

Application Note



Akademie věd České republiky
Ústav teorie informace a automatizace AV ČR, v.v.i.

EdkDSP Accelerator IP Evaluation in Vivado 2014.4 Artix7 AC701 board

Jiří Kadlec

kadlec@utia.cas.cz

phone: +420 2 6605 22 16

UTIA AV CR, v.v.i.

Revision history:

Rev.	Date	Author	Description
1	30.11.2015	Jiří Kadlec	Description of Vivado 2014.4 Artix7 designs with EdkDSP accelerators and examples of use in the IoT applications, file system, www server and tftp server.

Acknowledgements:

This work has been partially supported by the Eniac JU project THINGS2DO “Thin but Great Silicon 2 Design Objects”, project number ENIAC JU 621221 and 7H14007 (Ministry of Education Youth and Sports of the Czech Republic [6]).

Table of contents

EdkDSP Accelerator IP Evaluation in Vivado 2014.4 Artix7 AC701 board 1

1. Summary 5

1.1 EdkDSP IP core evaluation package 5

1.2 What is included 6

2. Description of EdkDSP Accelerators in IoT Demonstrators 7

2.1 Description of EdkDSP accelerators and evaluation designs 7

2.2 Resources used by the designs 15

2.3 Use of external DDR3 memory 16

2.4 Re-programmability of EdkDSP accelerators 16

2.5 Debug of evaluation designs with the EdkDSP accelerator IPs 16

3. Installation and use of the evaluation package 18

3.1 Import of precompiled HW and SW projects into Xilinx SDK 2014.4 18

3.2 Evaluation of demo projects 22

3.3 Boot of the bitstream in Vivado 2014 Hardware Manager 23

3.4 Ethernet point to point connection with PC 26

3.5 Boot of the SW application 26

3.6 Use of the C compiler for the EdkDSP firmware with download from Ethernet 38

3.7 Use of the C compiler for the EdkDSP firmware without Ethernet 50

3.8 Use of ILA for debug of the EdkDSP accelerator IP 56

3.9 Display of Temperature, Debug and Verification 58

4. References 61

5. Evaluation version of Vivado 2014.4 Artix7 designs 62

6. Release version of designs for THINGS2DO project partners 63

7. Release version of Vivado 2014.4 Artix7 designs 65

Disclaimer 67

List of Figures

Figure 1: Evaluation package is combining MicroBlaze with 1 Gb Ethernet, WWW server, TFTP server with 5x (8xSIMD) EdkDSP floating point accelerator IP cores on Xilinx AC701board with Artix7	7
Figure 2: SoC design with 5 EdkDSP accelerator IPs on AC701 board	8
Figure 3: Hierarchical SoC design with EdkDSP IPs in the Vivado 2014.4 IP Integrator view	9
Figure 4: Xilinx MicroBlaze processor subsystem.....	10
Figure 5: Ethernet subsystem	10
Figure 6: I/O subsystem	11
Figure 7: Accelerator subsystem with 5 EdkDSP IPs; first accelerators with ILA debug probes.....	12
Figure 8: EdkDSP accelerator IP in Xilinx Vivado 2014.4 IP Integrator	13
Figure 9: Resources used by MicroBlaze and 5x (8xSIMD) EdkDSP, no ILA.....	15
Figure 10: Resources used by MicroBlaze and 5x (8xSIMD) EdkDSP, with ILA	15
Figure 11: Select the SDK Workspace	18
Figure 12: Include the UTIA EdkDSP Repository.....	19
Figure 13: Import existing projects into workspace	20
Figure 14: Select copy projects into workspace and finish the import of all projects	21
Figure 15: All projects are compiled. See IP blocks present in the design	22
Figure 16: Select and open the Hardware Manager tool from the Vivado 2014.4 initial menu	23
Figure 17: Select download.bit to program the Artix device and file defining the debug nets	24
Figure 18: Board HW is booted and debug nets are identified	24
Figure 19: Open PuTTY terminal.....	25
Figure 20: Select "Serial", select your COM port, set speed to 9600 and flow control to None	25
Figure 21: Select "bist_app.elf" code	26
Figure 22: Run bist_app.elf and select tests from the terminal keyboard (PC)	27
Figure 23: Run the edkdsp.elf application and select the EdkDSP Eval Op test	28
Figure 24: The EdkDSP basic vector floating point operations have been tested.....	29
Figure 25: Select "raw_axi_bce_fp12_eval_opl.elf application to test the lwIP services in RWW mode.....	30
Figure 26: The JavaScript has been loaded from the FPGA RAM based file system to your browser	31
Figure 27: The demo www server GUI	32
Figure 28: Test of basic operations of EdkDSP IP core	33
Figure 29: Start the socket_axi_bce_fp12_1x8_eval_op.elf demo application, working on top of the XilKernel OS	34
Figure 30: Test of vector operations is started from the www browser GUI. It is served by the lwIP library working on top of the XilKernel.....	35
Figure 31: Start the socket_axi_bce_fp12_1x8_fir_lms.elf application	36
Figure 32: The FIR and LMS computation is started from the web browser GUI.....	37
Figure 33: Evaluate the included C code for reprogramming of the EdkDSP accelerators	38
Figure 34: Start the VMware Player to run the C compiler for the EdkDSP accelerators as an Ubuntu binary user application	39
Figure 35: Mount the Windows 7 directory c:\VM_07as /mnt/cdrive in Ubuntu.....	40
Figure 36: Source the path to the EdkDSP C compiler tools.....	41
Figure 37: See the details of communication from the accelerator to MicroBlaze in the original code	42
Figure 38: Test has been performed and the tested EdkDSP accelerator created data file FP1101.TXT in the RAM file system located in the DDR3 of the AC701 board	43
Figure 39: Start TFTP client and get the file FP1101.TXT from the Artix7 FPGA to PC via Ethernet	44
Figure 40: Size of FP1101.TXT received from the Artix7 FPGA to PC via Ethernet. Confirm OK.....	44
Figure 41: Received FP1101.TXT file.....	45

Figure 42: Compile the C code with uncommented lines to display Input=00 instead of I=00.....	45
Figure 43: Select compiled binaries in Total Commander or in another file explorer	46
Figure 44: Drag and drop the firmware files to the TFTP client for the download to the Artix7 FPGA	47
Figure 45: Confirm Ano (yes in Czech...)	47
Figure 46: The TFTP server is indicating number of files and blocks transferred to Artix7 file system. Confirm OK.....	47
Figure 47: The TFTP server is indicating number of blocks uploaded to Artix7 file system	48
Figure 48: Confirm Ano (yes in Czech...)	49
Figure 49: FP1101.TXT received from the Artix7 FPGA to PC via Ethernet. Confirm OK.....	49
Figure 50: The console output indicates that 2 firmware files have been found.	49
Figure 51: Compile C source code for the accelerator by the EDKSPCC compiler	50
Figure 52: Select firmware header files and Ctrl-C Ctrl-V them to the edkdsp/src directory	51
Figure 53: Confirm to overwrite multiple files	52
Figure 54: See the updated edkdsp/src directory and section of the Microblaze source code, where the recompiled modified firmware is updated and EdkDSP accelerators are programmed.....	53
Figure 55: Recompile edkdsp project, download the .bit file and run the edkdsp.elf on Artix.....	54
Figure 56: Test the EdkDSP accelerator with the new firmware from the menu (type C).....	55
Figure 57: Vivado 2013.4 ILA display of the FIR filter computation	56
Figure 58: Vivado 2014.4 HW Manager ILA display of the FIR filter computation.....	57
Figure 59: Vivado 2014.4 HW Manager ILA display of adaptive LMS filter computation	58
Figure 60: Dashboard display: Temperature of the chip. No cooling at all.	59
Figure 61: Dashboard display: Temperature of the chip. Active cooling OFF-ON-OFF	59
Figure 62: PC screen snapshot. Demonstrates debug of EdkDSP IP core with ILA.....	60

1. Summary

1.1 EdkDSP IP core evaluation package

This application note describes precompiled Vivado 2014.4 Artix7 designs with the floating point EdkDSP accelerators and examples. The evaluation MicroBlaze SoC design with the AXI-lite bus is based on the Xilinx BIST (build in self-test) provided by Xilinx for the Artix7 AC701 board and the Vivado 2014.4 design flow. The network HW controller is supporting 1Gbit/100Mbit/10Mbit standards with HW DMA and a SW stack based on the lwIP TCP/IP stack library v1.4.1 with Xilinx adapter v2.2. The implementation follows guidelines described in the Xilinx application note XAPP1026 [3], [4]. The MicroBlaze processor is controlling 5 EdkDSP floating point accelerators. Each accelerator is organised as 8xSIMD reconfigurable computing data path, controlled by a PicoBlaze6 controller. This evaluation package is provided by UTIA for the Xilinx AC701 board with the 28nm Artix7 xc7a200t-2 device. This application note explains how to install and use the demonstrator on Windows7, (32 or 64 bit) and the Xilinx AC701 development board [1], [2].

