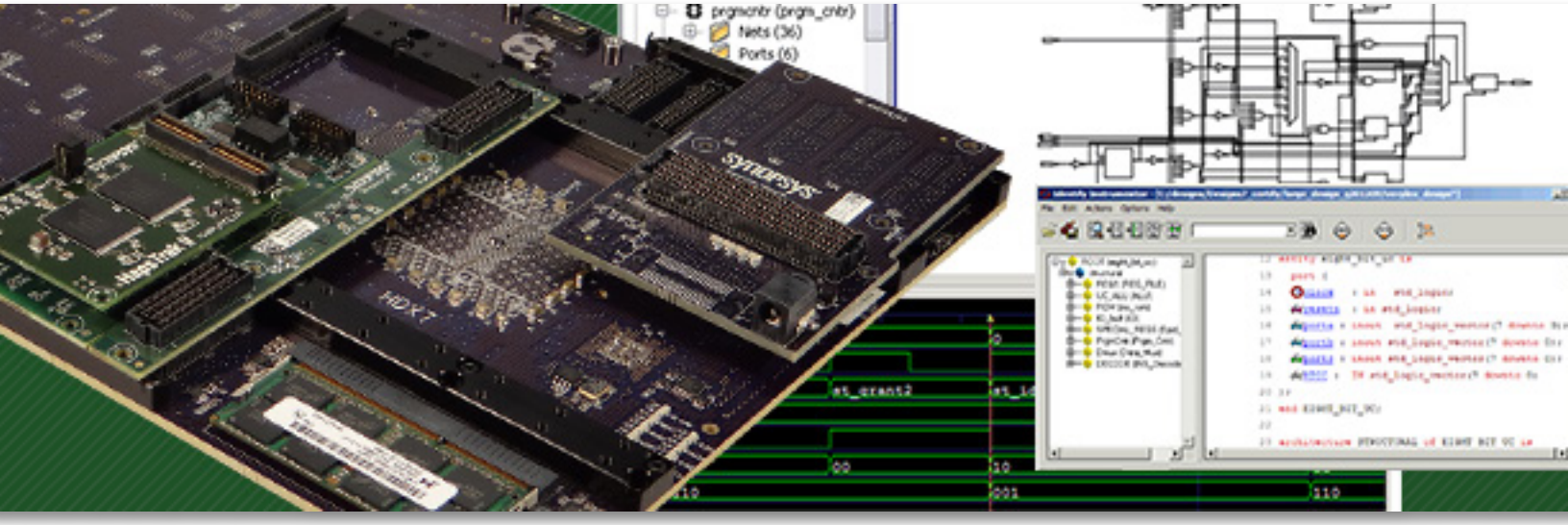
SYNOPSYS®
Accelerating Innovation



Synopsys' HAPS Developer eXpress FPGA-based Prototyping Solution

- Ideal for prototyping new ASIC blocks, IP, and subsystems
- Includes best of class HDL source compilers and FPGA-based prototyping synthesis
- Provides simulator-like visibility into prototype operation
- Integrates Xilinx Virtex-7 FPGA, DDR3 memory, PCI Express, and UMRBus
- Supports ANSI standard FMC and Synopsys HapsTrak 3 connectors
- Integrates with high-capacity Synopsys HAPS-70 series systems

Overview

Time is of the Essence.

FPGA-based prototypes deliver high-performance operation and real-world connectivity but unless they can be brought-up and deployed early in the ASIC development project the speed and connectivity benefits are of no use if the prototype is late. Prototypes must be rapidly assembled and ASIC RTL "drops" integrated and made operational for validation scenarios and software integration in weeks – not months. With development cycles shrinking and software content growing, the demand for software-driven, in-context validation of new RTL blocks and IP requires that FPGA-based prototypes be delivered as fast as possible.

The Synopsys HAPS Developer eXpress (HAPS-DX) provides best of class prototyping hardware and automation software tools in a package that is focused on rapid bring-up and accelerating the availability of ASIC RTL block and IP prototypes for design teams who need state-of-the-art prototyping solutions.

