



# Xylon logicBRICKS™ Integration into the Xilinx Platform Studio and EDK



This document presents a short overview of Xilinx EDK SW tools and Xylon logicBRICKS™ IP library's integration into this environment. It does not exhaustively explain technical details, and user interested in more information should consult related Xilinx and Xylon information's sources.

```
VARIABLE Col_int : INTEGER := 0;  
VARIABLE Col_vec, Col_temp : BIT_VECTOR (col_bits - 1 DOWNTO 0) := (OTHERS => '0');  
BEGIN  
  -- Advance Burst Counter  
  Burst_Counter := Burst_Counter + 1;  
  -- Burst Type  
  IF Burst_Type = '0' THEN  
    Col_int := Col_int + 1;  
    TO BITVECTOR (Col_int, Col_temp);  
    ELSIF Mode = '0' THEN  
    TO BITVECTOR (Col_int, Col_temp);  
    Col_temp (2) := Col_vec (2) XOR Col_brst (2);  
    Col_temp (1) := Col_vec (1) XOR Col_brst (1);  
    Col_temp (0) := Col_vec (0) XOR Col_brst (0);  
  END IF;  
  -- Burst Read/Write  
  IF Burst_Mode = '0' THEN  
    Col := Col_temp;  
  ELSE  
    Col := Col_vec;  
  END IF;  
  -- Burst Read/Write  
  IF Write_burst_mode = 1 AND Data_in_enable = 1 THEN  
    Data_in_enable := 0;  
  END IF;
```

**Is there  
anything more  
logical  
than  
logicBRICKS™ ?**

**logicBRICKS™ ?**



The following are registered trademarks of the Xilinx corporation: Xilinx and the Xilinx logo, Virtex™-4, Virtex™-II Pro, ChipScope™ Pro, ISE™, MicroBlaze™.

The following are registered trademarks of the IBM corporation: IBM and the IBM logo, PowerPC™, and CoreConnect™.



Designed by XYLOX


# Xilinx Platform Studio and EDK



- The Xilinx Embedded Development Kit contains the Platform Studio Tool Suite (XPS), EDK IP cores, design examples, ...
- Xilinx's Award Winning XPS is an integrated development environment for quick custom embedded platform creation
- The XPS enables easy reuse of design modules and IPs, and smooth control of HW/SW partition
- HW/SW modules are configurable through XPS GUI
- Explored within the EDK, designers' ideas can be implemented by Xilinx ISE™ software
- Full integration with third-party simulators, and straightforward testing & verification on development hardware
- The fastest path from an idea to the final product



# The EDK HW Related Details



The EDK supports CoreConnect™ bus architecture. The CoreConnect™ is an IBM-developed on-chip bus architecture that enables interconnections of IP cores from multiple sources. It includes the Processor Local Bus (PLB), the On-chip Peripheral Bus (OPB), a bus bridge, arbiters, and a Device Control Register (DCR) bus.

The Xilinx soft-CPU MicroBlaze™ is a flexible 32-bit RISC CPU. **MicroBlaze** It is a part of the EDK. Its usage is royalties free, and the processor cannot never become obsolete!

**PowerPC™** The IBM PowerPC™ 405 core is a hard 32-bit RISC CPU core immersed directly into the Xilinx FPGA fabric. The PowerPC™ is currently supported in the Virtex™-4 and Virtex™-II Pro families.



# The EDK SW Related Details



A part of the XPS is the Platform Studio Software Development Kit (SDK). It is an intuitive environment for software application development, profiling and debugging – based on Eclipse IDE.

The Xilinx Microprocessor Debugger (XMD) is also the part of the EDK. It can be used exclusively for debugging, or as a link between the GNU debugger and the target board.

Embedded systems' SW development is fully supported by GNU Software Development Tools:

- C/C++ compiler for MicroBlaze™ and PowerPC™ (gcc)
- Debugger for MicroBlaze™ and PowerPC™ (gdb)
- Other GNU utilities



# The EDK Miscellaneous

Previous slides bring out important EDK's features. There are many other features not mentioned in this document. Users are highly advised to check out available Xilinx resources at [www.xilinx.com](http://www.xilinx.com).

The EDK also includes various tools for development, debug and verification. Some of these tools are:

- Bus Functional Model Simulation (BFM)
- Base System Builder Wizard
- XMD – Xilinx Microprocessor Debug engine
- Graphical memory map manger
- ChipScope Pro hardware debugger
- Instruction set simulator for PowerPC and MicroBlaze
- etc.



The EDK offers various tools for work with numerous development HW boards. The tools can be also used with the Xylon's logiCRAFT2 Evaluation/Development platform.



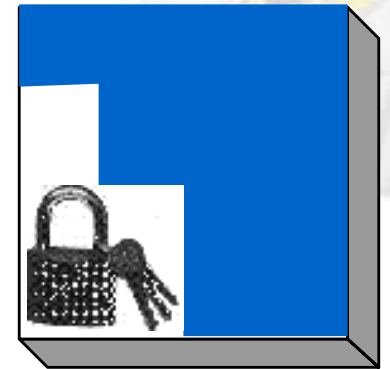
# The Xylon logicBRICKS™

- The logicBRICKS™ is a library of IP cores optimized for Xilinx FPGA devices
- All logicBRICKS™ IPs are fully embedded into the Xilinx EDK tools
- Tremendously shortened integration & verification time
- No particular skills beyond general Xilinx tools knowledge are required



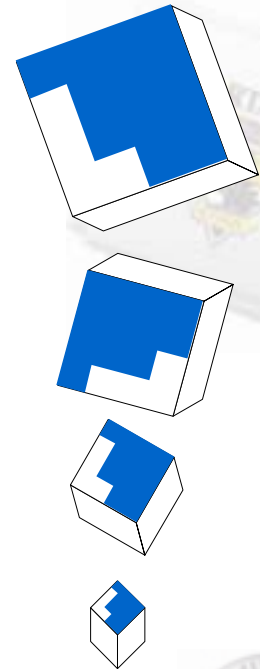
# Standard vs. Evaluation logicBRICKS™ IP

- Xylon provides evaluation logicBRICKS™ IP versions on request
- There are no structural and functional differences between standard and evaluation IP versions
- Both cores' versions come in encrypted forms, but the evaluation IPs are protected in ways that keep Xylon's intellectual property safe
- A risk free evaluation of IPs throughout all steps of the FPGA design flow

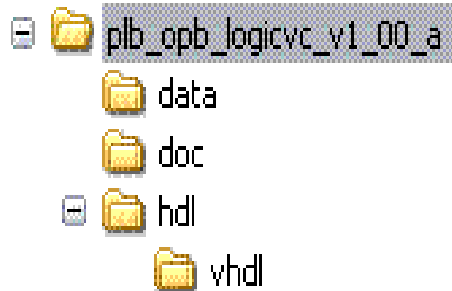


# The logicBRICKS™ IPs' Configurability

- All logicBRICKS™ IPs can be configured by changing various parameters through the EDK's GUI
- Versatile parameters' values tune up the IP's features set
- The IP's FPGA slice consumption depends on chosen configuration. Dropping out of unneeded features decreases the slice consumption.
- logicBRICKS™ software drivers are automatically changed in a way that reflects HW configuration changes



