CyberWorkBench® High-Level Synthesis and Verification

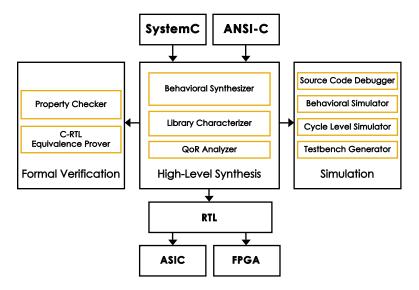
High-Level Synthesis and Verification

CyberWorkBench® enables higher design efficiency, low power and high performance of the chip by allowing designers to implement hardware at the algorithmic level. This "All-in-C" high-level synthesis and verification tool set for ASIC and FPGAs (Xilinx/Altera) reduces the development time and cost significantly.

Designers can describe hardware at higher abstraction level using SystemC and ANSI-C and using CyberWorkBench they can generate highly optimized RTL for their ASIC and FPGAs (Xilinx/Altera) chip. Automatic pipelining, power optimization and powerful parallelism extraction allows designers to generate smaller and low power designs compared to manual RTL design approach.

Top Benefits

- Support for both control dominated circuits and datapath module
- Dedicated technology support for Altera® & Xilinx® **FPGAs**
- Best-in-class High-Level Synthesizer that features automatic pipelining, power optimization, powerful parallelism extraction
- Powerful graphical analysis capabilities for synthesized circuits
- C-based Formal Verification using assertions and properties
- Automatic top level structural description generator to connect C-based modules and legacy RTL modules
- Powerful SystemC source code debugger
- Legacy/IP RTL code to SystemC converter for easier migration to C-based design flow



"All-in-C" Synthesis

Behavioral synthesizer in CyberWorkBench can synthesize any type of application including control dominated circuits and datapath. This best-in-class high-level synthesizer features automatic pipelining, power optimization and powerful parallelism extraction to reduce chip area and power through maximum resource sharing. Designers with IP/RTL legacy modules can use top level structural description generator and can easily connect to C-based modules. To improve the design productivity CyberWorkBench also includes numerous behavioral IPs that can be retargeted to different implementation technologies or system requirements.

"All-in-C" Verification

CyberWorkBench provides powerful static and dynamic verification tools to make debugging of larger designs much easier. Formal verification of the high level source code using C level property checker enables designers to describe assertions and properties directly in C source code. Built-in automated testbench generator cuts verification time by allowing re-usage of untimed C stimulus in SystemC and RTL simulation.



CyberWorkBench®

	Enterprise	Professional	Standard	Basic
Version	Ver5.2	Ver5.2	Ver5.2	Ver5.
Target Device	ASIC / FPGA	FPGA	FPGA	FPGA
Common Functionality	•	•	•	•
Option	•	•	•	
Input Description Size	No Limit	No Limit	Limited	Limite
How To Run				
GUI	•	•	•	•
Command Line	•	•		
Target Device				
ASIC	•			
ALTERA Stratix	•	•		
(ILINX Virtex	•			
ALTERA Arria	•		•*1	
KILINX Kintex	•	•	•*1	
ALTERA Cyclone	•		•*1	• •
(ILINX Artix (Spartan)			•*1	• 4
License Type				
Jser Lock Licence				
Node Lock Licence	•	•		
Floating Licence				
License Term				
1 year				
2 years	•			
3 years				
Perpetual				

^{*1:} Available for one of the FPGA Series marked (\cdot)

Advantages

Design Cost Reduction:

- Description reduction: 5-30%, simulation speed 100X
- Design man-month reduction (e.g. 80MM 10MM)
- Design period for HW (blue) and SW (red) are both reduced
- Higher reliability: fast HW-SW co-verification (cycle accurate)

Chip Cost Reduction: Smaller Area/Power

- HLS can generate smaller and lower power designs compared to manual RTL designs through maximum resource sharing
- · Area and performance optimization for Altera, Xilinx FPGA

Applications

- Digital Circuits: mobile phone, base station, computer,
 transmission equipment, STB, digital camera, printer, controllers
- Acceleration of server: Big data (e.g. High Frequency Trading)
- Replacement of real time processing MPU: sensor monitoring and motor control
- Controller: Factory Automation, Medical, Automobile

Ph. +86.21.6875.2030 info@aldec.com.cn India Ph. +91.80.3255.1030 sales-in@aldec.com