- Integrated Xilinx Virtex-7 690T FPGA device provides up to 4 million ASIC gates of capacity with Configurable Logic Block (CLBs), RAM, and DSP resources ideal for ASIC block module and IP validation
- I/O interfaces compatible with both industry standard FPGA Mezzanine Card (FMC) and HAPS HapsTrak 3 formats provide designers with a wide selection of daughter boards reducing your effort to assemble prototypes with real-world interfaces
- Linux OS compatible prototype automation and debug software included with every HAPS-DX system speeds-bring up
- Compatible design flow and hardware interfaces with HAPS-70 Series systems expand prototype capacity to make full SoC validation feasible
- HDL compilers support popular formats, recognize synthesis coding styles, and DesignWare IP. ASIC design constraint recognition of Synopsys Design Constraints (SDC) and Universal Power Format (UPF) speeds the migration of timing and power intent into the prototype
- Fast HDL compiler modes reduce the review time of RTL and provides up to 4 times faster throughput than traditional FPGA synthesis tools. Fast prototype bring-up options like HAPS Clock Optimization (HCO) allow even the most complex ASIC clocking schemes to be implemented quickly in clock-limited FPGA architectures
- RTL debug and high-capacity storage options provide up to 8 gigabytes of storage and a simulator-like RTL debug interface for design troubleshooting and protocol compliance checks
- Integrated Universal Multi-Resource Bus (UMRBus) hardware interface and C/C++/Tcl APIs provide intimate control and visibility of the prototype from a host workstation