# The logicBRICKS™ IPs' HW Structure



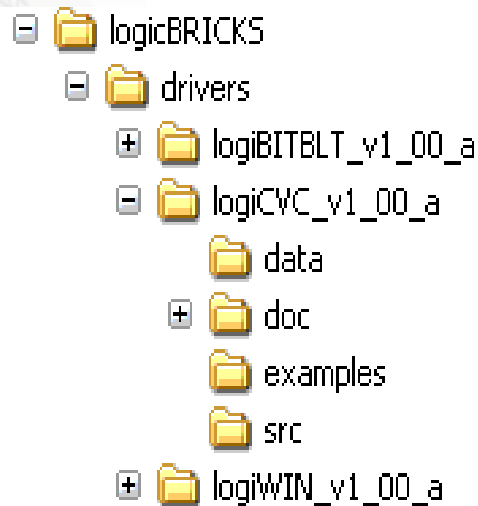
The logicBRICKS™ IP cores' HW part is structured as follows:

- DATA – contains .mpd and .pao definition files. If it is necessary, a .bbd file might be included
- DOC – related documentation
- HDL – VHDL encrypted source files

Cores should be stored in one of user's IP repositories!



# The logicBRICKS™ IPs' SW Structure



The logicBRICKS™ IP cores' SW part is structured as follows:

- DATA – contains .mdd and .tcl definition files that configures SW driver accordingly to the selected features set
- DOC – related driver's documentation
- SRC – driver's C sources

Drivers should be stored in one of user's IP repositories!



# The logicBRICKS™ Simulation Libraries



- The logicBRICKS™ IP cores are always delivered with precompiled simulation libraries

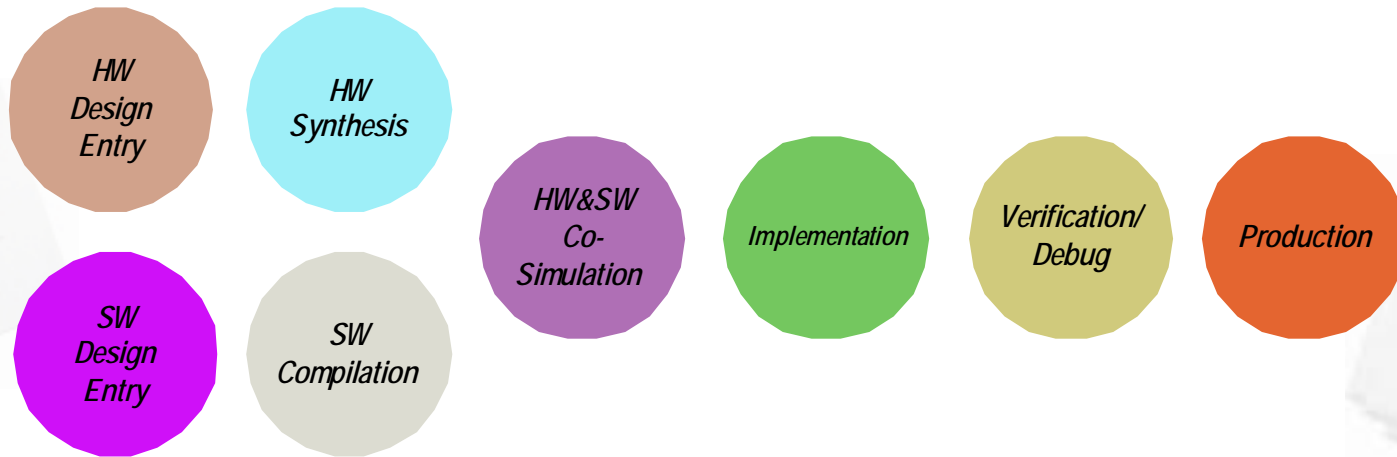
- The precompiled libraries enables flexible simulation that reflects all changes in IP's configuration
- Precompiled libraries must be placed to user's EDK simulation libraries' repository by Xilinx program *compedklib*

Run *compedklib* using the following syntax:

```
Compedklib -X <complxib_output_directory>  
-E <compedklib_output_directory>  
-lp <user_core_library>  
-o <compedklib_output_directory>
```



# The EDK Design Flow



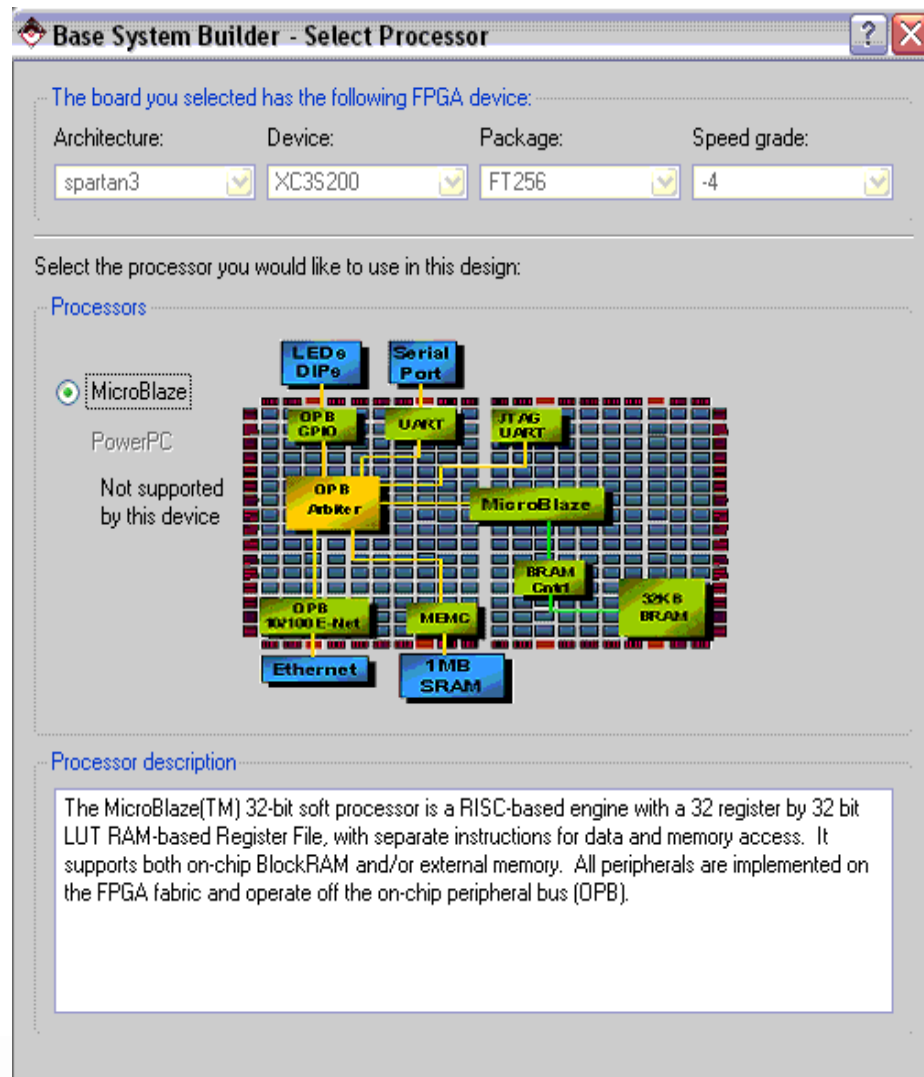
- The EDK is an integrated environment for HW & SW co-design. The whole design flow, except the simulation's execution, is covered by Xilinx EDK and ISE software packages.

