



# Microteam

ELECTRONICS • AUTOMATION • ROBOTICS • AUTOMOTIVE

# FPGA#

# FPGA DEVELOPMENT GOES AGILE

## DITCH THE TESTBENCH!

The FPGA# paradigm brings the **speed and agility** of modern software development to the world of **RTL level design for FPGAs**. It enables the developer to focus on the key functionality right from the start. This is achieved by minimizing the work and time needed for getting the first simulation to run. Complex algorithms can be tailored piece by piece, working towards the final implementation while having the system level model running early in the development cycle.

Whereas competing approaches abstract the algorithm development and modelling away from the actual VHDL implementation, FPGA# works on the RTL level. This guarantees cycle accurate equivalence to VHDL, enabling full utilization of the FPGA performance.

## BENEFITS

FPGA# speeds up the design process in many ways:

- "Code as you need": single entities can be simulated without building proper VHDL top level entity or VHDL testbenches
- Algorithms are simulated in seconds, without time-consuming VHDL synthesis
- The results are visualized exactly as needed, even on the actual end system GUI
- All real-world I/O of a Windows PC are available: Ethernet, serial buses, file I/O... There's no need to prepare special test data sets for the VHDL simulator
- Syntax checking, symbol name validation etc. at editing time with IntelliSense
- 100 % equivalence to synthesizable VHDL, including cycle accuracy



## NOT ANOTHER SYSTEMC

FPGA# is a way of writing and simulating synthesizable RTL code. By synthesizable, we mean code that is realized in an FPGA without any abstraction layer. While SystemC is a modelling framework that employs the C++ language for functional simulation of hardware, FPGA# is a fast-forward way of building the actual FPGA implementation and simulating it as early as possible.

## FAMILIAR ENVIRONMENT AND LANGUAGE

FPGA# employs tools that are familiar to most developers: Microsoft Visual Studio, C#, Windows Forms or WPF. The application is written and compiled under Visual Studio using the FPGA# Toolbox. After analysis, the resulting VHDL code is compiled using any suitable VHDL synthesis tool, depending on the chip vendor.


## WHAT WE DO

Some examples of FPGA-based customer products developed by Microteam:

- **CoaXPRESS:** A dual-link, 2 x 6.25 Gbps, CXP 1.1 compliant machine vision camera
- **Real-time FFT/IFFT:** A parallel signal processing system that computes the FFT and IFFT of a multichannel input signal in some microseconds using a low-cost FPGA. The device also features a configurable RMS power limiter with programmable decay.
- **Data Acquisition:** Several data acquisition solutions for data rates between 80 and 400 MSPS over several simultaneous, time-skewed channels

## WHAT'S YOUR APPLICATION?

Microteam has wide experience in real-time signal processing as well as in development of embedded systems since 1981. For more information about references and prices please contact [sales@microteam.fi](mailto:sales@microteam.fi)



Haarlankatu 1B 33230  
Tampere, Finland  
[microteam.fi](http://microteam.fi)