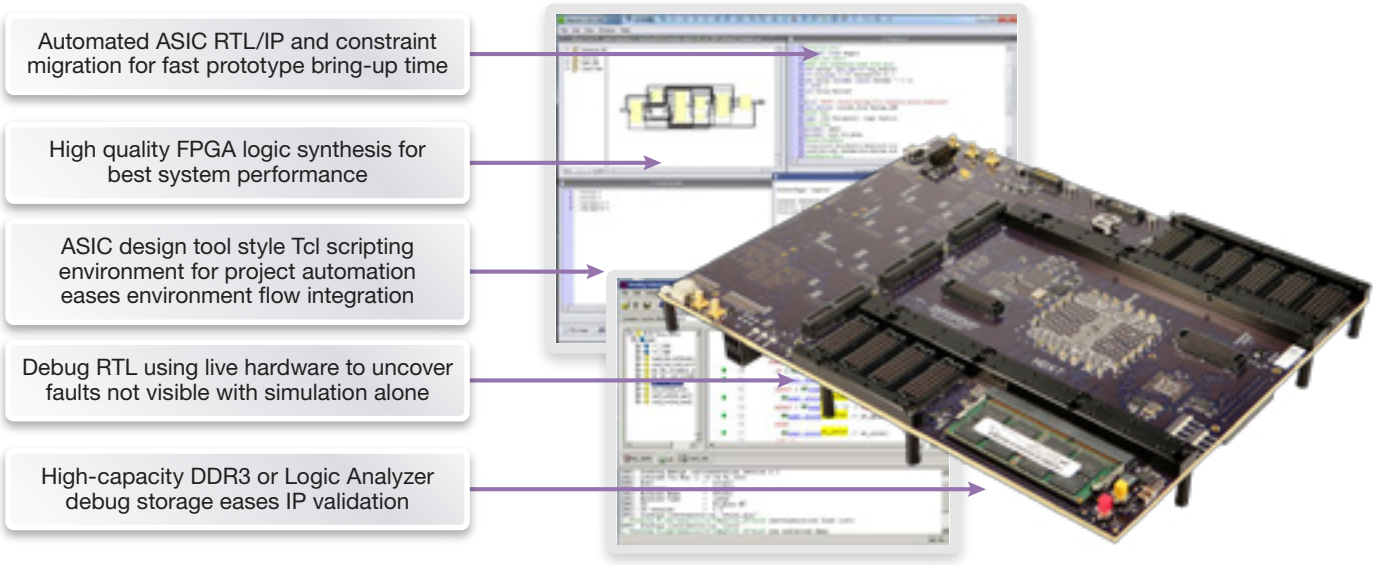


Figure 1: HAPS-DX hardware system with automation and debug software

	HAPS-DX7 S4
FPGA Type	Virtex-7 690T
ASIC Gate Capacity	Up to 4 million
DDR3 SDRAM Capacity	Up to 8 gigabyte
User Accessible Virtex-7 General Purpose I/O	500 (50 I/Os per HapsTrak 3 I/O connector) +
	10 (GPIO connector) +
	40 (10 per HSIO MGB connector) +
	16 (8 per HSIO FMC connector)
User Accessible Virtex-7 GTH Transceivers	56 (10 GTH transceivers per HSIO MGB Connector Socket) (8 GTH transceivers per HSIO FMC Connector Socket)
User LEDs	(4) Red/Green Dual Color
HapsTrak 3 I/O Connector Sockets	10 (160 position SEAF Open Pin Field Array Sockets) (HapsTrak 3 to HapsTrak II adapter cards available)
HSIO MGB Connector Sockets	4 (80-pin Vertical Edge Rate Card Sockets) (access to 10 GTH transceivers + 10 GPIO per socket)
HSIO FMC Connector Sockets	2 (60-pin SEAF Open Pin Field Array Sockets) (access to 8 GTH transceivers + 8 GPIO per socket) (used with HapsTrak 3 to FMC adapter boards)
GPIO (General Purpose I/O) Connector Socket	1 (2x7-pin, 2.00mm pitch header) (access to 10 GPIO)
DDR3 SODIMM Connector Socket	1 (DDR3 SODIMM 204 position right-angle socket)
Clock Resources	1 PLL with 3 clock nets to the FPGA, 1 external PLL input, 2 external PLL outputs, 2 external coax clock I/Os, frequency range 5-200 MHz for PLL inputs, 160 kHz – 700 MHz for PLL outputs
Programmable Voltage Regions	1.8V, 1.5V, 1.35V, or 1.2V
Clock Regions	4
Debug Modes	RTL Level Debug, Sample Mux Groups, Multi-FPGA Distributed Debug, Deep Trace Debug, Real Time Debug with Logic Analyzer
Daughter Board Portfolio	PCIe, SATA, Ethernet, DDR2, SRAM, FLASH, MSDRAM, MICTOR, FMC compatible with HapsTrak 3 mezzanine card
Prototype Automation Software	HAPS-DX synthesis and implementation tools included
RTL Debug and Troubleshooting Software	HAPS-DX RTL debug and system assembly validation tools included
System Control Software	System configuration and monitoring software tools included
Configuration	SD Card (up to 10 boot configurations selectable via rotary switch), UMRBus via Configuration and Data Exchange (CDE) interface, JTAG, USB 2.0
Encryption Key	Battery backup support
Power Supply Unit Input	110-240 AC, 12V
Accessories Included	Power Supply
Optional Accessories	PCIe edge connector board HapsTrak 3 to FMC adapter board (access to 160 GPIO + 10 GTH transceiver channels)

Table 1: HAPS-DX7 S4 system features

About the FPGA Mezzanine Card (FMC) Standard

FMC is an ANSI standard that provides a standard mezzanine card form factor, connectors, and modular interface to an FPGA located on a carrier board like the HAPS-DX. Decoupling the I/O interfaces from the FPGA simplifies I/O interface module design while maximizing carrier card reuse. FMC was developed by a consortium of companies ranging from FPGA vendors to end users.

Key Benefits of FMC include:

- Data throughput: Individual signaling speeds up to 10 Gb/s
- Latency: Elimination of protocol overhead removes latency and ensures deterministic data delivery
- Design simplicity: Expertise in protocol standards is not required
- System overhead: Simplifying the system design reduces power consumption and material cost
- Design reuse: Promotes the ability to retarget existing FPGA/carrier card designs to a new I/O

The HAPS-DX system is designed to allow two HapsTrak 3 to FMC adapter boards to be mounted. Each adapter provides the interface to a High Pin Count (HPC) connector with 400 pins and provides access to 160 single-ended I/Os and 10 differential GTH I/Os of the Virtex-7 690T FPGA device.