- HW&SW Co-Simulation is prepared within the EDK, but requires third-party HDL simulator

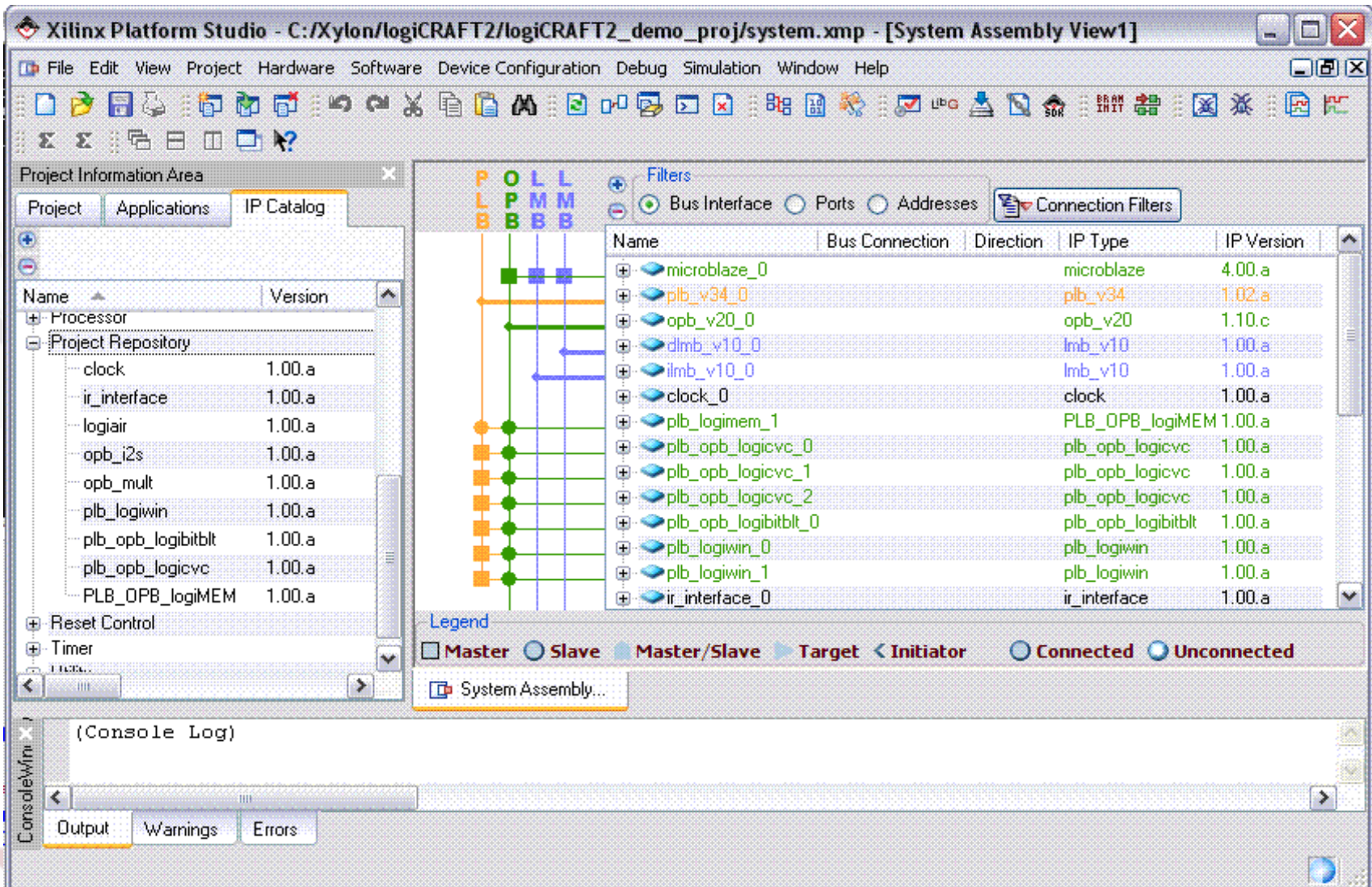
- Implementation can be done within the EDK, but is supported by the Xilinx ISE

# Start EDK Design with logicBRICKS™

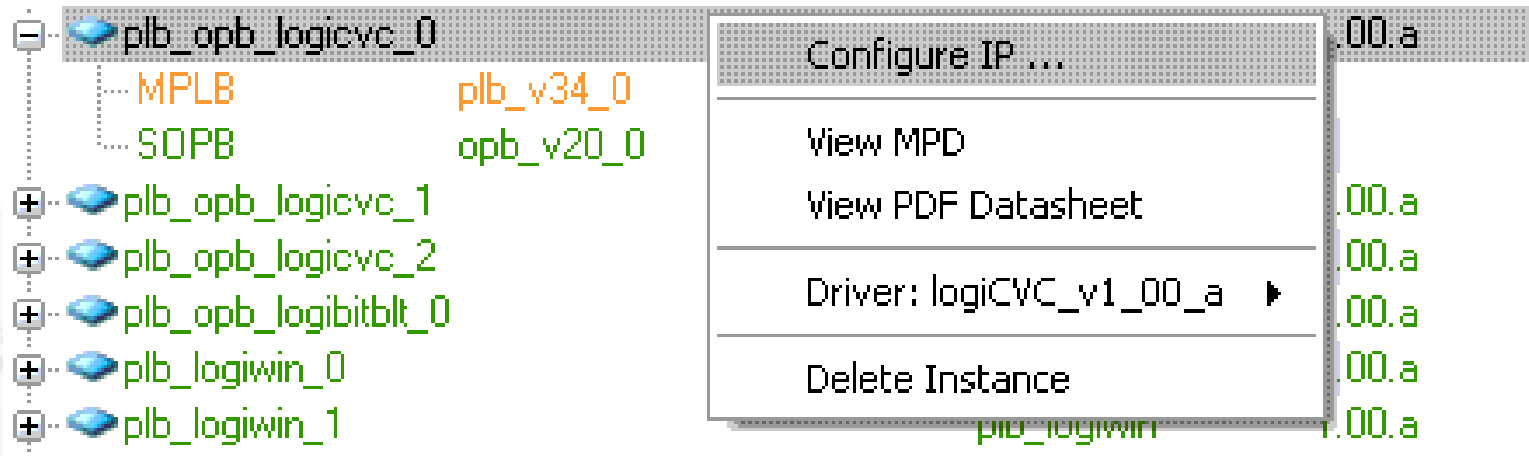
- logicBRICKS™ IP cores behave as all other EDK IPs
- Available EDK Design Entry methods are very productive. Designers can build different SoRC architectures by using IPs and bus modules
- There is no error-prone HDL handwriting
- The picture presents Base System Builder (BBS) wizard that automates system building on defined development boards.
- logiCRAFT2 support for BBS will be available soon!