These key features are demonstrated:

- WWW server running on Artix7 AC701 board with the lwip141 stack running in RAW mode on “bare metal” with standalone bsp or SOCKET mode with the Xilkernel bsp, supporting the POSIX compatible threads.
- TFTP server running on Artix7 AC701 board with the lwIP stack running in RAW mode or SOCKET mode.
- RAM based file system with file system present in the DDR3 memory on the AC701 board.
- 5 reprogrammable floating point accelerators for local embedded computing on the Artix7 28nm chip.
- Demo implementation of an adaptive acoustic noise cancellation on 1 of the 5 EdkDSP accelerators is computing the recursive adaptive LMS algorithm for identification of regression filter with 2000 coefficients in single precision floating point arithmetic with this sustained performance:
 - 754,0 MFLOP/s on a single 125 MHz (8xSIMD) EdkDSP accelerator (only 1 of 5 units is used).
 - 8,6 MFLOP/s on the 100 MHz MicroBlaze processor with the floating point HW unit.
- The EdkDSP accelerators can be reprogrammed by firmware. Programming of firmware is possible in C with the use of the UTIA EDKDSP C compiler. Each accelerator can store two firmware programs simultaneously. Each accelerator can swap firmware programs in only few clock cycles in the runtime.
- This evaluation package supports download of alternative firmware to EdkDSP accelerators from internet in parallel with the execution of the current firmware. This is demonstrated by the download of firmware by the TFTP server and by swap of the firmware (computing the acoustic room-response by FIR filter) to firmware performing the adaptive LMS identification of filter coefficients in the adaptive acoustic noise cancellation demo.
- The EdkDSP accelerator is providing single-precision floating point results bit-exact identical to the reference software implementations running on the MicroBlaze with the Xilinx HW single precision floating point unit.
- Single 125 MHz (8xSIMD) EdkDSP accelerator is 87x faster than computation on the performance optimized 100 MHz MicroBlaze with HW floating point unit, in the presented case of the 2000 tap adaptive LMS filter.
- The floating point 2000 tap coefficients FIR filter (acoustics room model) is computed by single 125 MHz (8xSIMD) EdkDSP accelerator with the floating point performance of 1111 MFLOP/s. Peak performance (only theoretical) of each 125 MHz (8xSIMD) EdkDSP accelerator is 2 GFLOP/s.
- Peak performance of the five instances of 125 MHz (8xSIMD) EdkDSP accelerators implemented in this demo design is 10 GFLOP/s (this is only theoretical peak figure).
- This evaluation package presents (8xSIMD) EdkDSP accelerator family with a single pipelined floating point divider data path. The IP cores differ by supported vector floating point operations, area used on the device and by power consumption.
- Precompiled evaluation designs also support debug of one EdkDSP IP core accelerator in the Vivado 2014.4 Hardware Manager in real-time. This is possible due to the support of the in-circuit logic analyser (ILA).

1.2 What is included

The evaluation package includes precompiled Vivado 2014.4 Artix7 designs with floating point EdkDSP accelerators and examples in form of Xilinx SDK 2014.4 SW projects for Windows 7 (32 or 64 bit):

- 8 evaluation versions of precompiled Artix7 designs. Each design contains one MicroBlaze and five instances of the EdkDSP accelerators. Each accelerator has 8xSIMD floating point data paths and programmable PicoBlaze6 controller for scheduling of floating point vector operations in the accelerator. The MicroBlaze works with 100 MHz system clock and EdkDSP accelerators use 125 MHz clock. The MicroBlaze processor works with 1Gb Ethernet with DMA controller and 1GB DDR3 memory. Designs are compiled in Xilinx Vivado 2014.4.
- UTIA is providing source code for the demo applications and SW projects for the Xilinx SDK 2014.4. These source code projects are compiled with the UTIA library libwal.a serving for the EdkDSP communication and the library libmfsimage.a with the initial file system supporting the simple www server GUI example.
- The included evaluation designs with UTIA EdkDSP accelerators have HW limitation of maximal number of performed accelerated vector operations.
- The UTIA EDKDSP C compiler is provided in form of 4 binary applications running in the Ubuntu OS installed in the VMware Workstation 12 Player.
- The firmware for accelerators is provided in source code and also in format of binary file headers to enable initial evaluation of the EdkDSP accelerator IP cores without the need to install the EDKDSPCC C compiler.
- UTIA partners of the Eniac THINGS2DO [6] project, can get from UTIA the release version of Vivado 2014.4 HW design projects with the evaluation versions of the EdkDSP IP core accelerators (in the Vivado 2014.4 IP netlist format) for free. See chapter 6 for specification of deliverables for the Eniac THINGS2DO [6] project partners with the license details.
- Release versions of Vivado 2014.4 HW design projects and release version of EdkDSP IP core accelerators for the Xilinx AC701 board is offered by UTIA. All customers can order and buy from UTIA the release version of this demo. It includes the Vivado 2014.4 HW design projects with the EdkDSP accelerators (in the Vivado 2014.4 IP netlist format) with the HW limitation of maximal number of performed vector operations removed. See sections 7 of this application note for specification of deliverables and license details.

2. Description of EdkDSP Accelerators in IoT Demonstrators

2.1 Description of EdkDSP accelerators and evaluation designs

This application note describes how to set-up and use of 8 HW designs running on one MicroBlaze processor with five (8xSIMD) EdkDSP accelerators on Xilinx AC701 board. See Figure 1 and Figure 2 and Figure 62.