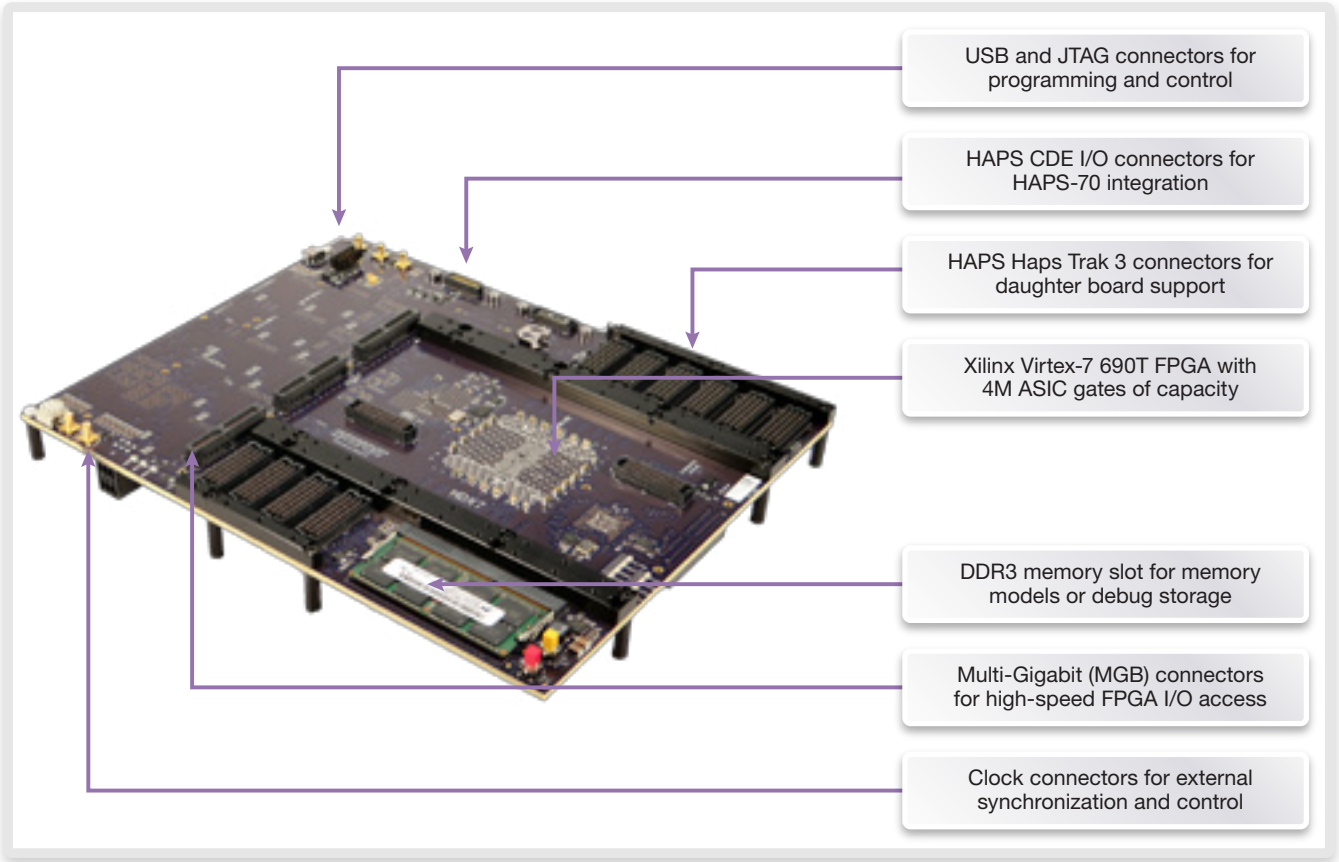


Figure 2: HAPS-DX hardware overview

Standalone Validation of ASIC RTL Blocks and IP

RTL validation by HAPS-DX is the most popular application for FPGA-based prototypes because real-time clock performance and connectivity to high-fidelity interfaces make it an ideal self-contained validation environment for a Design-Under-Test (DUT). An embedded CPU subsystem of the prototype design serves as a test jig to execute the software stack. FMC or HapsTrak 3 daughter boards provide PHY interfaces for a wide variety of protocols and systems and the HAPS-DX provides easy connectivity via JTAG to external workstations running a software IDE.

