : in std_logic; DOUT : out std_logic_vector(S downto 0)); end reg;				
architecture reg of reg is begin				
process (CLK,RESET)				
if RESET = '1' then DOUT <= (others => '0); else				
if CLK event and CLK = 'il then if LOAD = 'il then DOUT <= DIN;				
end if: end if: end if:				
end process;				
end reg;				
				1
		ilter : OF	F	
		inter : OF		



Automatic Testbench Generation

- > Automatic generation of simulation testbench in different styles:
 - Generate testbench template for any design unit.
 - Generate complete testbench from test vector inputs such as waveform file.
 - Generate complete testbench that compares simulation result against golden test vectors.
- Automatic Testbench Generation for Finite State Machines (FSM) in three different styles

Testbench Generator Wizard			×		
Define test vectors					
Test vectors from file Select this check box if you want to saved in a waveform file.) use previously cre	ated test vectors			
Select a test vector file: C:\Mv Designs\Samples 111 x64	VEIFO mix\src\wav	es.asdb			
		Testbench Gene	ration Setting	5	×
Signals found in file: CLK1 CLK2 CLR	UUT ports CLK1 CLK2 CLR	Select method Strategy1 Strategy2 Strategy3	Minimal test of transition grap Minimal test of Test of gettin and next retu	f moving through all vertexes of FSM star h f moving through all transitions g every FSM state starting from the initial ming to it by using the reset signal OK	te one Cancel
	< Back	Next >	Cancel		

Γλ In	nut Stimulus.awc*	:					×	
File	Edit Search Vie	w Workspace Design	Simulation	Waveform Tr	Niedew Help		3	
i ne								
	<u> </u>		≦ ‡‡ ‡∰	# @ (@(@	【< 【 【 【 【 【 【 【 【 (<			
Mode	Hierarchy	Signal name	Туре	Value	· · · · 20 · · · · 40 · · · · 60 · · · · 80 · · · · 100 · · · · 120 · · · · 140 · · · · 180 · · · · 180 · · · 200 ·	• • 2	20 ns	
лг	/top_tb	CLK1	STD	0 to 1			<u> </u>	
лг	/top_tb	CLK2	STD	1 to 0				
лı	/top_tb	CLR	STD	0				
Curso	Cursor 1							
						Þ	 	
						wave.	asdb SIM	



MATLAB and Simulink Co-Simulation

- Direct Co-Simulation Interface to:
 - > MATLAB
 - > Simulink
- > Call MATLAB functions directly from HDL code or call Active-HDL from MATLAB
- > High speed system-level testbench and co-verification environment





Simple and fast setup of all Simulink co-simulation parameters



Unit Linting

- > Build-in interface to ALINT-PRO allows t lint selected units and present results w **Active-HDL User Interface**
- > Pre-linting allows to get rid of most issu before full linting and synthesis process

