

# Synthesizable Bus Functional Models

## Duolog Helix™ Based Verification IP

### Features

- Flexible, VHDL-based architecture
- Fully synthesizable bus functional model
- Microcontroller based (Helix™) with standardized BFM interface
- Simple, extensible microcontroller instruction set
- Memory or FIFO based control
- Tcl interface for easy testcase development
- Easy integration with VHDL, Verilog, Vera, e, SystemC, or SystemVerilog testbenches
- Simple clocking mechanism
- ESL support for transaction-level control
- Works with emulation and hardware co-simulation/acceleration
- Reduces communication bandwidth by abstracting signal level interface to a transaction level interface
- Small gate count
- Fast speed and performance
- Provides error monitoring and self-test capabilities
- Several modes of operation
- Well-defined protocol engine interface for rapid BFM development and easy maintenance
- Highly configurable

### Benefits

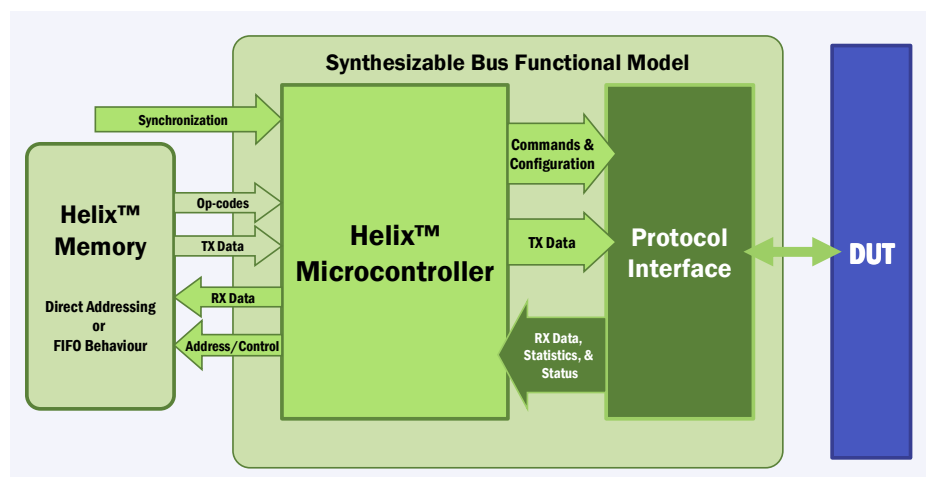
- Provides a standardized BFM interface for easier test writing and porting
- Reusable in simulation, emulation, hardware acceleration, and FPGA prototyping
- Helix™ Microcontroller accelerates emulation by pushing communication intensive operations into the hardware
- Works seamlessly with other ESL technologies
- Simplifies test-case and test-bench development

### OVERVIEW

The Duolog Helix™ based verification IP provides a standard, configurable, synthesizable, and programmable bus functional model, which can be used at all levels of verification. At the heart of its functionality is the flexible Helix™ microcontroller.

The model's versatility enables rapid testbench and testcase development. Likewise, the model is ideal for use in emulation and prototyping for early system debug and software development.

Written in VHDL, the Helix™ Bus Functional Model can easily integrate into all industry standard verification technologies.



## Description

Duolog's Synthesizable Bus Functional Model technology is built around the Helix microcontroller core. The microcontroller communicates with the protocol state machine, providing a high-level abstraction of the BFM's control, synchronization with the test-bench, error monitoring, and memory control. The microcontroller has a simple, yet extensible, instruction set which simplifies the BFM programming interface.

Protocol sequences can be encapsulated into single instructions, significantly reducing communication bandwidth with the BFM and simplifying the programming. The Helix microcontroller occupies a small gate count\*, operating across disciplines from FPGA prototyping to System-On-Chip verification.

## BFM Architecture

The Duolog Synthesizable Bus Functional Model architecture consists of 3 components: the Protocol Generator (PG), the Helix Microcontroller, and a Multi-function Memory. The PG consists of a protocol FSM, which interfaces with the memory via the Helix microcontroller. The PG can both read data from memory to transmit to the design under test, or store received data for subsequent comparison with golden reference data.

The memory has two operational modes: *Direct Addressing* and *FIFO mode*. In Direct Addressing mode, the PG can read and write to memory in an address based fashion. In FIFO mode, simple queue-based memory access is used. The memory backdoor provides access to the testbench and to the data memory, which a transactor or embedded processor can access for sending or checking received data.

## Reusability

Traditional high-level verification language BFM's have speed and software configurability advantages, but are incapable of operating at speed in emulation and prototyping for software level testing; therefore, either the BFM must be re-written to accommodate the emulation platform or interface with the emulator at the signal level, increasing the communication bandwidth with the emulator and severely limiting the operational frequency of the design in emulation.