PCI Express Connected Prototyping

A prototype system directly plugged into the PCIe slot of a host workstation enables high-volume data streaming to a DUT making this a powerful validation scenario for media controllers and quality review. The HAPS-DX PCIe paddle board and end point core eases memory mapped access to the DUT.

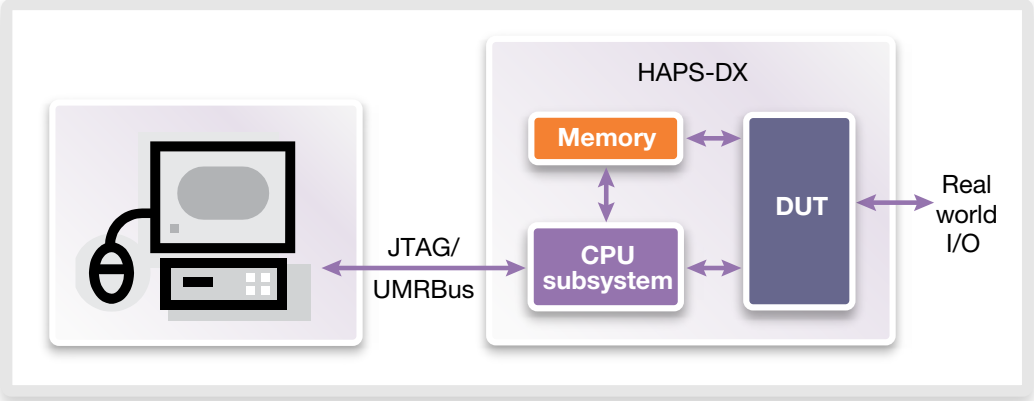


Figure 3: Standalone validation with HAPS-DX

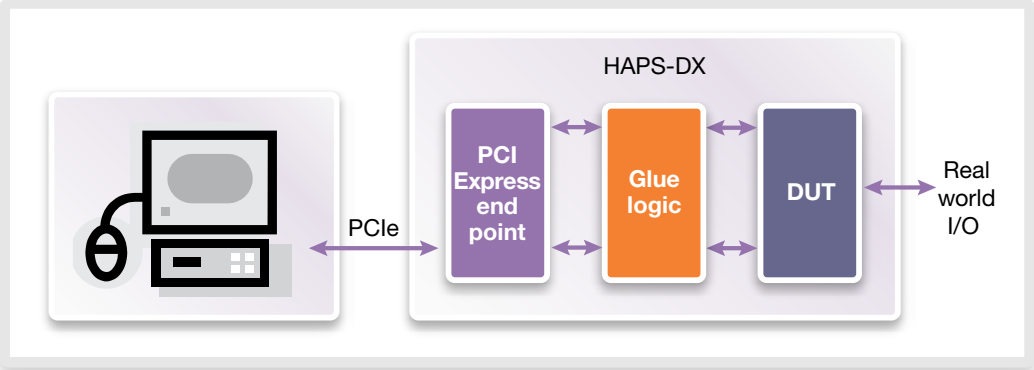


Figure 4: PCIe connected prototype with HAPS-DX

Hybrid Prototyping

A prototyping system that can mix SystemC/TLM based models with FPGA-based prototype hardware provides design teams a way to make prototypes available months sooner than traditional methods because now RTL availability is not a gating factor for bring-up. A virtual prototype communicates to the RTL DUT via bus protocol transactors to bridge loosely-timed models with cycle-accurate hardware. The DUT RTL is validated in the context of a virtual processor subsystem running a software stack comprised of OS and application software. The HAPS-DX TBV Suite (sold separately) enables hybrid scenarios mixing Synopsys Virtualizer Development Kits (VDKs) with a HAPS-DX system.