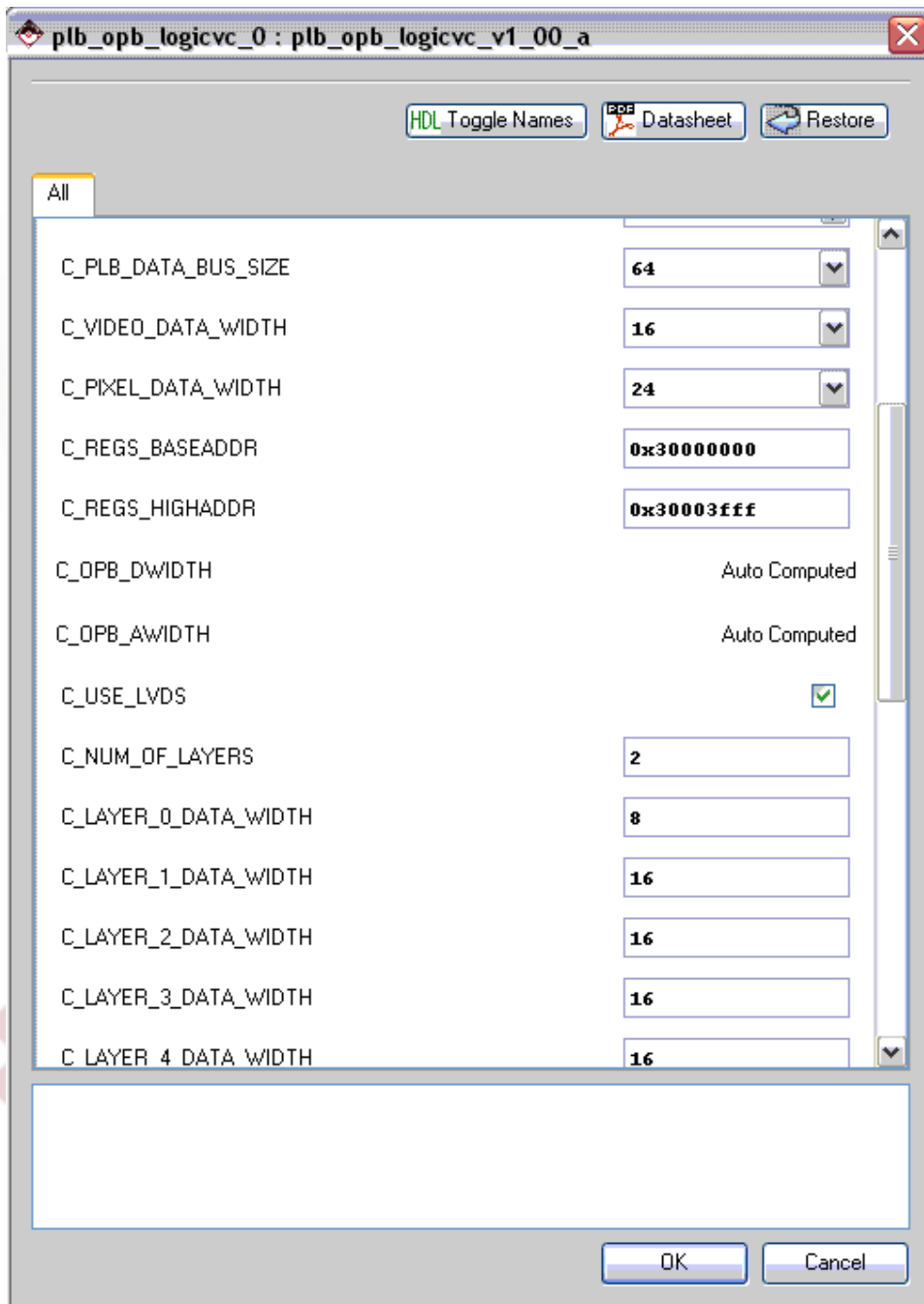
SoRC = System On Re-Programmable Chip



The picture shows XPS's workspace with opened System Assembly View on the right side. Selected IPs from IP Catalog tree in the left pane are connected into a single system. Connectivity, addressing, etc. are set in the System Assembly View.



- The picture shows a part of System Assembly View with three different instances of Xylon's logiCVC Compact Video Controller
- A single mouse right-click operation opens an IP configuration dialog box
- The SoRC bus architecture is presented in a simple manner – the above picture indicates that the logiCVC connects to the PLB and the OPB bus
- Access to documentation
- SW driver settings



- The 'Configure IP' option opens dialog box that enables an easy IP setting
- Shown are IP parameters that can be set by the user
- Input parameters' ranges must be within defined ranges that are automatically checked by the EDK



Filters:  Bus Interface  Ports  Addresses  Filters (Applied)

Name	Net	Direction	Class	Sensitivity	Range	IP Type	IP Version
opb_v20_0						opb_v20	1.10.c
dmb_v10_0						lmb_v10	1.00.a
ilmb_v10_0						lmb_v10	1.00.a
clock_0						clock	1.00.a
plb_logimem_1						PLB_OPB_logiMEM	1.00.a
plb_opb_logicvc_0						plb_opb_logicvc	1.00.a
... PLB_Clk	plb_clk	I	Clk				
... OPB_Clk	opb_clk	I	CLK				
... vcdivsel	vcdivsel_0	0			[1:0]		
... vclkssel	vclkssel_0	0			[2:0]		
... en_vdd	en_vdd_0	0					
... en_blight	en_blight_0	0					
... en_vee	en_vee_0	0					
... enable_m	enable_m_0	0					
... d_pix	d_pix_0	IO			[C_PIXEL_DATA_WIDTH-1:0]		
... hsync	hsync_0	IO					
... vsync	vsync_0	IO					
... pix_clk	pix_clk_0	IO					
... blank	blank_0	IO					
... disp_aux	No Connection	IO					
... line_2k	line_2k_0	0					
... bpp16_8	bpp16_8_0	0					
... overlay_en	net_gnd	I					
... overlay_mux	overlay_mux_0	0					
... e_vsync	net_gnd	I					
... e_hsync_fall	net_gnd	I					
... e_video_not_pre...	net_vcc	I					
... e_video_synchr...	e_video_synchronized_00						
... vclk	vclk	I					

System Assembly... BLOCK DIAGRAM

The picture shows IP port connections that can be set in System Assembly View – tab Ports. Defined IP ports are automatically shown on the left side, and the user must type-in internal nets' names in the column 'Net'.