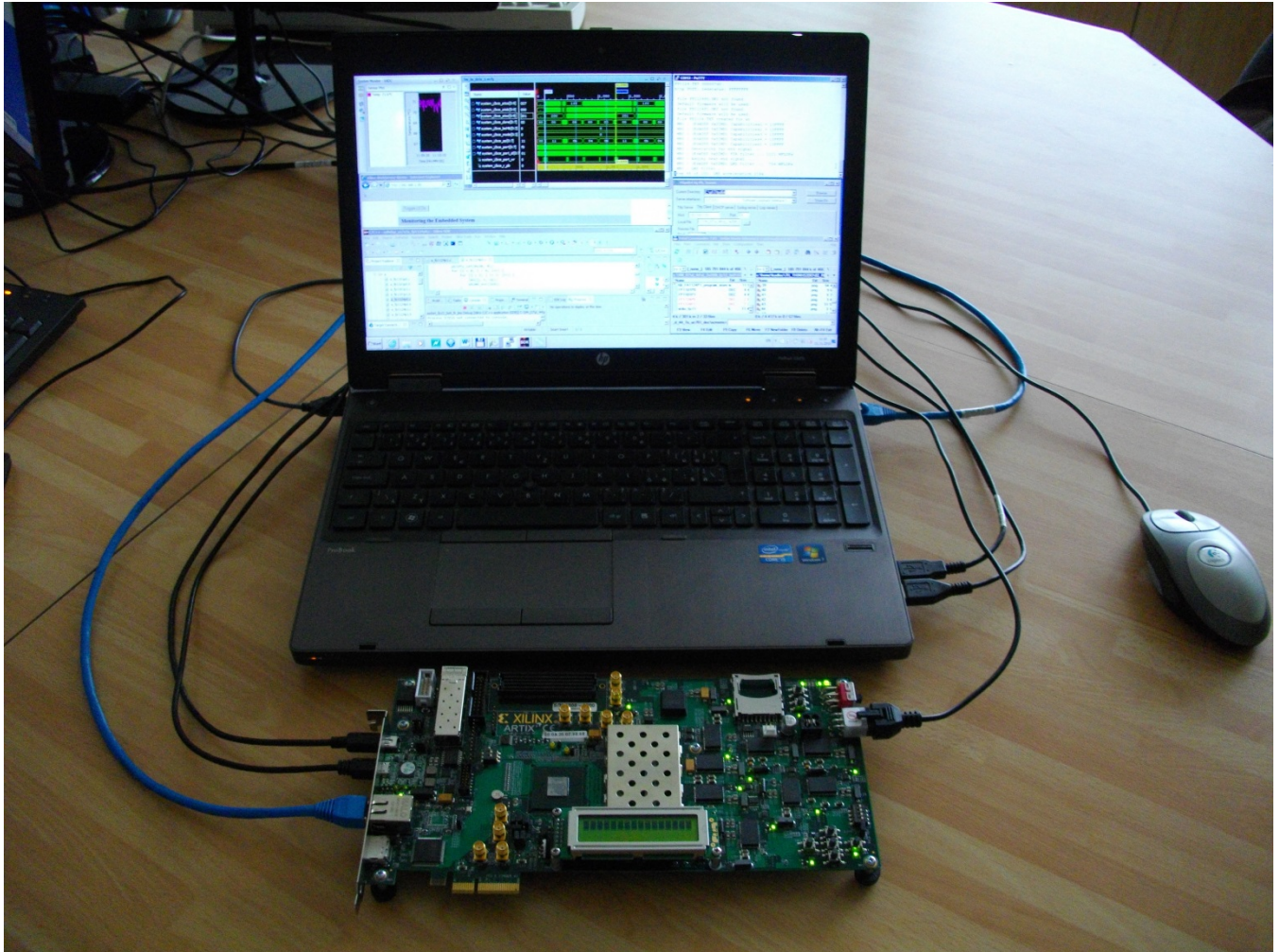


Figure 1: Evaluation package is combining MicroBlaze with 1 Gb Ethernet, WWW server, TFTP server with 5x (8xSIMD) EdkDSP floating point accelerator IP cores on Xilinx AC701 board with Artix7

HW designs precompiled in Vivado 2014.4 combine MicroBlaze and five 8xSIMD EdkDSP accelerators. All designs demonstrate use of 8xSIMD EdkDSP floating point accelerators on 32 bit AXI-lite bus of the Xilinx MicroBlaze processor on the Xilinx Artix7 AC701 board. See Figure 2.

Common properties of precompiled Vivado 2014.4 evaluation designs:

- The EdkDSP floating point accelerators are reconfigurable during runtime by change of firmware.
- All HW evaluation designs have been compiled in Xilinx VIVADO 2014.4 with SW projects for SDK 2014.4.
- 4 designs are precompiled without ILA debug support and 4 designs include ILA debug support

The demonstrator package includes source code of set of SW demos prepared for easy import of projects and compilation in the Xilinx SDK 2014.4.

