Celoxica

DK4

What's new in DK4

For DK version 4

Authors: RG

Document number: UM-2012-4

Customer Support at http://www.celoxica.com/support/

Celoxica in Europe	Celoxica in Japan	Celoxica in the Americas
T: +44 (0) 1235 863 656	T: +81 (0) 45 331 0218	T: +1 800 570 7004
E: sales.emea@celoxica.com	E: sales.japan@celoxica.com	E: sales.america@celoxica.com

1 Introduction

This manual describes the changes in the latest version of DK, and also lists the changes in the previous version.

1.1 What's new in DK 4

1.1.1 Changes in devices supported

Support has been added for Spartan 3E and 3L devices.

1.1.2 Changes in reset

You may now specify a reset to be active low, using the `active_low` specification.

You may now specify a reset to be synchronous, using the `synchronous` specificationThis means that it will occur on the next clock tick, as well as the default asynchronous (meaning it will occur immediately).

The ROC files (reset on configuration) for Xilinx devices (`xilroc.v` and `xilroc.vhd`) are no longer supplied, as Xilinx devices will now be reset automatically. If you have existing HDL files produced by earlier versions of DK, you will still need the `xilroc` files from your previous DK installation.

1.1.3 Compilation improvements

The DK compiler has the following improved features:

- Optimiser improvements
- Adder tree balancing
- Hardware constant propagation
- Improved mapping to embedded ALUs, including registers within DSP/multiplier blocks
- Memory is now used much more efficiently, meaning that larger designs can be compiled unpartitioned, in a shorter amount of time.

1.1.4 Synthesis tool options supported

Support has been added to target Synopsys Design Compiler

Targeting Leonardo Spectrum is deprecated and is no longer documented.

1.2 What's new in DK3.1

1.2.1 Changes in devices supported

Support has been added for Altera Stratix II, Cyclone II and Xilinx Virtex-4 devices.

1.2.2 VHDL and Verilog

DK outputs VHDL which conforms to `IEEE 1076.6-1999` standard for VHDL Register Transfer Level (RTL) synthesis and uses the `numeric_std` standard synthesis package rather than the `std_logic_arith` package.

Besides Verilog-1995, DK now has the ability to output Verilog which conforms to `IEEE 1364-2001` standard for Verilog hardware description language (Verilog-2001), supporting signed arithmetics and other additions to the language.

Both the VHDL and Verilog output have been updated to increase readability and similarity to the style of manually written HDL.

1.2.3 Report file

The compiler now generates XML compilation reports. These files contain messages about errors, optimisations, and other information about the build of a project.

1.2.4 Channels across clock domains

Channels across clock domains have new synchronization hardware. The timing of the synchronization hardware is controlled by four new clock specifications: `minperiod`, `resolutiontime`, `paranoia` and `unconstrainedperiod`.

1.2.5 Channels and FIFOS

Channels can now be implemented as FIFOs using the `fifolength` specification. FIFOs longer than 2 will be created either in distributed RAM (by default), or in block RAM (by applying the `block` RAM specification to them).

Example

```
chan int 8 small_FIFO with {fifolength = 1023, block = "BlockRAM"};
```

1.2.6 Compilation improvements

DK features improved compilation speed and results. Better quality of results can now be achieved through improvements to the retimer, as well as the hardware generation and optimisation modules.

1.3 What's new in DK3

1.3.1 Retiming

The retiming option moves flip-flops around in the circuit to try and meet the clock period rate specified with the rate specification in the `set clock` statement for a `main` function.

It preserves the timing for logic between clock domains and the timing for logic adjacent to interfaces, but moves flip-flops in other parts of the circuit until the respective clock periods are met. The retimer also moves flip-flops at interfaces in order to meet the `intime` and `outtime` specifications. After this, the flip-flops are moved around again to minimize the number of them in the circuit, whilst conserving the specified clock period.

To enable retiming, select the Enable retiming option on the Synthesis tab in Project Settings, or use the `-retime` option if you are using the command-line compiler.

1.3.2 Automatic mapping to embedded ALUs

A new option to target embedded ALUs on selected devices has been added.

Some FPGA devices possess embedded ALU primitives, which the compiler has the ability to target automatically. Rather than leave it up to the user to specify where special ALU units should be used, the compiler intelligently uses them where they will provide the most improvement in performance over the equivalent logic.

Currently, the following ALU primitives and configurations are supported:

- 18-bit multiplier blocks on Xilinx Virtex-II, Virtex-II Pro, Virtex-II Pro X, Spartan-3
- DSP blocks in multiplier mode on Altera Stratix, Stratix GX

To turn ALU mapping on, check the Enable mapping to ALUs box on the Synthesis tab in Project Settings, or use the `-N+alumap/-N-alumap` option if you are using the command-line compiler.

1.3.3 Automatic pipelining of RAM access

You can now speed up memory accesses to on-chip synchronous RAMs on many devices, by using pipelined memory accesses.