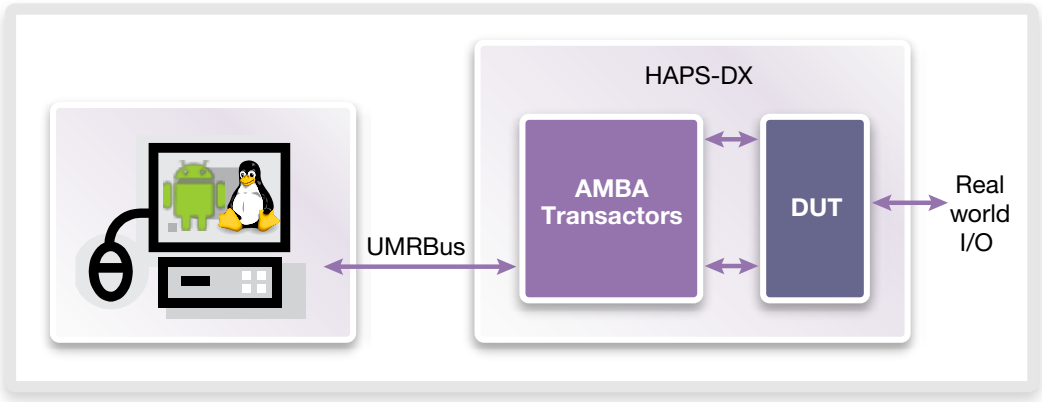


Figure 5: Hybrid prototyping with HAPS-DX

Reuse and Integration With HAPS-70

HAPS-DX is designed to serve as a module or daughter board for a HAPS-70 system when larger capacity systems are required. Reuse of prototype modules validated by standalone HAPS-DX systems helps design teams avoid long re-synthesis and place-and-route processes. The control logic of a HAPS-70 provides the communication infrastructure like user I/O, configuration, and clock/reset distribution necessary to integrate hardware systems that include HAPS-DX, FMC standard, and Synopsys HapsTrak 3 daughter board PHYs.