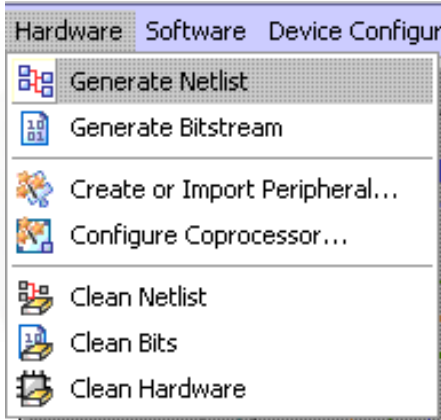
Filters:  Bus Interface  Ports  Addresses

Instance	Name	Address	Base Address	High Address	Size	Lock	ICache	DCache	Bus Connection	Dir
opb_v20_0					U	<input checked="" type="checkbox"/>				
plb_v34_0	SDCR		0b0000000000	0b0000001111	16	<input checked="" type="checkbox"/>			No Connection	
d1mb_bram_if_cntrl	SLMB		0x00000000	0x00001fff	8K	<input checked="" type="checkbox"/>			d1mb_v10_0	
ilmb_bram_if_cntrl	SLMB		0x00000000	0x00001fff	8K	<input checked="" type="checkbox"/>			ilmb_v10_0	
plb_opb_logibitblt_0		BB	0x10000000	0x17ffffff	128M	<input checked="" type="checkbox"/>				
plb_opb_logicvc_0		VM	0x10000000	0x10ffffff	16M	<input checked="" type="checkbox"/>				
plb_logimem_1	SOPB	OPB	0x10000000	0x17ffffff	128M	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	opb_v20_0	
plb_logimem_1	SPLB	PLB	0x10000000	0x17ffffff	128M	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	plb_v34_0	
plb_logiwin_0	MPLB	VM	0x10200000	0x103fffff	2M	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	plb_v34_0	
plb_opb_logicvc_1		VM	0x11000000	0x11ffffff	16M	<input checked="" type="checkbox"/>				
plb_logiwin_1	MPLB	VM	0x11200000	0x113fffff	2M	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	plb_v34_0	
plb_opb_logicvc_2		VM	0x12000000	0x12ffffff	16M	<input checked="" type="checkbox"/>				
plb_opb_logicvc_0	SOPB	REGS	0x30000000	0x30003fff	16K	<input checked="" type="checkbox"/>			opb_v20_0	
plb_opb_logicvc_1	SOPB	REGS	0x30004000	0x30007fff	16K	<input checked="" type="checkbox"/>			opb_v20_0	
plb_opb_logicvc_2	SOPB	REGS	0x30008000	0x3000bfff	16K	<input checked="" type="checkbox"/>			opb_v20_0	
plb_logiwin_0	SOPB	REGS	0x3000c000	0x3000c0ff	256	<input checked="" type="checkbox"/>			opb_v20_0	
plb_logiwin_1	SOPB	REGS	0x3000d000	0x3000d0ff	256	<input checked="" type="checkbox"/>			opb_v20_0	
plb_opb_logibitblt_0	SOPB	REGS	0x3000e000	0x3000e0ff	256	<input checked="" type="checkbox"/>			opb_v20_0	
debug_module	SOPB		0x50000000	0x500000ff	256	<input checked="" type="checkbox"/>			opb_v20_0	
opb_timer_0	SOPB		0x50001000	0x500010ff	256	<input checked="" type="checkbox"/>			opb_v20_0	
opb_uartlite_0	SOPB		0x50002000	0x500020ff	256	<input checked="" type="checkbox"/>			opb_v20_0	
opb_uartlite_1	SOPB		0x50002100	0x500021ff	256	<input checked="" type="checkbox"/>			opb_v20_0	
opb_i2s_0	SOPB	REGS	0x50006000	0x500060ff	256	<input checked="" type="checkbox"/>			opb_v20_0	
opb_iic_0	SOPB		0x50667000	0x506671ff	512	<input checked="" type="checkbox"/>			opb_v20_0	
opb_emc_flash	SOPB	MEMO	0x60000000	0x607fffff	8M	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	opb_v20_0	

System Assembly... BLOCK DIAGRAM

Setting SoRC's address space is the last step in the design entry. Addresses can be also automatically generated. Address ranges are automatically checked by the EDK.

# The EDK Design Synthesis



- All necessary IPs must be configured and interconnected as it is described in the previous chapter
- ‘Generate Netlist’ menu option starts design’s netlist synthesis

## The EDK Design Implementation

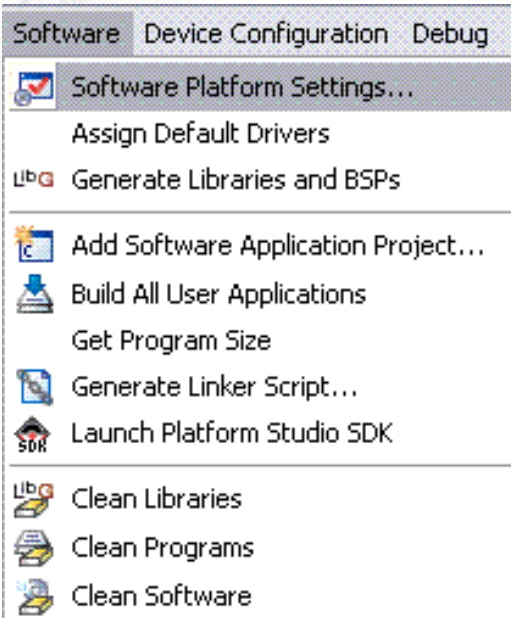
- The design’s bitstream can be generated upon a successful netlist synthesis process (see picture above)
- The bitstream is an actual FPGA configuration file
- The bitstream can be generated within the EDK, or the netlist can be exported for implementation into the ISE™