By default, the DK compiler uses an inverted version of the main Handel-C clock to drive on-chip synchronous memories. This allows it to conform to Handel-C's timing semantics of one clock cycle per assignment, but it can potentially halve the maximum clock rate for a design.

Handel-C can pipeline accesses to on-chip SSRAMs if you write your code in a certain way. The effect is that the memory is driven by the main (non-inverted) Handel-C clock, potentially doubling the clock rate for the design, and accesses are performed with 1 clock cycle of latency.

The transform only applies to certain devices and configurations:

- Actel: BlockRAM on ProASIC and ProASIC+
- Altera: EAB on Apex20K, Apex20KE, Apex20KC, Excalibur ARM and Mercury; EAB on ApexII, except for single-port RAMs and true dual-port RAMs; M512 on Stratix and Stratix GX, except for true dual-port RAMs; M4K on Stratix, Stratix GX and Cyclone, except for single-port RAMs and true dual-port RAMs
- Xilinx: BlockRAM on Virtex-II, Virtex-II Pro & Spartan-3 (not Virtex or Spartan-II)

To turn the transform on, check the Enable memory pipelining transformations box on the Synthesis tab in Project Settings, or use the `-N+piperam`/`-N-piperam` option if you are using the command-line compiler.

1.3.4 Improved constant multiplication/division

The logic generation for constant multipliers and dividers has been improved in terms of area and speed.

1.3.5 Changes in devices supported

Support has been added for Xilinx Virtex-II Pro X devices.

1.3.6 VHDL and Verilog

A new VHDL and Verilog output style has been added: Aldec Active-HDL.

1.3.7 Simulation changes

The netlist simulator is no longer available. This simulator has been deprecated since DK1.1.

You can no longer use Borland as a backend compiler for simulation. The alternative compilers are Microsoft Visual C++ and GCC. You can install GCC as part of the DK

installation process, or download it as part of the Cygwin package from http://sources.redhat.com/cygwin / (http://sources.redhat.com/cygwin/).

1.3.8 GUI changes

The Project Settings dialog has changed. There is a new Synthesis tab and the Compiler tab has been removed. The Debugger tab is now called the Debug tab. Some of the options are now on different tabs.

1.3.9 Changes in operating system support

Support for Microsoft NT4 is deprecated.

1.4 Using code from previous versions of DK

You should be able to use existing Handel-C code with DK3.1. Possible issues with migrating from versions of DK older than 2.0 are listed below.

Code using old Handel-C file extensions

The old extensions for Handel-C files (.c, .h, .lib, .obj) are no longer supported. The new extensions are .hcc, .hch, .hcl and .hco respectively.

Old Handel-C files with .c and .h extensions will still be recognized, but you may need to update the names of include files in your source code. Handel-C header files are now only supplied as .hch files. Libraries supplied with DK are now only supplied as .hcl files.

New location for fixed-point and standard macro libraries

The fixed-point library (fixed.hcl) and standard macro library (stdlib.hcl) are now part of the Platform Developer's Kit. You will need to update any paths referring to these libraries or their header files.

Code using devices that are no longer supported

If you have code targeting Xilinx 4000 or Actel anti-fuse devices, you will need to specify a different target.

VHDL output

VHDL files produced from Handel-C now have the extension .vhd instead of .vhdl. Old versions of the ROC files (*roc.vhdl) are no longer provided. The new ROC files for VHDL output are simroc.vhd and xilroc.vhd. You no longer need to link to a ROC file if you are targeting Altera devices.

Variable initialization

You must now explicitly assign a value to local non-static variables. If you do not do this, the results will be unpredictable. There is no default initialization value.

(Global and static variables have a default initialization value of zero.)

Co-simulating Handel-C and VHDL code

The tools for co-simulating Handel-C with VHDL and Verilog have been moved to the Platform Developer's Kit. You can download the kit from the Celoxica support web site.

The old method of co-simulating Handel-C and VHDL using `PluginModelSim.dll` is no longer supported.

DK installation directory

The default installation directory for DK is now called `DK` instead of `DK1`.

The name of the DK-installation directory environment variable has changed from `CELOXICA_DK1_HOME` to `CELOXICA_DK_HOME`. You need to update any references to this variable in your code.

1.5 Additional resources

You can find further information on the Celoxica Web site.

KnowledgeBase	Available from `http://www.celoxica/support/`. You will need your support password.
Technical library	`http://www.celoxica.com/techlib/` Contains application notes, white papers, FAQs, data sheets etc.

1.6 DK licenses

If you do not have a full DK license, you may be restricted in the type of output you can generate (e.g. EDIF, VHDL and Verilog may not be available). Your license may also restrict the devices you can generate code for, or your ability to use Handel-C interfaces. Licenses are also available for various Handel-C libraries, such as those targeting Celoxica's reconfigurable platforms.

If you do not have a license to use interfaces you can only target interfaces within any libraries you hold licenses for. You do this by using the macros provided in the libraries. For example, you might use a PAL macro to target external memory on a board.

DK library licenses are generated using RSA Data Security, Inc. MD5 Message-Digest Algorithm.

2 Index