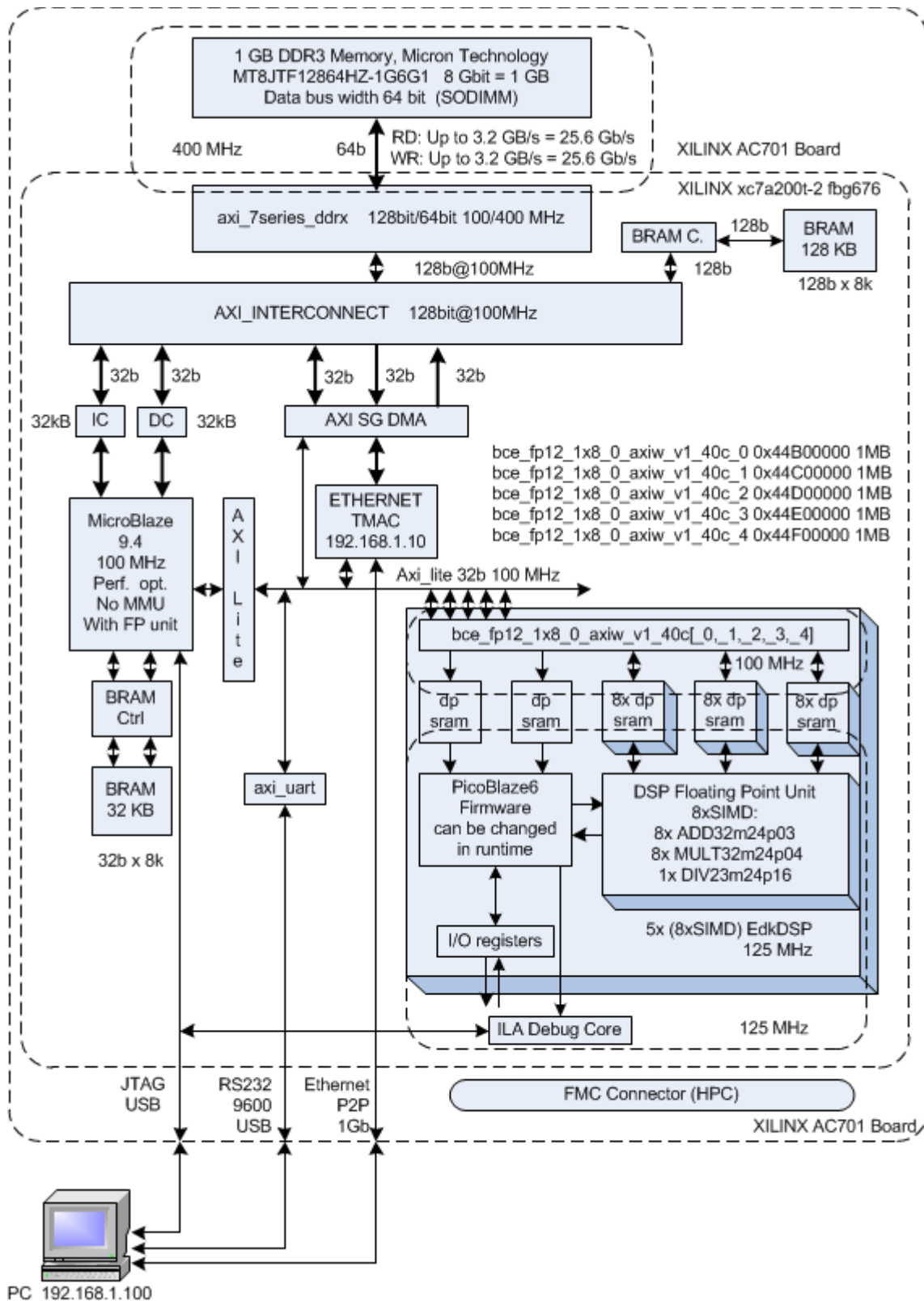


Figure 2: SoC design with 5 EdkDSP accelerator IPs on AC701 board

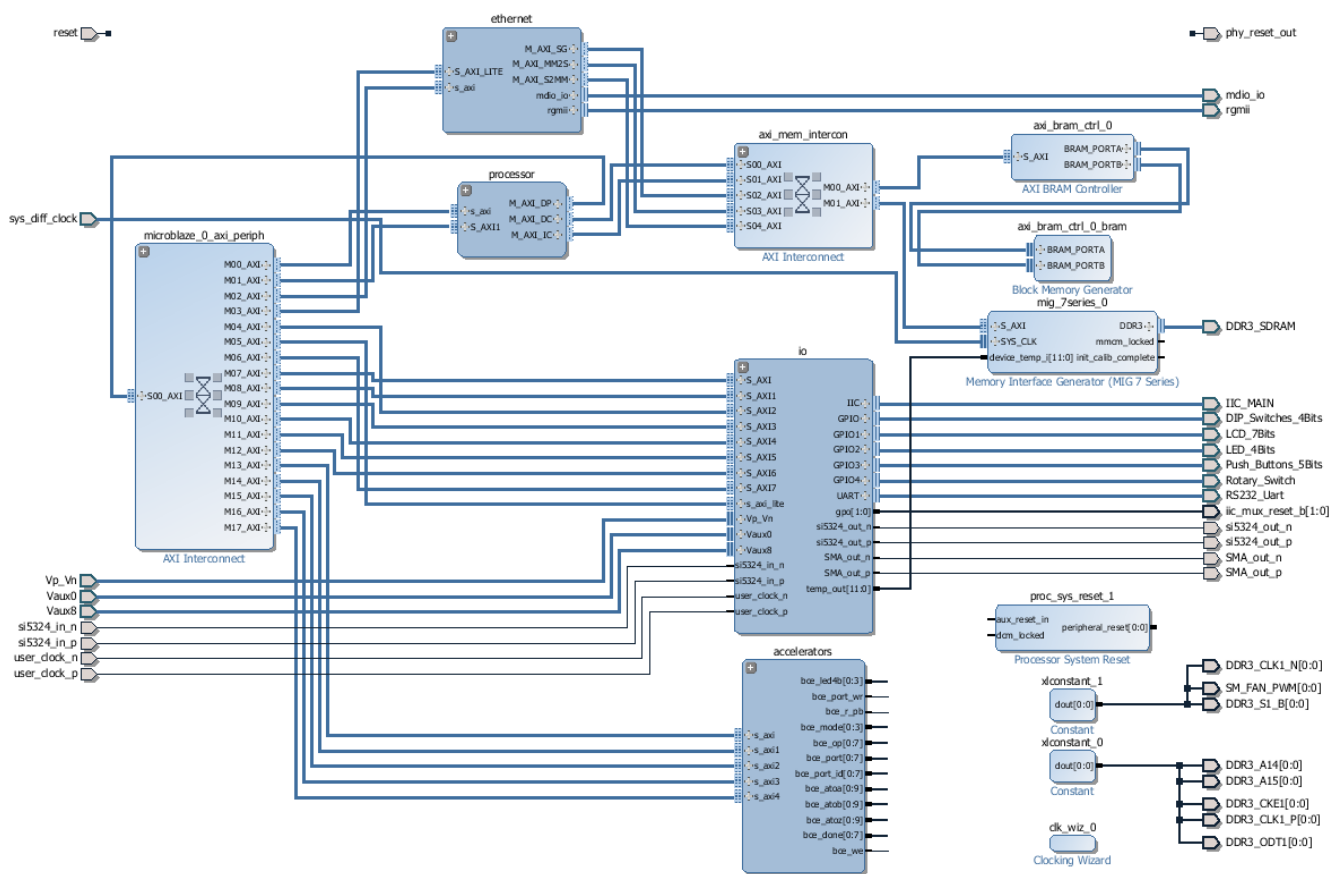


Figure 3: Hierarchical SoC design with EdkDSP IPs in the Vivado 2014.4 IP Integrator view

- All EdkDSP accelerators are memory mapped on the 32 bit AXI-lite bus. Each (8xSIMD) EdkDSP accelerator has reserved address space 1 MByte. See Figure 2.
- Figure 3 describes the SoC with MicroBlaze in hierarchical view.
- Processor block includes MicroBlaze processor with BRAM block for the initial boot. See Figure 4.
- The 1Gb Ethernet subsystem is presented in Figure 5.
- Hierarchical block "io" contains standard I/O peripheral interfaces, timer, uart, analog digital converter, temperature sensor. See Figure 6.
- Hierarchical block "accelerators" contains five (8xSIMD) EdkDSP accelerators. See Figure 7.
- Single (8xSIMD) EdkDSP accelerator IP core is presented in Figure 8 with better resolution with the input, output and debug probe signals.

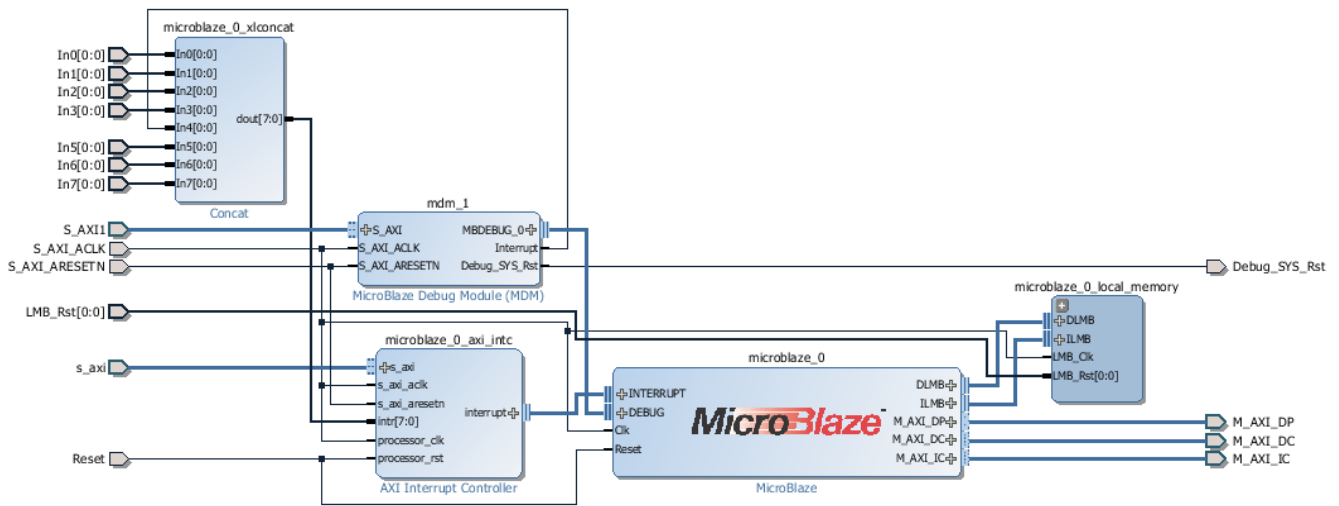


Figure 4: Xilinx MicroBlaze processor subsystem.