Implementation

SW  
Design  
Entry

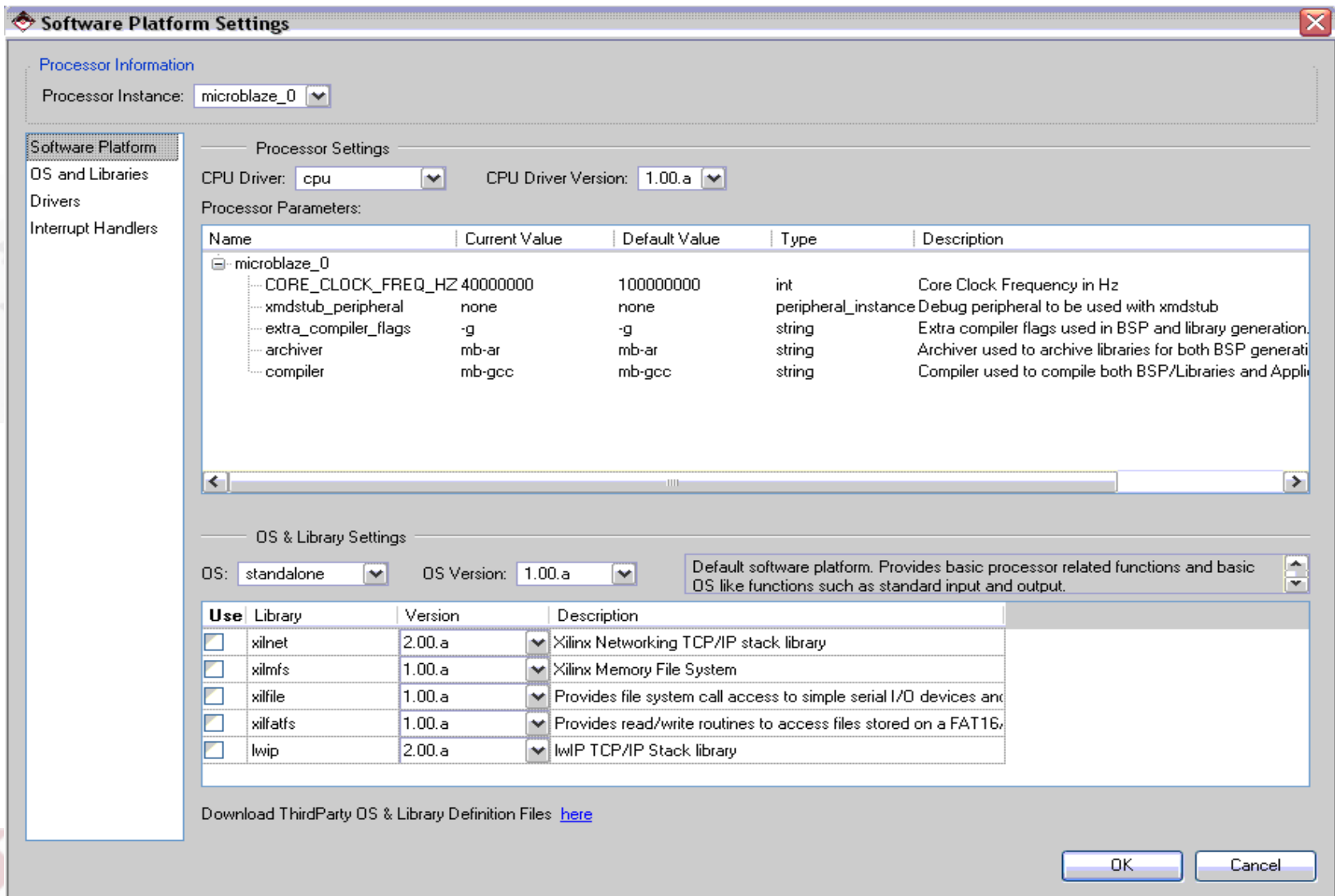
SW  
Compilation

# The EDK SW Design Entry



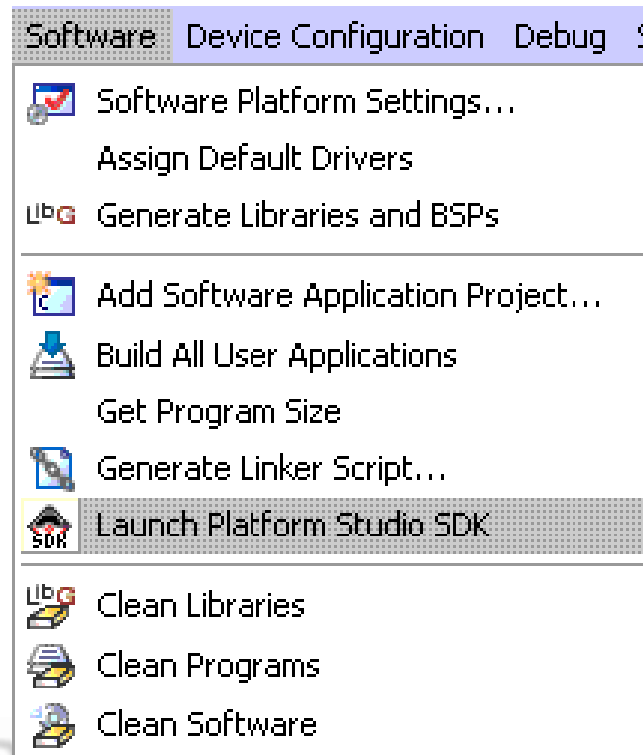
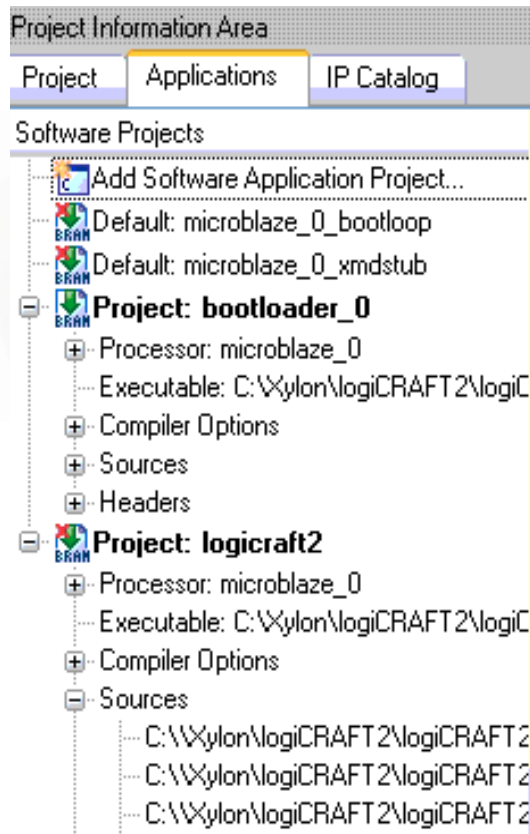
- The EDK is an integrated environment for the HW/SW co-design
- It contains all tools and features necessary for productive software development
- The pull-down menu lists SW actions that can be made within the EDK
- The user application can be fully embedded within the FPGA fabric!