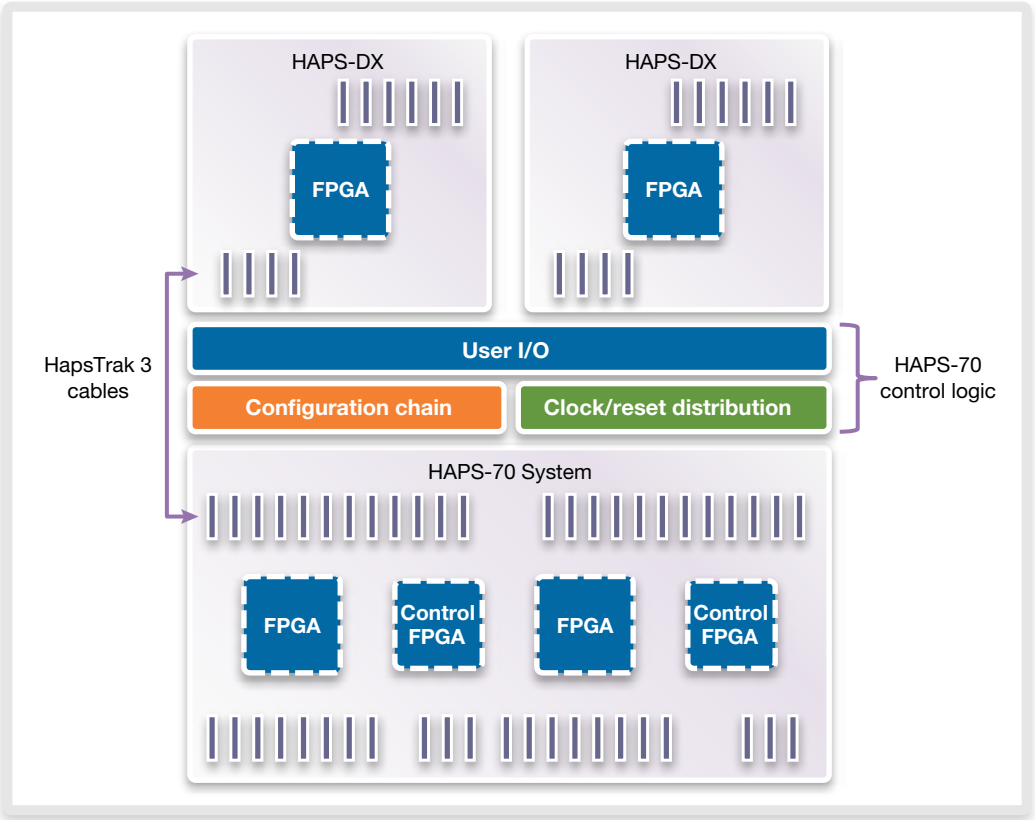


Figure 6: Modular HAPS-DX reuse with HAPS-70

HAPS-DX Synthesis and Implementation Features	Benefits
Multi-million gate capacity	No need to break up designs into many small blocks
Full HDL support	Prototype VHDL, Verilog, SystemVerilog, or mixed-language designs
Gated-clock conversion	Automatic mapping of gated-clocks into FPGA designs without source-code modification
Synopsys Design Constraint (SDC) support	Recognition of SDC speeds the migration of timing intent into the prototype
Universal Power Format (UPF) support	Infer and prototype isolation/retention logic directly from UPF format files
ASIC IP compatible	DesignWare, IP-XACT, and IEEE-P1735 support for quick migration
Incremental synthesis flow	Block-based flow supports Xilinx Vivado design preservation flow for fast turnaround
Fast HDL analysis	Fast compile mode 4 times faster with minimal quality of result impact
Continue-on-error	Reduce iteration by identifying multiple errors in single synthesis run
Post-compile netlist editor	Scriptable interface to remove or insert components prior to FPGA implementation
Encapsulate Xilinx Vivado place and route	Robust interface to FPGA backend implementation with logic netlist and design constraints
Diagnostics interface	Quickly filter and interpret HDL compiler and synthesis messages with hypertext interface
Spreadsheet-style constraint editor – SCOPE	Fast constraints setup and review relative to HDL source or graphical views
RTL and technology views - HDL Analyst	Graphical output of RTL and state machines eases review of inferred and mapped logic
TCL-based command line interface	Ease EDA tool flow automation

HAPS-DX Debug and Bring-Up Features	Benefits
Instrument design from RTL source code	Quickly select signals and code branches for sampling and/or triggering
Debug design in RTL source code	Rapid debug of results and the ability to get useful data with less debug logic
Implement state machine style triggers	Useful for creating complex triggering conditions to isolate system conditions
Enumerated data type preservation	Displays data in RTL source as symbolic data rather than bit-level, ideal for state machines
Export sample data as VCD or FSDB format	Visualize data with a variety of viewers including GTKWave, Synopsys nWave or DVE
Export debug vectors	Record in-system HAPS-DX state as test vectors for simulation and fault isolation
Unlimited sequential trigger conditions	Allows any series of events to be used as a capture trigger
Cross triggering	Triggers from one clock domain can trigger and sample in another clock domain
Pipelined debug logic	Minimal or no timing impact on original design
Area reporting of instrumentation logic	Provides accurate feedback on FPGA resource consumed
HDL Analyst integration	Graphical post-compile RTL and technology views provide expanded design access
Synopsys Verdi/Siloti integration	ESDB/FSDB data exchange for easy simulation view import and root-cause analysis
Selective sampling and multiplex groups	Quickly focus sample capture on periods of interest and maximize signal visibility
Deep trace or real-time debug	High-capacity sample storage uses on-board DDR3 SDRAM or an external logic analyzer
Workstation connectivity via UMRBus	Access Client Application Interface Modules (CAPIMs) from user programs or Tcl
HAPS aware bring-up utilities	Validate system configuration and daughter board locations for fast duplication
Incremental instrumentation option	Quickly adjust register and port connections to debug logic for faster turnaround
TCL-based command line interface	Allows automation of instrumentation or debug via scripts

Prototype Connectivity Options

A workstation connection to the HAPS-DX prototype enables a variety of use modes for monitoring, control, or even hybrid prototypes.

- **HAPS-DX UMRBus** (Universal Multi-Resource Bus) interface is a complete and reliable set of components that allow bi-directional data exchange at runtime between software (C/C++ or Tcl/TK applications) and hardware (DUT). Each HAPS-DX system provides an on-board UMRBus interface circuitry for an easy PCIe or USB connection to host workstation.
- **SCE-MI** — HAPS-DX Transaction-Based Validation (TBV) Suite product (sold separately) includes a SCE-MI standard transport infrastructure to connect untimed software models to design-under-test (DUT) models executing within a hardware system like an FPGA-based prototype. The Synopsys SCE-MI communication link is automatically generated for each channel which interconnects transactor models in a HAPS Series FPGA-based prototype to untimed or RTL C/C++/SystemC models on a workstation.
- **AMBA** — The HAPS-DX Transaction-Based Validation (TBV) Suite product (sold separately) includes a transactor library for AMBA interconnect to enable data exchange between a loosely-timed transaction-level model (TLM) and a cycle-accurate FPGA hardware implementation. The transactors give designers the flexibility to partition the SoC design between the SystemC/TLM virtual and FPGA-based prototyping environments at the natural block-level boundaries of the AMBA interconnect.