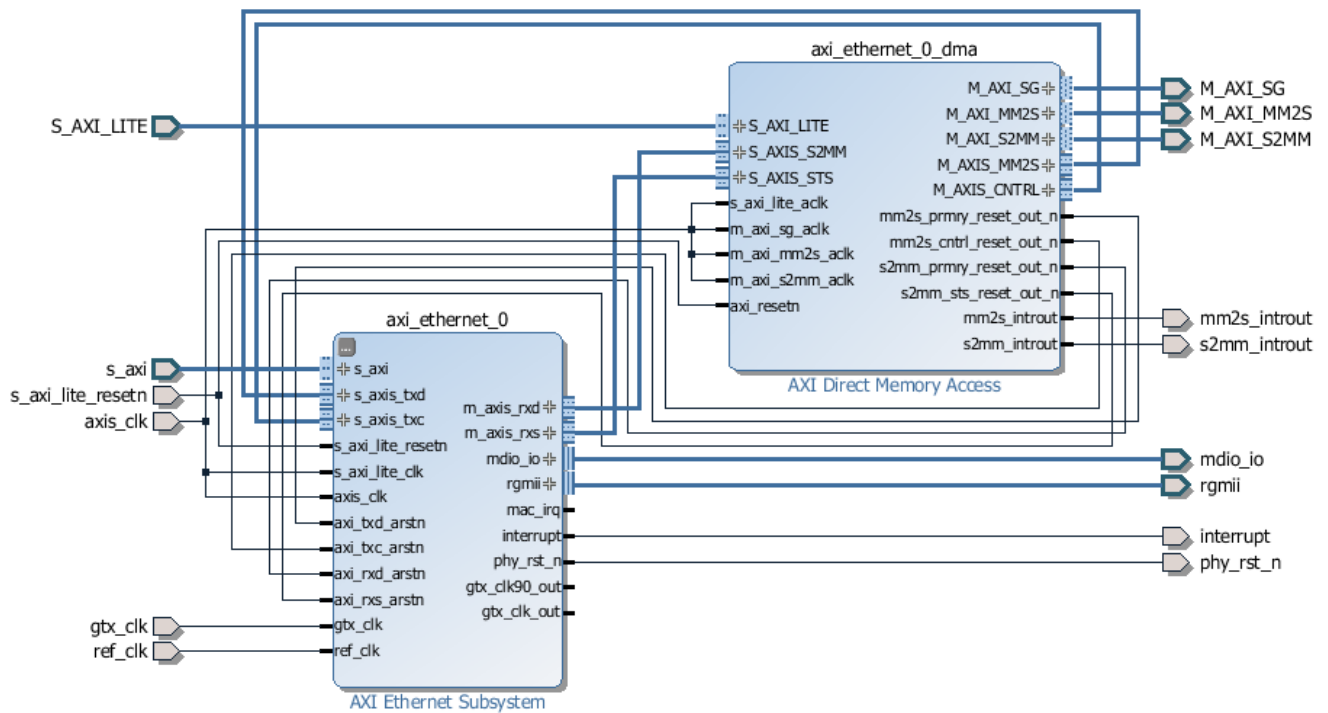


Figure 5: Ethernet subsystem

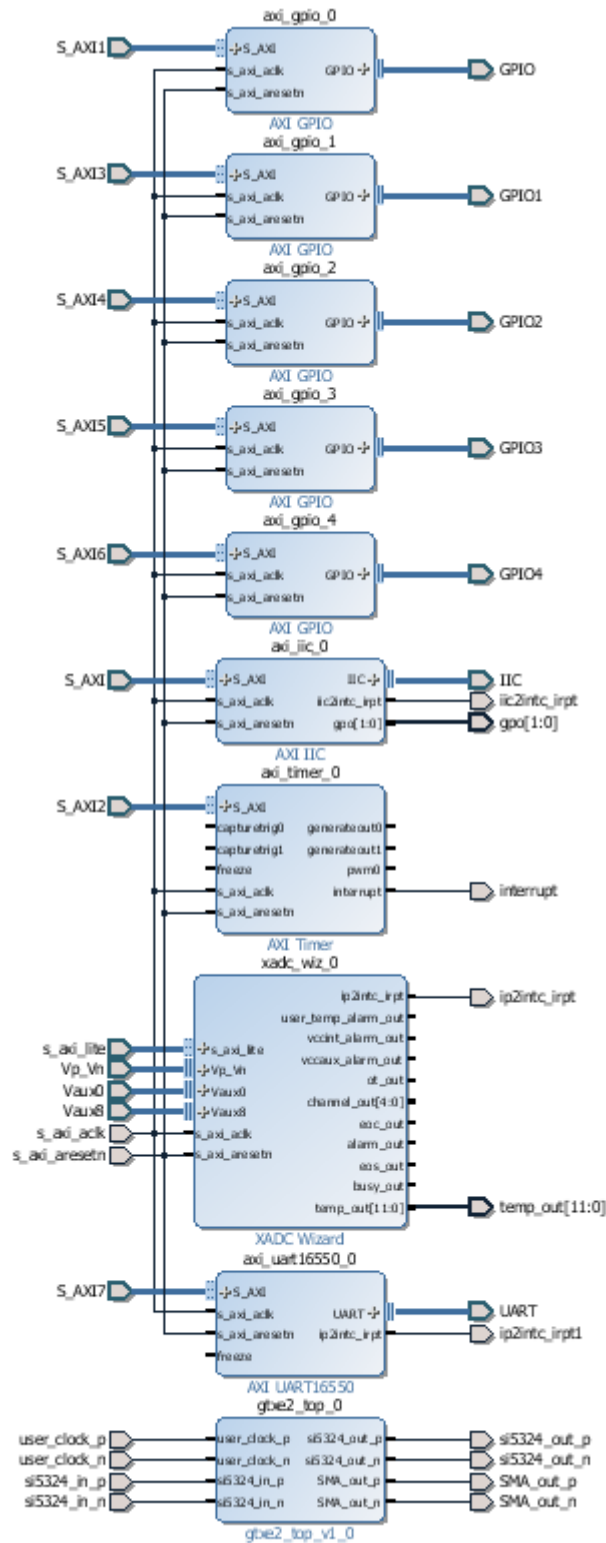


Figure 6: I/O subsystem

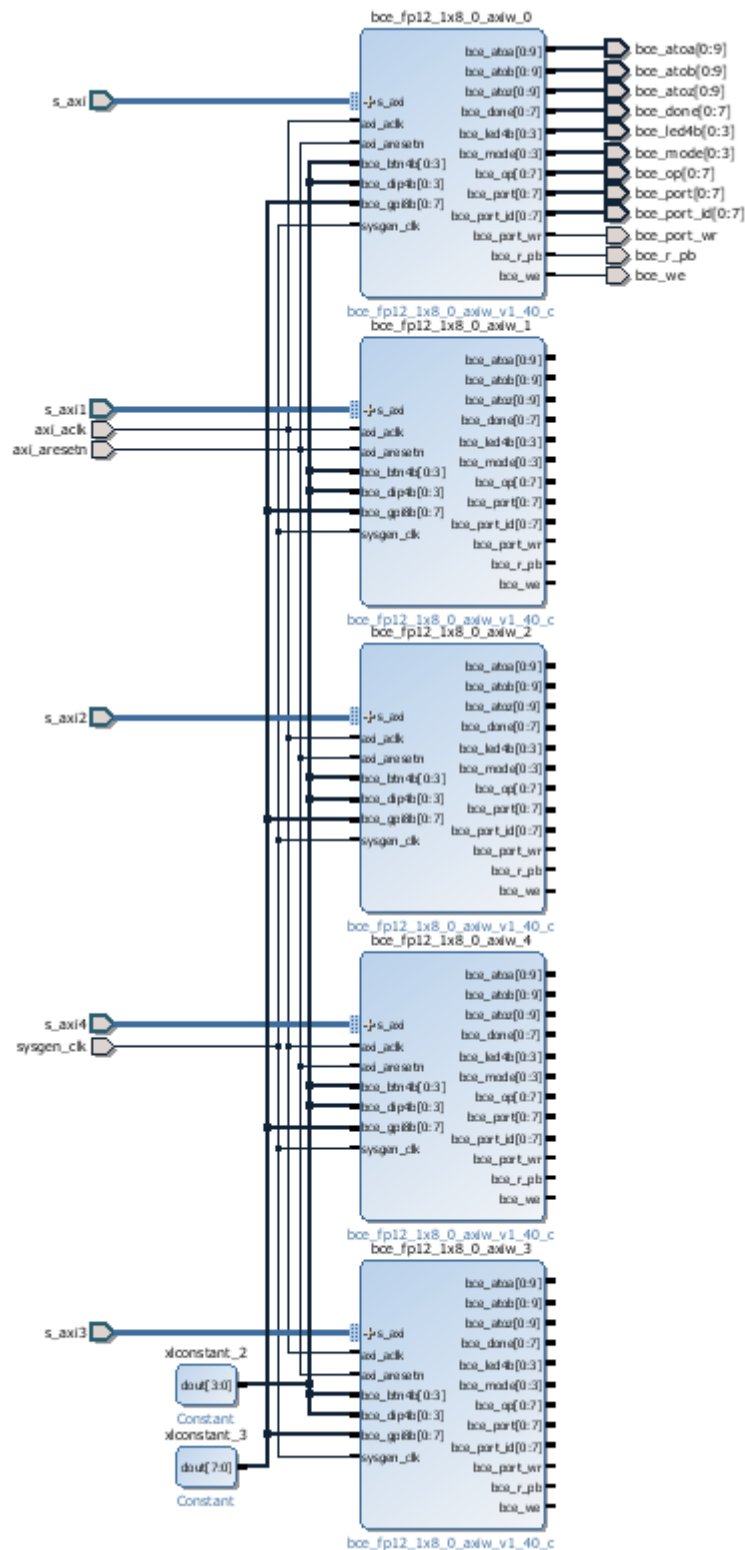


Figure 7: Accelerator subsystem with 5 EdkDSP IPs; first accelerators with ILA debug probes

