

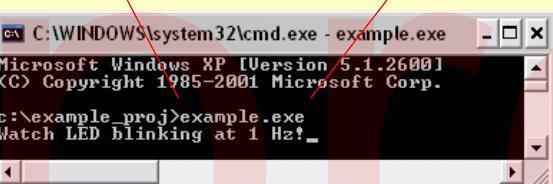
CESYS Unified Development Kit - UDK

```
bsp.h:  
-----  
#define LED_ADDR 0x08150000  
#define LED_MASK (1<<4)  
  
bsp.cpp:  
-----  
  
cout << "Watch LED blinking at 1 Hz!";  
  
// one minute blinking  
for (int i=0; i<60; ++i) {  
    // switch LED on 500 msec  
    CeDev->WriteRegister(LED_ADDR, LED_MASK);  
    Sleep(500);  
    // switch LED off 500 msec  
    CeDev->WriteRegister(LED_ADDR, 0);  
    Sleep(500);  
}
```

Standardized, universal
CESYS USB/PCI(e)
Software API
'CeUni.lib'

High Speed
Data Link

FPGA
Development



PCI
EXPRESS®

PCI

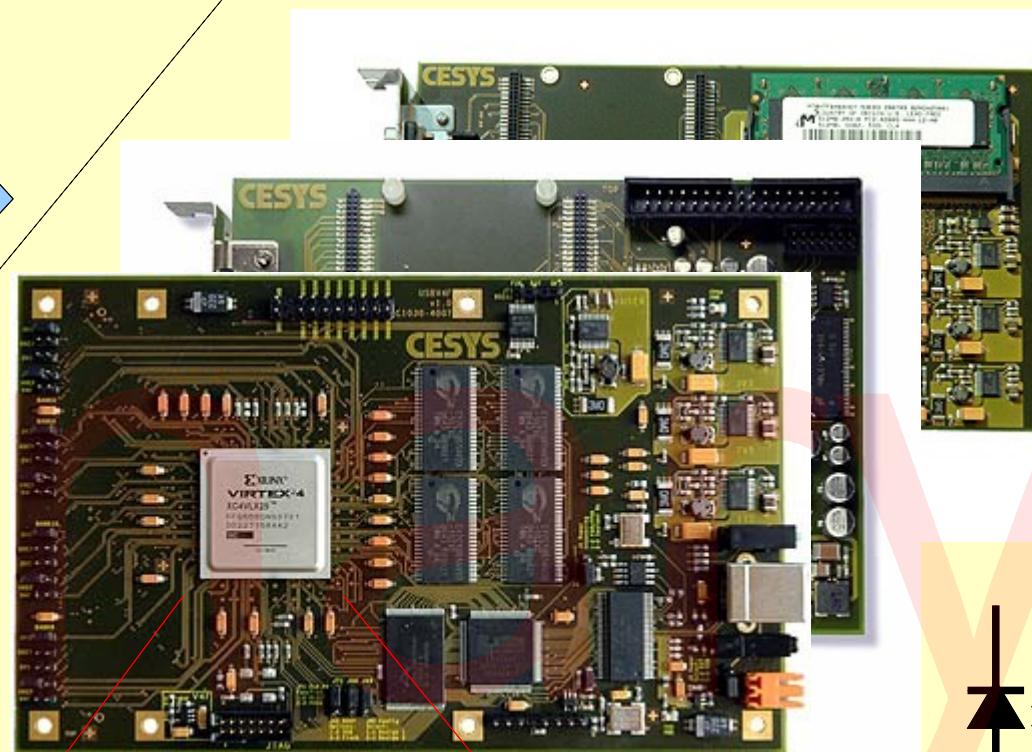
Hi-SPEED
USB

Software
Development

Industry Standard,
Transparent, Exchangeable
Data Channels for Soft-/Hardware-Developers
treated as simple Black-Boxes!
PCI(e)/USB-complexity
completely hidden from end users!