library IEEE use IEEE.STD use IEEE.STD	; _LOGIC_1164.all; _LOGIC_ARITH.all;
use IEEE.STD	LOGIC_UNSIGNED.all:
entity binzb	CO15
BIN:	in STD LOGIC VECTOR (4 downto 0);
BCD	H: out STD LOGIC VECTOR (1 downto 0) :
BCD	L: out STD_LOGIC_VECTOR (3 downto 0));
end bin2bcd;	
architecture	BIN2BCD_ARCH_of_bin2bcd_is
10000	Warning(s):
begin	STARC_VHDL.1.1.6.1 Architecture "bin2bcd(BIN2BCD_ARCH)". The "BIN2BCD_ARCH" architecture name does
BCD 1 <=	"0001" when "00001" "01011" "10101" "11111"
	"0010" when "00010" "01100" "10110".
	"0011" when "00011" "01101" "10111".
~~~~~~	"0100" when "00100"   "01110"   "11000",
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
Concolo	
 # ALINT-PRO # ALINT-PRO # ALINT-PRO # ALINT-PRO # ALINT-PRO does not mat 	C:/My_Designs/Samples_111_x64/Bjack/src/disp_units.vhd : (76, 1): Indentation of "H_ C:/My_Designs/Samples_111_x64/Bjack/src/disp_units.vhd : (81, 1): Indentation of "LE C:/My_Designs/Samples_111_x64/Bjack/src/disp_units.vhd : (83, 1): Indentation of "LE Warning: STARC_VHDL.1.1.6.1 C:/My_Designs/Samples_111_x64/Bjack/src/disp_units.vhd : ch required regular expression "RTL" that should be used for the RTL code description Warning: RMM_VHDL 5.2.5.2 C:/My_Designs/Samples_111_x64/Bjack/src/disp_units.vhd :
 # ALINT-PRO "H_TMP" sign # ALINT-PRO "L_TMP", "H # Linting f 	al declaration(s). Level: Recommendation 2. Warning: STARC_VHDL.3.5.6.3_a C:/My_Designs/Samples_111_x64/Bjack/src/disp_units.vh TMP" signal declaration(s). It is recommended to add comments in the same line with nished successfully.
<pre></pre>	Warning: STARC_VHDL.3.5.6.3_a C:/My_Designs/Samples_111_x64/Bjack/Sic/disp_units.vh TMP" signal declaration(s). It is recommended to add comments in the same line with .nished successfully.

	🕞 Design Browser		÷ 🗙	🔳 🔒 fre	q_top.bc
	testbench_cnt_b	cd_conf	~	X G 🛛	330
	O. Unsorted	1.0 1.5			
0	U Workspac	2	· · · ·		
	⊡ U freq_mete				
vithin	- The Readm				
	1 ⊞ 🔤 🗹 and2.v	hd			
		D o kal			
	3 ⊞∰⊘h	Open			
		Open in		>	
		Add file to design			
163	7 ⊞ ∰ ☉ fr	Refresh contents			:::::
5	- 🔂 C	Export tags			
	10 ⊞ ী 🖋 te – பு w	Assign HDL Files			F_INF
	- 🖧 w 🐎 .	Compile		F11	PATTE
	- <u>%</u> A	Exclude from Compilation	١		RES
	🕀 🎹 🦷 🔂	Run Unit Linting			CT/
	₽3	Synthesis Check			51A
		C			

ot match required regular expression "RTL" that should be used for the RTL code description. Level: Recommendation 1

'MP" concurrent statement is 2 space character(s) less than it is required. L" concurrent statement is 2 space character(s) less than it is required. H" concurrent statement is 2 space character(s) less than it is required. (50, 1): Architecture "bcd2led(BCD2LED_ARCH)". The "BCD2LED_ARCH" architecture name Level: Recommendation 1. 53, 2): Architecture "bcd2led(BCD2LED_ARCH)". Comments are missed for the "L_TMP", : (53, 2): Architecture "bcd2led(BCD2LED_ARCH)". No comments are provided for the at declaration in order to improve the code readability. Level: Recommendation 2



ALINT-PRO Interface

- Detects wide variety of design problems at design entry
 - > Be aware of *hidden bugs* (from synthesizability to CDC) and DFT issues)
 - > **Prevent issues** at subsequent stages of the flow (avoid costly re-spins)
- Get started quickly using predefined *design checking* policies
- > Pick a dedicated *rule plug-in* (STARC, DO-254, RMM, CDC and more)
- Available as an option for DM and PE configurations
- Included in EE configuration





Assertion Based Verification SVA and PSL

> PSL and SystemVerilog

- > Properties/assertions/covers embedded in HDL code or kept in separate files
- > Assertion and cover messages in simulator console
- View assertions and covers in a dedicated Assertion Viewer (status and statistics for each assertion/cover)
- > Set Assertion Breakpoints and edit them in the new Assertion tab of Breakpoint Editor
- Assertions and Covers tabs in Code Coverage Viewer : view statistics for the entire monitored design
- > Assertions displayed in Structure tab of the Design Browser
- > HDL Editor and Language Assistant: syntax highlighting, code auto-complete, and predefined language templates.



Conclusion

- Common Kernel Simulator supporting mixed VHDL, Verilog, EDIF, SystemC and SystemVerilog simulation
- Integrated Design Entry and Verification Environment
- Interfaces to FPGA and Logic Synthesis product from universal Design Flow manager – vendor independent
- > Ability to invoke simulations in batch mode or GUI, compete user control > Advanced Debugging and Analysis Tools
- Support for Encrypted IP delivery from FPGA Vendors and IP Providers > Powerful interface to Matlab/Simulink for signal processing applications
- co-simulation
- Assertions Support Available SVA, PSL > Advanced verification: UVM, OSVVM, UVVM, Cocotb.





Corp. Headquarters

2260 Corporate Circle, Suite 400 Henderson, NV 89074

Ph +1.702.990.4400 Fax +1.702.990.4414 sales@aldec.com

Training <u>https://www.aldec.com/training</u>

Support <u>https://www.aldec.com/support</u>

https://www.aldec.com/company/blog Blog

> You Tube in E 31

Solutions

Riviera-PRO[™] **Advanced Verification Platform**

Active-HDL[™] FPGA Design and Simulation

ALINT-PRO[™] Design Rule Checking

HES-DVM[™] HW/SW Validation Platform

TySOM™ High Performance Embedded Development **Spec-TRACER™ Requirements Lifecycle Management**

DO-254/CTS™ FPGA Level In-Target Testing

HES™ SoC/ASIC Prototyping

RTAX/RTSX Prototyping Microchip[™] Rad-Tolerant Devices