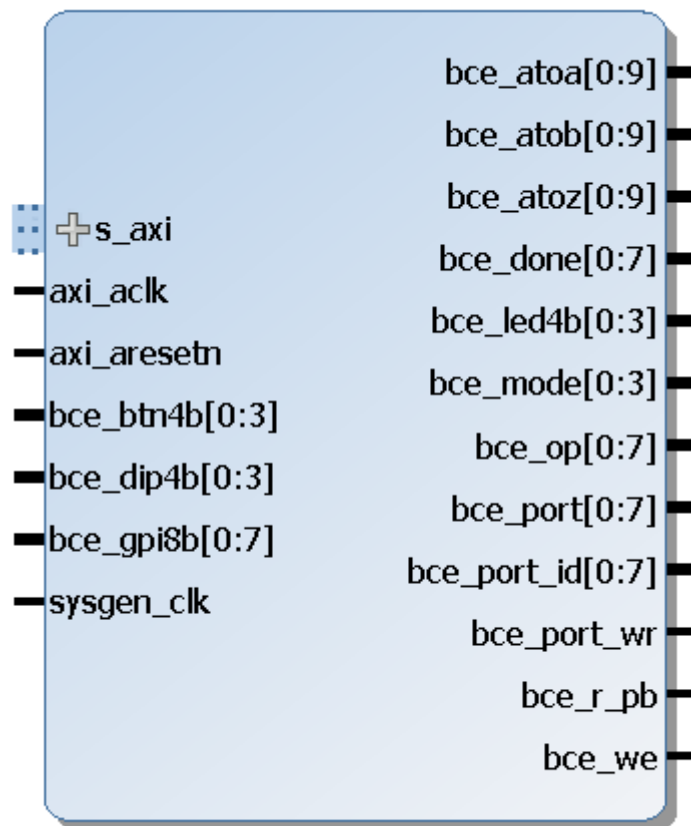


Figure 8: EdkDSP accelerator IP in Xilinx Vivado 2014.4 IP Integrator

The SoC system described in Figure 2 and Figure 3 serves for evaluation of four different EdkDSP floating point accelerator IP cores **bce_fp12_1x8_0_axiw_v1_[10|20|30|40]**. Four grades [10|20|30|40] of the EdkDSP accelerator IP differ in HW-supported vector computing capabilities:

Accelerator **bce_fp12_1x8_0_axiw_v1_10** is area optimized and supports only data transfers and vector floating point operations FPADD, FPSUB in 8 SIMD data paths.

Accelerator **bce_fp12_1x8_0_axiw_v1_20** performs identical operations as **bce_fp12_1x8_0_axiw_v1_10** plus the vector floating point MAC operations in 8 SIMD data paths. MAC is supported for length of vectors 1 up to 10. This accelerator is optimized for applications like floating point matrix multiplication with one row and column dimensions ≤ 10 .

Accelerator **bce_fp12_1x8_0_axiw_v1_30** supports identical operations as **bce_fp12_1x8_0_axiw_v1_20** plus HW accelerated computation the floating point vector by vector dot products performed in 8 SIMD data paths. It is optimized for parallel computation of up to 8 FIR or LMS filters, each with size up to 250 coefficients. It is also effective in case of floating point matrix by matrix multiplications, where one of the dimensions is large (in the range from 11 to 250).

Accelerators **bce_fp12_1x8_0_axiw_v1_40** support identical operations as **bce_fp12_1x8_0_axiw_v1_30** plus an additional HW support of dot product. It is computed in 8 data paths with HW-supported wind-up into single scalar result propagated into all SIMD planes.

All **bce_fp12_1x8_0_axiw_v1_[10|20|30|40]** accelerators support single data path for, pipelined, floating-point division (FPDIV) with vector operands taken from the first SIMD plain and the result vector propagated into all 8

All **bce_fp12_1x8_0_axiw_v1_[10|20|30|40]** accelerators are suitable for applications like adaptive normalised NLMS filters and the square root free versions of adaptive RLS QR filters and adaptive RLS LATTICE filters.

The debug probes (see Figure 2, Figure 3, Figure 7 and Figure 8) can provide visibility auto-generated addresses and the schedule of vector operation in the first (8xSIMD) EdkDSP accelerator IP core.

Concrete processed floating point data are not displayed. Floating point data can be better analysed directly by the MicroBlaze debugger. MicroBlaze program can access content of dual-ported memories of the (8xSIMD) EdkDSP accelerator, copy them to corresponding C variables located in the DDR3 memory and display/analyze them in the MicroBlaze debugger.

This is description of ILA debug probe ports of the (8xSIMD) EdkDSP accelerator (the instantiated ILA stores 32k samples for all probes with the 125 MHz clock):

- **bce_atoa[0:9]** Memory A address (addressing 1024 32 bit floating point values)
- **bce_atob[0:9]** Memory B address (addressing 1024 32 bit floating point values)
- **bce_atoz[0:9]** Memory Z address (addressing 1024 32 bit floating point values)
- **bce_done[0:7]** Vector operation in progress or finished
- **bce_led4b[0:3]** 4bit output, intended for led signalling. Unconnected in the design.
- **bce_mode[0:3]** Mode of communication protocol PicoBlaze6 - MicroBlaze
- **bce_op[0:7]** Vector operation to be performed.
- **bce_port[0:7]** Data on external port.
- **bce_port_id[0:7]** External port address. Address space [0x0 ... 0x1F] are reserved for internal construction of the WLIW instruction to the 8xSIMD vector processing unit of the EdkDSP. Address space [0x20 ... 0xFF] can be used by the user.
- **bce_port_wr** Write strobe related to writing of 8bit data to the external port address
- **bce_r_pb** Reset of the PicoBlaze6
- **bce_we** Write strobe related to writing of a WLIW instruction to the 8xSIMD vector processing unit of the EdkDSP.

These probes are used for the real-time analysis of the computation inside of the 8xSIMD vector processing unit of the EdkDSP. This helps with the debug of the coordination of the PicoBlaze6 firmware code, the vector processing unit together with MicroBlaze code.

2.2 Resources used by the designs

Resources used by the 4 presented designs without ILA are summarised in Figure 8 and by 4 designs with ILA in Figure 9

7a200t-2 No ILA mem	fp	fp	fp	fp	fp	Design size			Performance	
	Add Mul	Mac	Dot Prod	S8 Prod	Div	FFs %	LUTs %	Bram No (of)	LMS Mflop/s	FIR Mflop/s
ac701_bist						10	21	74 (365)		
(5x) fp12_1x8_10	8x				1x	21,1	52,3	254 (365)		
(5x) fp12_1x8_20	8x	8x			1x	23,0	52,8	254 (365)		
(5x) fp12_1x8_30	8x	8x	8x		1x	25,0	55,8	254 (365)		
(5x) fp12_1x8_40	8x	8x	8x	1x	1x	25,1	58,1	254 (365)	(5x) 754	(5x) 1111

Figure 9: Resources used by MicroBlaze and 5x (8xSIMD) EdkDSP, no ILA

7a200t-2 With ILA mem	fp	fp	fp	fp	fp	Design size			Performance	
	Add Mul	Mac	Dot Prod	S8 Prod	Div	FFs %	LUTs %	Bram No (of)	LMS Mflop/s	FIR Mflop/s
ac701_bist						10	21	74 (365)		
(5x) fp12_1x8_10	8x				1x	23,3	56,6	310 (365)		
(5x) fp12_1x8_20	8x	8x			1x	25,3	57,0	310 (365)		
(5x) fp12_1x8_30	8x	8x	8x		1x	27,2	69,7	310 (365)		
(5x) fp12_1x8_40	8x	8x	8x	1x	1x	27,3	62,2	310 (365)	(5x) 754	(5x) 1111

Figure 10: Resources used by MicroBlaze and 5x (8xSIMD) EdkDSP, with ILA

The ac701_bist design describes resources used by the MicroBlaze SoC without EdkDSP accelerators and without ILA memories. The internal block RAM memory is set to 32KB and 128KB. All evaluation designs with EdkDSP accelerators work with:

- **40 single precision 3-stage pipelined floating point add/sub** units each performing up to 125 MFLOP/s.
 - **40 single precision 4-stage pipelined floating point multiply** units each performing up to 125 MFLOP/s.
 - **5 PicoBlaze6 8-bit controllers** with 125 MHz system clock, each executing 62.5 Mil. instructions/s.
 - **1 MicroBlaze (ver 9.4) 32-bit processor** (100 MHz) working with one single precision pipelined floating point add/sub unit and one single precision pipelined floating point multiply unit, 32 KB data cache and 32 KB instruction cache.
 - **5 single precision 16-stage pipelined floating point divide** units each performing up to 125 MFLOP/s.
- The designs can use accelerators `bce_fp12_1x8_0_axiw_v1_[10|20|30|40]` with different HW supported operations. This is reflected in the difference of resources used by the designs. See Figure 9 and Figure 10.