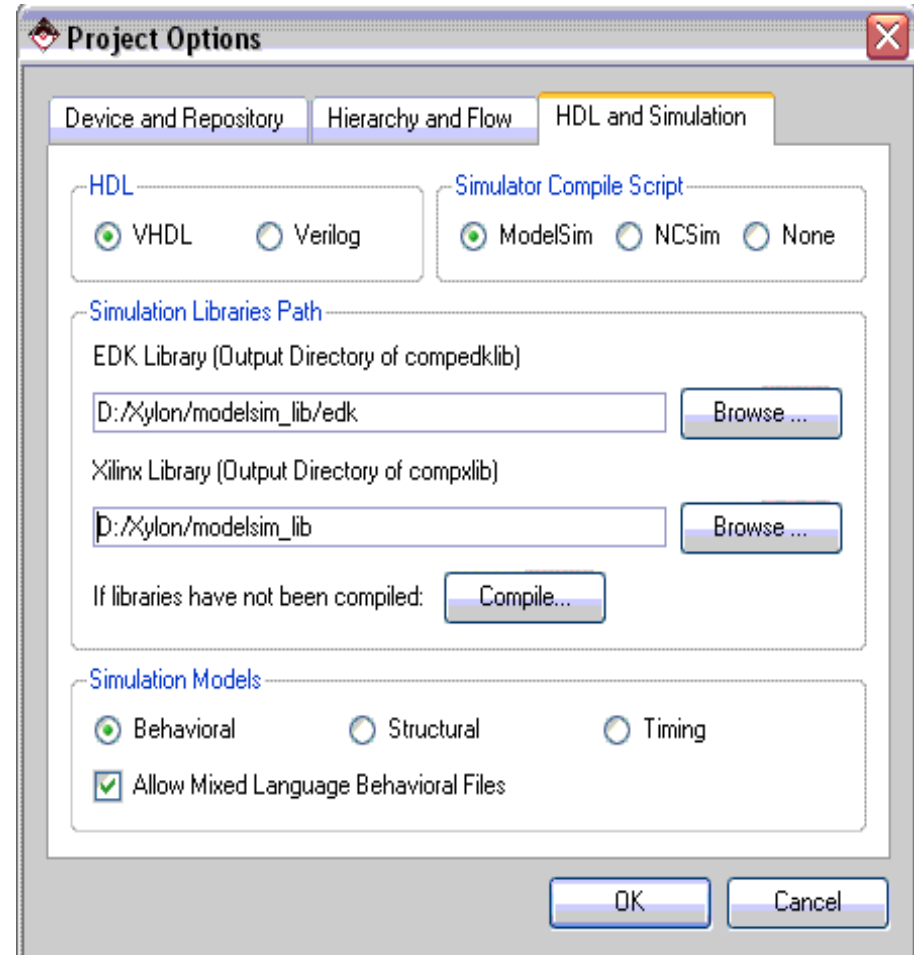
Software Platform Settings can be set in well-structured dialog boxes. Libraries, drivers, interrupt handlers, and other details can be set in the dialog box from above.

- SW applications can be handled from Project Information Area (left),
- or the Developer can use the Platform Studio SDK (bellow)



# The EDK HW-SW Co-Simulation

- Different kinds of simulation are possible at different stages of the embedded device's development: behavioral, structural, and the timing simulation.
- Tight integration of HW and SW tools enables previously hardly achievable design methods
- For an example, simulation's preparation of an FPGA design featuring the MicroBlaze™ soft-CPU, various peripheral IPs, and a SW application stored in Block RAMs, requires a single mouse-click
- Prepared simulation files must be imported in third party simulator!



# The EDK Verification/Debug

Besides traditional verification/debug techniques applicable to embedded electronic design, some newer and exciting techniques can be applied due to a versatility of the FPGA technology.



Xilinx ChipScope™ Pro collects real-time verification tools that provide on-chip debug. The ChipScope™ Pro inserts IP cores (logic analyzer, bus analyzer, etc.) directly into an inspected FPGA design. Combined with the software application, these IPs build a system that enables FPGA's internal signals capturing and analysis – the logic analyzer.

Xilinx offers different JTAG cables that can be used for in-system re-programmability, software download, HW and SW real-time debug, flash memory programming, and so on.

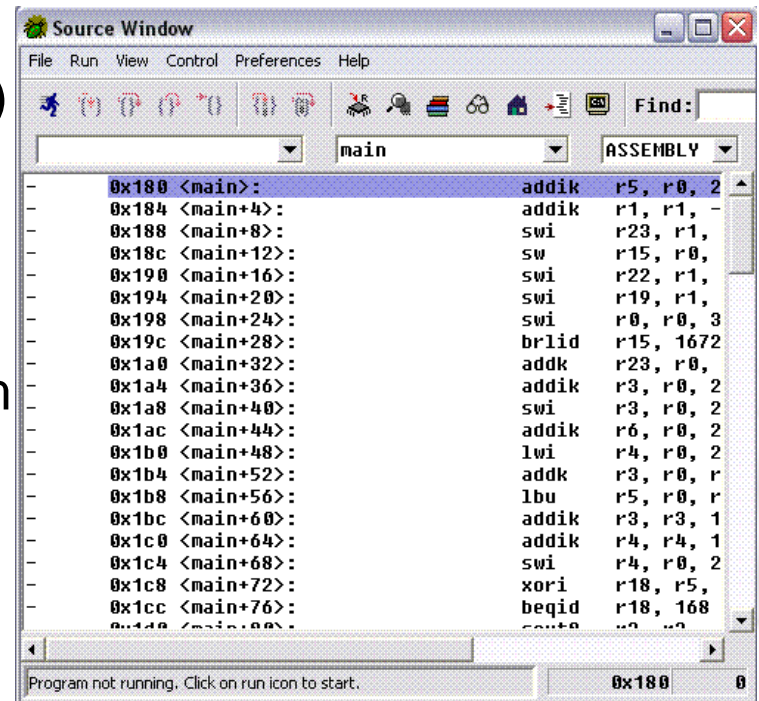


The JTAG enables an interface to internal FPGA states, and does not waste silicon resources available to the user.

The Xilinx Microprocessor Debugger (XMD) enables debugging and verification of programs running on:

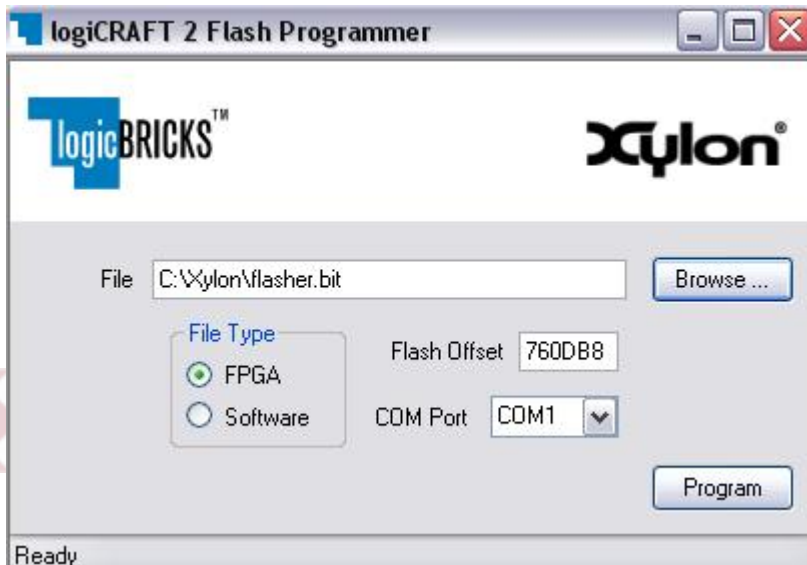
- a real hardware
- Cycle-accurate ISS
- MicroBlaze cycle-accurate Virtual Platform

The XMD also supports the GNU debugger TCP protocol that enables a remote debugging.