Standardized,
WISHBONE based, 32 Bit,
IP-Core Interconnection Logic
used in CESYS FPGA
example designs

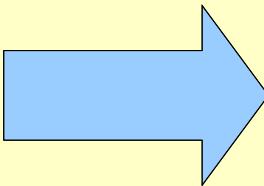
```
package bsp_pkg.vhd;  
-----  
  
constant LED_ADDR : std_logic_vector(31 downto 0) := x"08150000";  
constant LED_MASK : natural := 4;  
  
entity/architecture bsp.vhd:  
-----  
  
-- wishbone write cycle  
if (wishbone.master.cyc and wishbone.master.stb and wishbone.master.we) = '1' then  
    -- wishbone address decoding  
    case wishbone.master.adr is  
        when LED_ADDR =>  
            led <= wishbone.master.dat(LED_MASK);  
        end case;  
    end if;
```



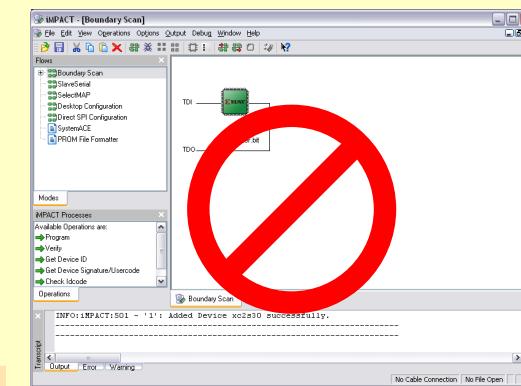
CESYS easy FPGA configuration solutions

Features:

- Download bitstream directly to FPGA or on-chip configuration FLASH
- No change of bitstream data format for FLASH programming needed
- Configuration via existing highspeed standard data channels USB or PCI(e)
- Use a simple CESYS API call in your custom software application
- Use precompiled, ready-to-use CESYS development and prototyping software tools CESYS-Mon, CESYS-Lab and CESYS-FPGA-Design-Converter
- Compile and merge one or more FPGA bitstreams to your software binary executable to create single click applications with FPGA auto-configure functionality:
Software Binary + FPGA Bitstream = „do_all_4me.exe“



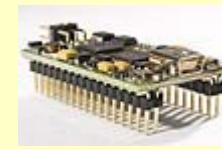
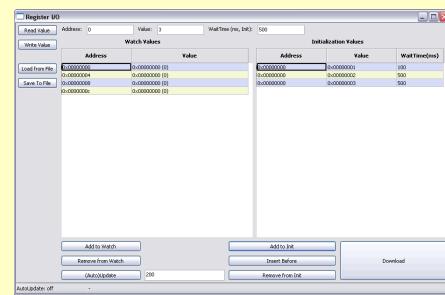
No JTAG connection or additional 3rd-party hardware/software programming tools (like parallel cable/Xilinx ISE impact) required!



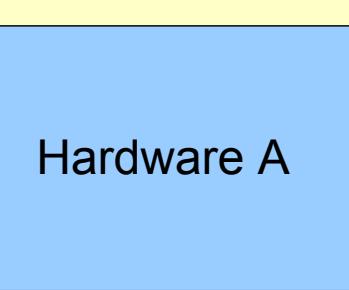
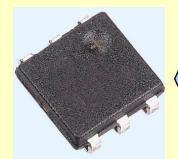
DS097_01_112601

Demonstration:

Single software application
may control several different
hardware configurations



Connect serial number chip or PROM
with unique ID to same FPGA pin in
every hardware configuration!