2.3 Use of external DDR3 memory

Presented evaluation designs are running on the Xilinx AC701 development board [1], [2]. See Figure 1. It is using the 1GB DDR3 memory with clock signal 400 MHz. The external DDR3 memory is connected to Xilinx Artix7 xc7a200t-2 chip by 64 bit wide data path.

2.4 Re-programmability of EdkDSP accelerators

Each (8xSIMD) EdkDSP floating point accelerator subsystem contains one reprogrammable Xilinx PicoBlaze6 8-bit controller and the floating point (8xSIMD) DSP data paths. The performance of the accelerator is application specific. The Xilinx PicoBlaze6 processor has fixed configuration with maximal size of the program memory 4096 (18 bit wide) words and 64 Bytes scratch pad RAM memory.

Each (8xSIMD) EdkDSP accelerator IP works with 2 separate program memory blocks. Both program memories are accessible by the MicroBlaze processor via the AXI-lite bus. The MicroBlaze application can write new firmware to the currently unused program memory, while the PicoBlaze6 controller is executing firmware from the second program memory. All 5 EdkDSP accelerator IPs can run on parallel and each accelerator can use its own different firmware code.

2.5 Debug of evaluation designs with the EdkDSP accelerator IPs

MicroBlaze program can communicate individually with each of the 5 EdkDSP accelerators via the AXI-lite bus. The communication is using UTIA Worker Abstraction Layer (WAL) library API. This API is also used for support of exchange of debug information and for the access from the accelerator to the MicroBlaze terminal.

The PicoBlaze6 controller [5] can exchange data and text via the 8 bit communication data path with the MicroBlaze processor. This path is used to communicate parameters to the accelerators and to get messages or reports from accelerators for debugging. Text file with information from the accelerator can be stored in the RAM based file system of the MicroBlaze processor. This ASCII2 text file can be downloaded to PC by TFTP client via the p2p 1G Ethernet for inspection. It is possible due to the TFTP server running in background on the MicroBlaze processor.

Floating point data are accessed by the MicroBlaze processor via the dual ported block memories of EdkDSP accelerator IPs. The MicroBlaze side of the dual-port memories is mapped into the MicroBlaze memory. The MicroBlaze processor can copy data from the dual ported memories to the DDR3 global workspace and display floating point data in the debugger.

The computation performed in the (8xSIMD) EdkDSP accelerator IPs usually overlaps with MicroBlaze communication with the DDR3. It is supported by data and program cache. The Ping-Pong swap of memory banks is usually used by the accelerator firmware. The (8xSIMD) EdkDSP accelerator firmware is computing (in parallel) in some banks of its dual ported memories. MicroBlaze is communicating (sequentially) to/from DDR3 in another set of banks of these dual-port memories. This process can be stopped in synchronisation points, inspected and debugged by the MicroBlaze debugger running in the SDK 2014.4.

This standard C debugging approach is combined with the possibility to display probes for the first EdkDSP accelerator by the in-circuit-logic-analyser ILA. The ILA analyser is capable to display 32K of 125 MHz samples with user defined trigger conditions. ILA is part of the HW Manager utility present in the Xilinx Vivado 2014.4 tool.

Evaluation package provides support for fast iterations of a debug cycle for the EdkDSP accelerators IP firmware:

1. HW design and MicroBlaze code is running and does not need to be re-booted. It supports RAM based filesystem, TFTP server and WWW server.
2. EdkDSP accelerator IP firmware can be modified and recompiled from C source code by the EDSKDSP CC compiler running on the PC.
3. Compiled EdkDSP accelerator IP firmware can be downloaded via TFTP to the DDR3 filesystem of MicroBlaze.
4. The ILA trigger conditions can be modified and the ILA core can be armed in the HW Manager utility.
5. EdkDSP application is started (usually from the WWW GUI). MicroBlaze updates the new EdkDSP accelerator firmware and starts the computation.
6. The ILA is triggered and samples 32K samples of probe signals. This can be analysed together with the console output and possibly with text files uploaded to PC via TFTP from the MicroBlaze file system.
7. If results are OK, we are done. If not, we can go back to step 1.

This debug iteration loop 1-7 can be very fast. It includes only single compilation step from C source code to the EdkDSP firmware code. ILA probes are usually combined with the use of MicroBlaze debugger in the SDK 2014.4.

The MicroBlaze C source code is usually used for computation of “golden” reference data.

ILA display helps to visualise the internal synchronous data which drive the computation in the first EdkDSP accelerator with the single clock cycle (128 MHz resp. 8ns) resolution. This visibility often helps to the developer to gradually optimize the initial (already working, but not optimised) HW implementation of the firmware running in the (8x SIMD) EdkDSP accelerator IP core.

3. Installation and use of the evaluation package

3.1 Import of precompiled HW and SW projects into Xilinx SDK 2014.4

Unzip the evaluation package to directory of your choice. The directory `c:\VM_07` will be used in this application note. You will get these directories:

`C:\VM_07\d_44\d_7a200t_fp12_5x8`

`C:\VM_07\d_44\d_7a200t_fp12_5x8_IMPORT`

Select SDK 2014.4 workspace in `C:\VM_07\d_44\d_7a200t_fp12_5x8\SDK_Workspace`. See Figure 11.

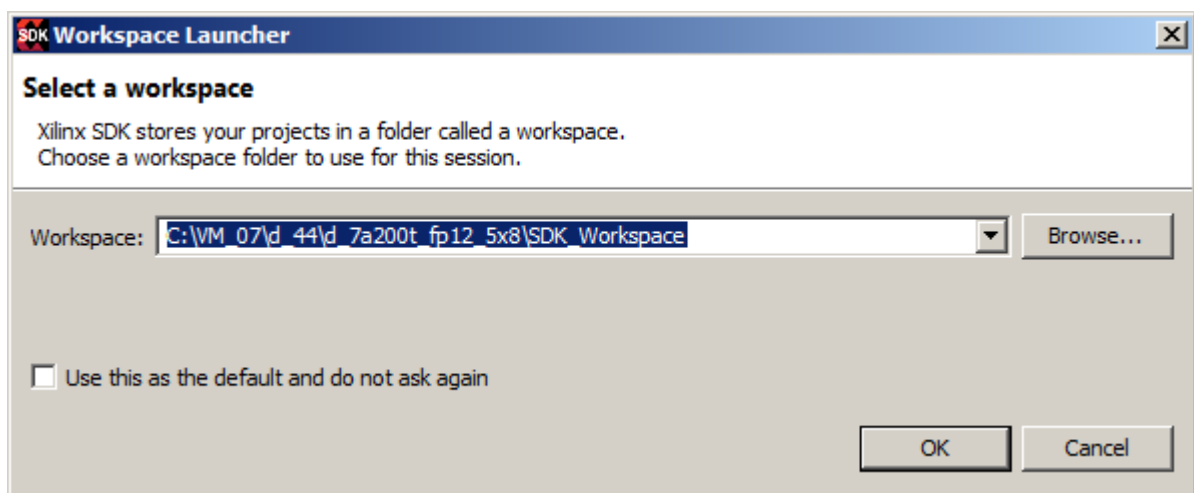


Figure 11: Select the SDK Workspace

Add `C:\VM_07\d_44\d_7a200t_fp12_5x8\repo_edkdsp` path to the `lwip141_v2_1` repository. See Figure 12.