High-Performance IP and Module Prototyping

The HAPS-DX logic synthesis tool delivers rapid runtimes using incremental synthesis flows, fast synthesis mode and automated block-based design. The continue-on-error compiler feature reduces the number of iterations required during synthesis by continuing to process even in the presence of erroneous modules and by generating an error log at the end of the synthesis step. This enables designers to fix all the errors in aggregate at the end instead of restarting the synthesis cycle after each individual error is encountered and fixed. Path-group technology makes design schedules more predictable by delivering results that are reproducible from one run to the next. A block-based RTL synthesis flow integrated with Xilinx Vivado's block-based place-and-route design preservation flow, shortens iteration runtimes, and preserves verified parts of the design from one run to the next. The HAPS-DX FPGA synthesis tool set offers the most comprehensive set of automated features for implementing FPGA-based prototypes. The tool's built-in gated-clock conversion capability and full integration with the DesignWare® Library's Datapath and Building Block IP enables ASIC RTL code to be implemented in an FPGA.

HAPS-DX FPGA synthesis employs a Behavior Extracting Synthesis Technology® (BEST™) and timing-driven logic synthesis engine. With true timing driven synthesis technology, the tool works to reduce area utilization after timing requirements have been met. When maximum timing performance is required, the software can use advanced logic optimizations such as re-timing and pipelining to boost performance.

Support for the IEEE 1801-2009 Unified Power Format (UPF) standard helps ensure low power design specifications for ASIC design and verification can be prototyped by HAPS-DX. Logic synthesis automatically infers isolation cells to force known values in power down modes and retention cells to restore the save state to a system.

HAPS-DX automation software easily integrates into your current process and you can use the flow-based GUI to learn how to use the tool. All graphical commands are also recorded as Tcl commands to allow you to easily script your design flow.

Non-Intrusive and High-Capacity Debug

HAPS-DX's non-intrusive approach to debug allows you to instrument your design without any changes to your RTL code. A hierarchy display lets you navigate quickly to the design module you need, icons indicate nodes available for probes or triggers and you can simply activate probes in your HDL design by using menus or script commands.

Debug storage may be expanded by targeting the HAPS-DX on-board DDR3 memory or an external Agilent or Tektronix logic analyzer via an optional HapsTrak 3 Mictor Daughter Board. These high-capacity storage options are ideal for maximizing signal visibility or for recording long periods of runtime during complex protocol validation scenarios.

HAPS-DX debug supports popular formats for data vector exchange like VCD, FSDB, and ESDB to ease integration with verification tools like Synopsys VCS, Verdi, and Siloti.

The instrumentation of HAPS-DX supports multiple clock domain triggering and clock cross triggering for inter-domain debug. Once the target FPGA is programmed, a debugger application communicates with the FPGA via the UMRBus or JTAG interface to interactively set trigger modes and view captured data of the live system. Debug trigger operating modes include cycles, events, pulse width, and watchdog. Use the modes to add clock delays and pulse widths to logic or branch triggers.

After a trigger condition is met, sample buffer history is extracted from the live hardware. The debugger application automatically translates the hardware level signals back to register-transfer-level (RTL) source code constructs. Bits are recombined as vectors and enumerated data types. Results are annotated directly onto the RTL source view.

Platform and FPGA Device Support

HAPS-DX automation and debug software supports Linux 64-bit operating systems. New HAPS-DX systems are supported as they become available.

For more information, visit www.synopsys.com/haps.
To learn more about the FPGA-based prototyping methodology, visit www.synopsys.com/fpmm

SYNOPSYS®
Accelerating Innovation

Synopsys, Inc. • 700 East Middlefield Road • Mountain View, CA 94043 • www.synopsys.com

11/13.RD.CS3594.