Hardware A

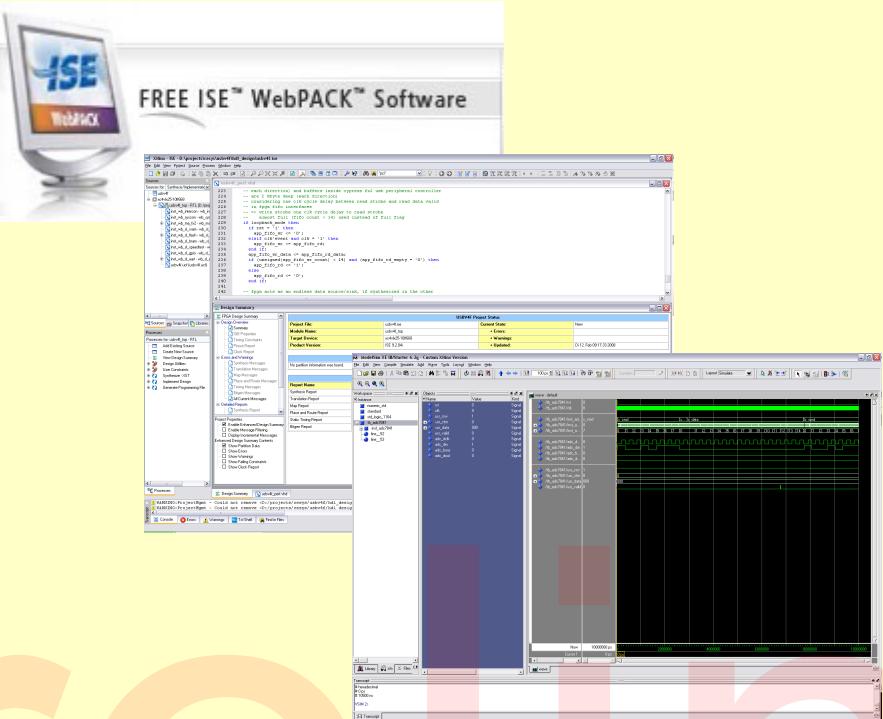
Hardware B

Hardware C

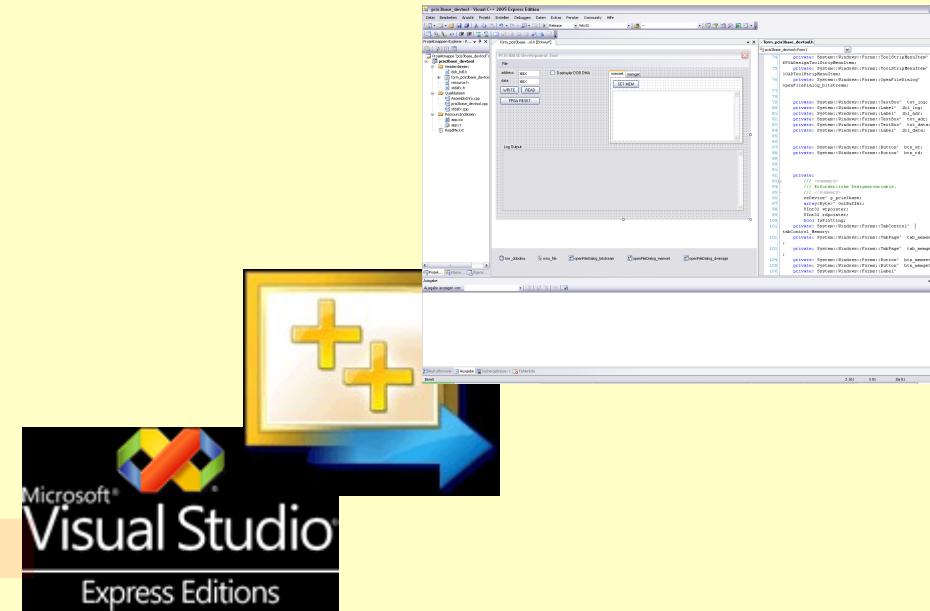
- ! • Dynamic FPGA design reconfiguration is possible
• during software application runtime!
• Example:
 - Load common design for all hardware configurations
 - Checking hardware version by i. e. reading out serial number or eeprom
 - Load appropriate design for hardware a, b, c, ...

**Straightforward, easy-to-use workflow using
Xilinx ISE WebPack, Modelsim XE Starter and MS VC++ Express Edition at no additional costs!**

FREE, DOWNLOADABLE Simulation, Synthesis and Place&Route Tools

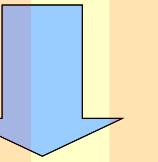


FREE, DOWNLOADABLE Software Development Tools

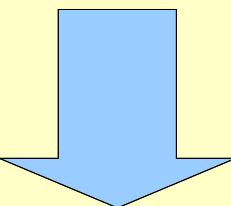
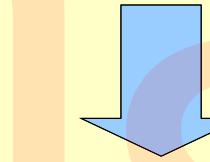


preliminary

FPGA binary bitstream „bsp.bin“



Software Application „bsp.exe“



User specific Hard- and Software-Application