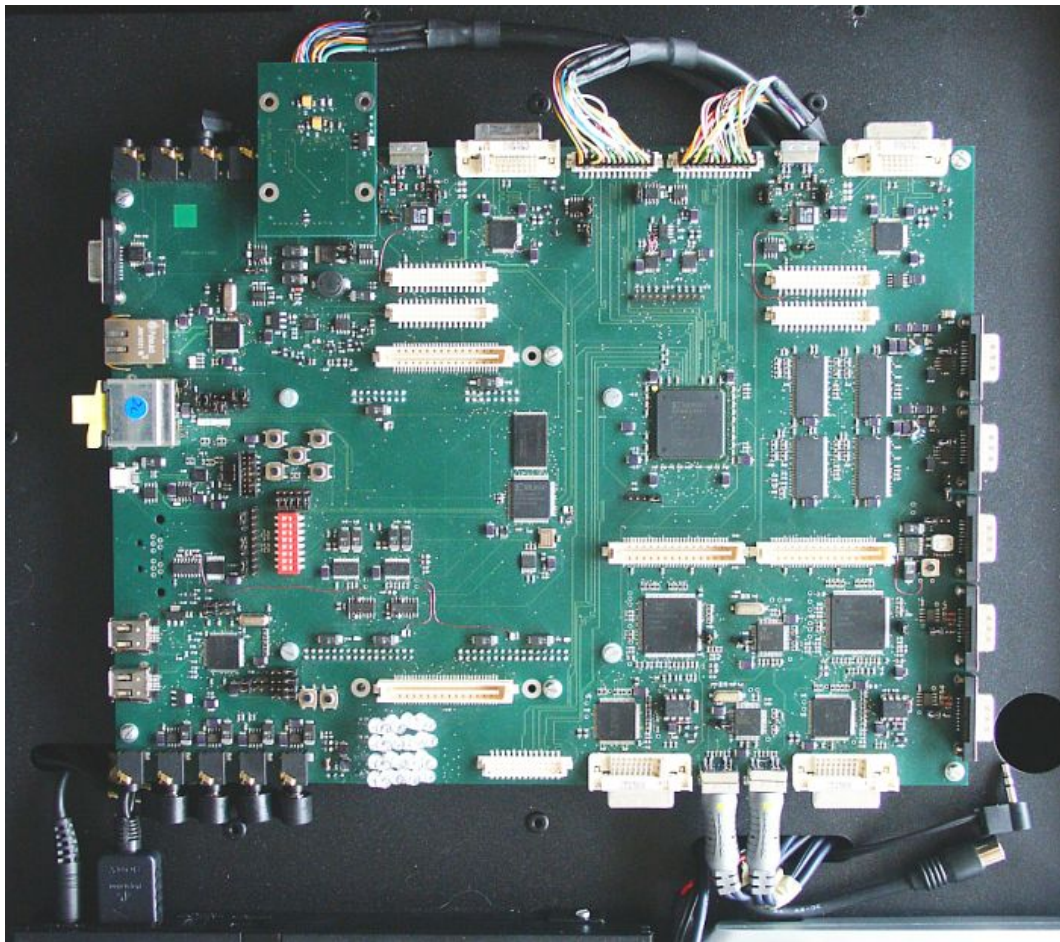
The screenshot shows the Source Window in XMD. The window title is "Source Window" and it has a menu bar with "File", "Run", "View", "Control", "Preferences", and "Help". Below the menu bar is a toolbar with various icons. The main area displays assembly code for the "main" function. The code is as follows:

```
0x180 <main>:      addik  r5, r0, 2
0x184 <main+4>:    addik  r1, r1, -
0x188 <main+8>:    swi   r23, r1, -
0x18c <main+12>:   sw     r15, r0,
0x190 <main+16>:   swi   r22, r1,
0x194 <main+20>:   swi   r19, r1,
0x198 <main+24>:   swi   r0, r0, 3
0x19c <main+28>:   brlid r15, 1672
0x1a0 <main+32>:   addk  r23, r0,
0x1a4 <main+36>:   addik  r3, r0, 2
0x1a8 <main+40>:   swi   r3, r0, 2
0x1ac <main+44>:   addik  r6, r0, 2
0x1b0 <main+48>:   lwi   r4, r0, 2
0x1b4 <main+52>:   addk  r3, r0, r
0x1b8 <main+56>:   lbu   r5, r0, r
0x1bc <main+60>:   addik  r3, r3, 1
0x1c0 <main+64>:   addik  r4, r4, 1
0x1c4 <main+68>:   swi   r4, r0, 2
0x1c8 <main+72>:   xori  r18, r5,
0x1cc <main+76>:   beqid r18, 168
0x1d0 <main+80>:   cont
```



Xylon provides a tool for fast Flash memory programming. The SW application works in combination with a special FPGA design that implements the Flash interface.

The tool enables quick changes in HW/SW during the development.



• Xylon logiCRAFT2 Evaluation/Development platform, specially designed for Multimedia and Infotainment applications.

• The logiCRAFT2 can be used in combination with the EDK tools



Designed by XYLON

# Contact Info

## Germany office:

### Xylon Germany

Uhlandstr. 19  
49525 Lengerich/Westf. Germany  
Tel.: 49 5481 327 937  
Fax: 49 5481 327 938  
email: [info@logicbricks.de](mailto:info@logicbricks.de)

## Japan office:

### Sinby Corporation

3-7-6 Sin-chiba Chuo-ku  
Chiba-shi Chiba-ken  
260-0031 Japan  
Phone: 81-43-244-9714  
Fax: 81-43-244-9714  
email: [sales@sinby.com](mailto:sales@sinby.com)

## Headquarter:

**XYLON d.o.o.** Fallerovo Setaliste 22, 10000 Zagreb, Croatia - EUROPE  
Tel: 385-1-368-0026, Fax: 385-1-365-5167

email: [info@logicBRICKS.com](mailto:info@logicBRICKS.com)

<http://www.logicBRICKS.com>



Designed by XYLON