BFMs written at an RTL level work at speed in both emulation and prototyping; however, they often suffer from non-standardized programming interfaces and require signal level control which is neither easy to interface with, nor easily ported to other test-benches, nor facilitating portable test writing.

The Helix-based Synthesizable BFM's strikes the balance between these cases. It provides reusability across all hardware verification platforms while providing a simple and standardized user interface. It easily integrates into existing testbenches, and is reusable across module and SoC level verification as well as across many different projects.

## Ease of Use and Portability

The Helix microcontroller receives its instructions from its local memory. The memory can be preloaded using Tcl instructions, allowing for fast setup of the BFM in simulation as well as in emulation. Since the memory can be preloaded in emulation, communication between the host system and the BFM in-circuit is minimal while the DUT is in operation. The Tcl interface means that test writing becomes as easy as scripting, and since Tcl is built into all standard EDA tools, it is portable between most platforms and environments. Because BFM programming is abstracted to a transaction level, tests remain portable between one project and the next.

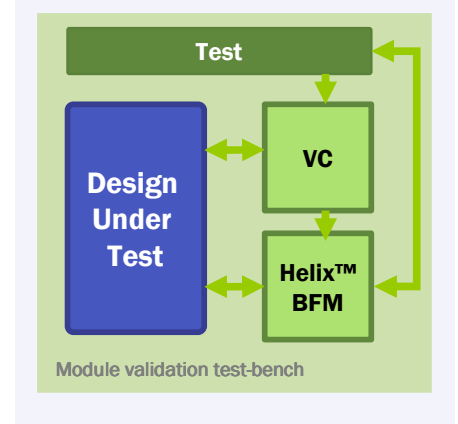
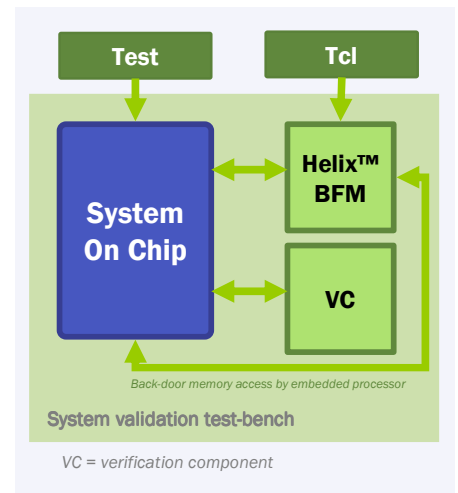
The Helix Microcontroller also has the ability to be fed instructions one-by-one, waiting until it receives its next command. This allows it to be used in a transaction based environment in which commands are sent to the BFM as the test sequences are executed. This ability allows the Helix to accommodate whatever environment it is situated in.

## Rapid Development Time

With Duolog's synthesizable BFM technology, you can focus on what matters – verifying the Design-Under-Test. Our technology simplifies the BFM development cycle by allowing engineers to focus on just the BFM protocol while providing a standardized framework that ensures reuse, a simplified programming model, and portability. BFM's can rapidly be developed since engineers focus on the Protocol-Under-Test without concerns about the framework that goes around it.

## A Proven Track Record

Synthesizable BFM technology developed by Duolog has been proven and used extensively on more than 20 large scale System-on-Chip Verification Environments. The programming ease, test portability, and reusability of the technology across the various levels of hardware verification have all contributed to the success of our customers. Over 100 BFM's have been developed by Duolog.



**Helix™ Synthesizable BFM's in situ in a testbench**  
 Many possibilities: (top) Helix™ used for module validation; (above) Helix™ used for system validation

## Evaluation Models

An UART and a SPI model are available for evaluation of the Helix synthesizable BFM technology.

## About Duolog

Founded in 1999, Duolog Technologies is a supplier of market leading ESL tools and solutions enabling SoC Flow Automation.

Our tool offering is built on Socrates, a framework which enables rapid and effective SoC integration and verification of complex designs.

Tools within Socrates include SoC Register Management, Spirit-compliant SoC Connectivity and Testbench Automation.

\* On a Xilinx Virtex II FPGA, a Helix based BFM is typically between 700-900 emulation gates.



Duolog Technologies, Nova Centre, Belfield, Dublin 4, Ireland. Tel: +353-1-217 8400 Fax: +353-1-217 8401

Email: [info@duolog.com](mailto:info@duolog.com) Web: [www.duolog.com](http://www.duolog.com)

All information is subject to change without notice. All Rights Reserved. Duolog, Helix and the Duolog logo are trademarks of Duolog Technologies Limited. Other trademarks are trademarks of their respective owners.  
 © 2007 Duolog Technologies Ltd. Registration No. 316532 – VAT No. IE6336532L