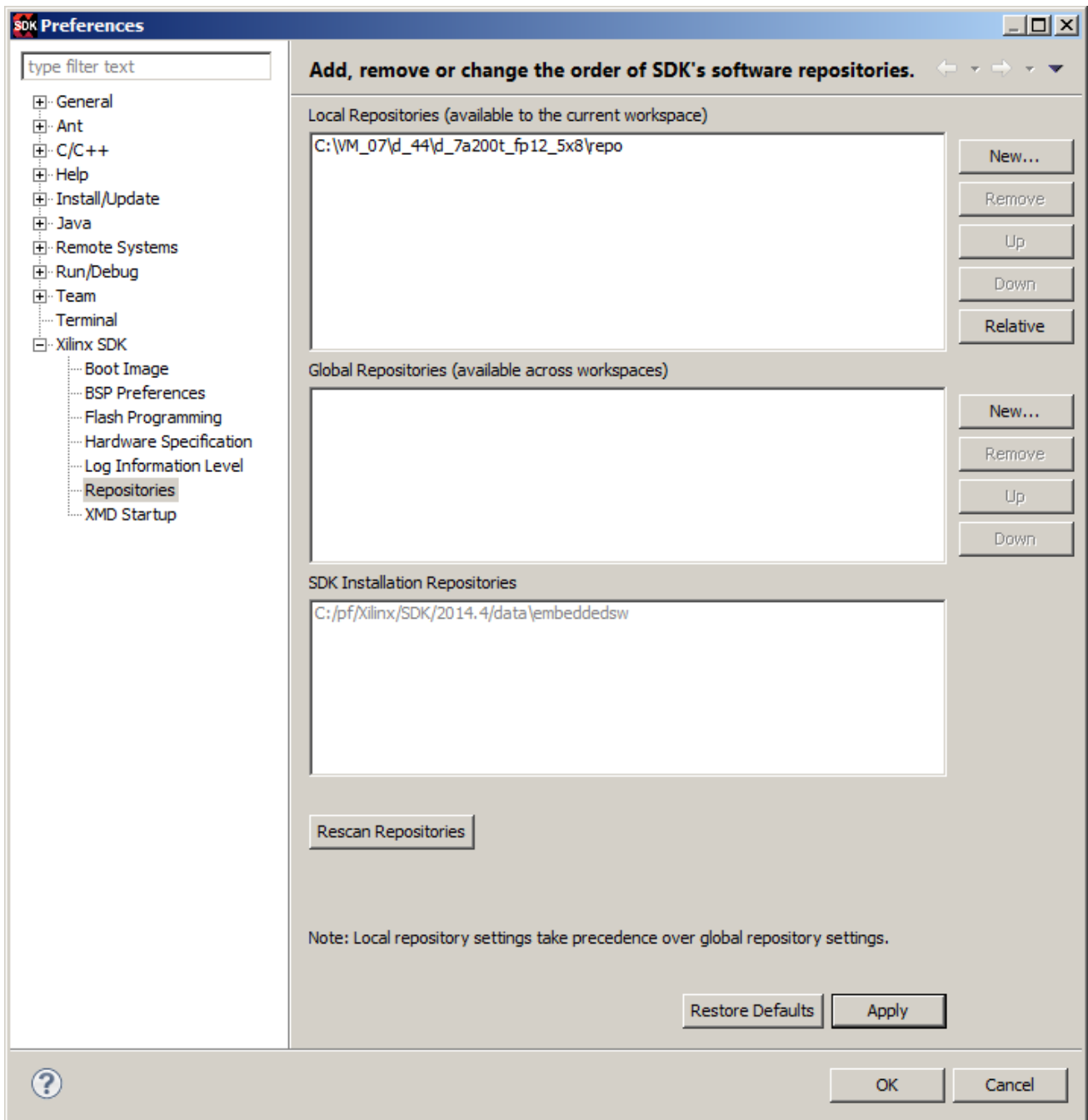


Figure 12: Include the UTIA EdkDSP Repository

Click on the “Rescan Repositories” button. Click on the “Apply button”, and finally click on the OK button. The path to the SW drivers has been defined.

Predefined HW and SW projects can be imported into SDK now. Select:

File -> Import -> General -> Existing Projects into Workspace
Click on Next button. See Figure 13.

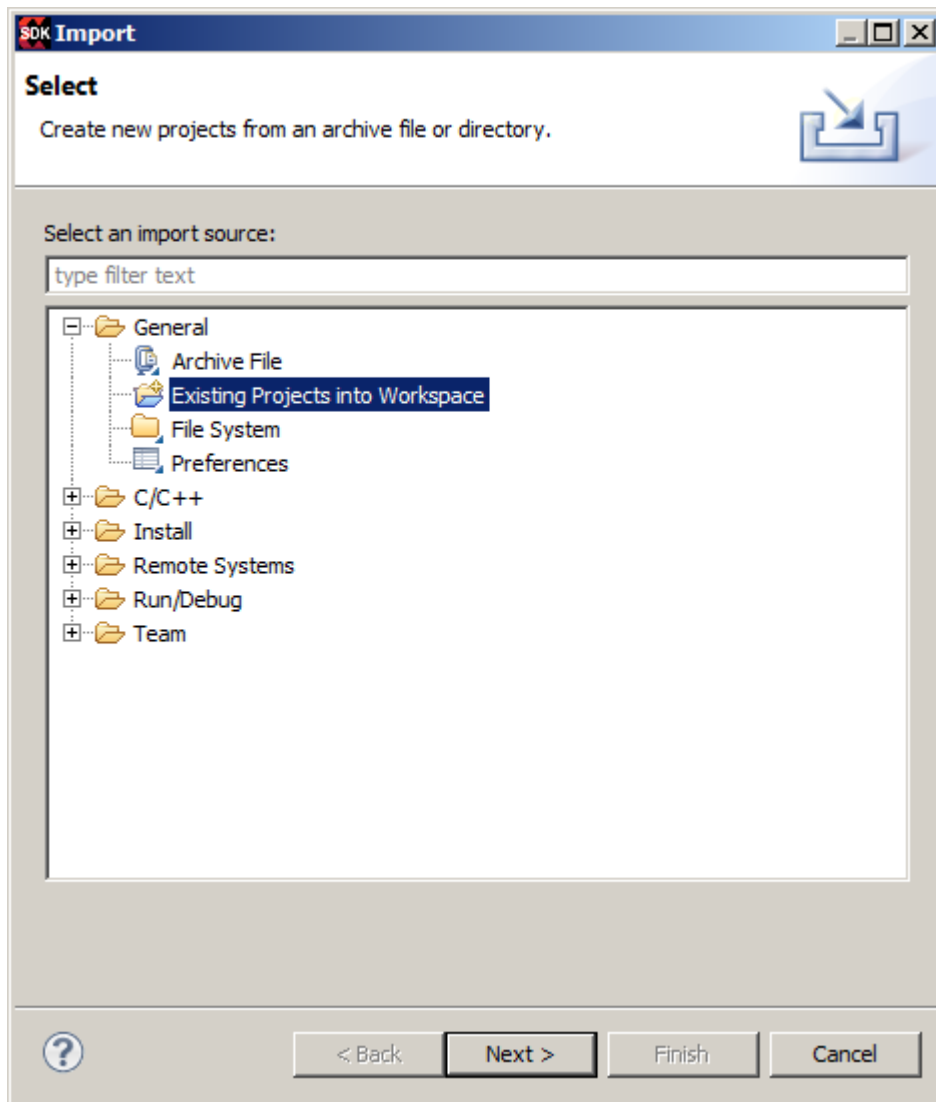


Figure 13: Import existing projects into workspace

Select the directory with projects to be imported. See Figure 14.

C:\VM_07\d_34_7a\d_7a200t_fp12_5x8_IMPORT

Set the “Copy projects into workspace” check box.

Click on Finish button. See Figure 14.

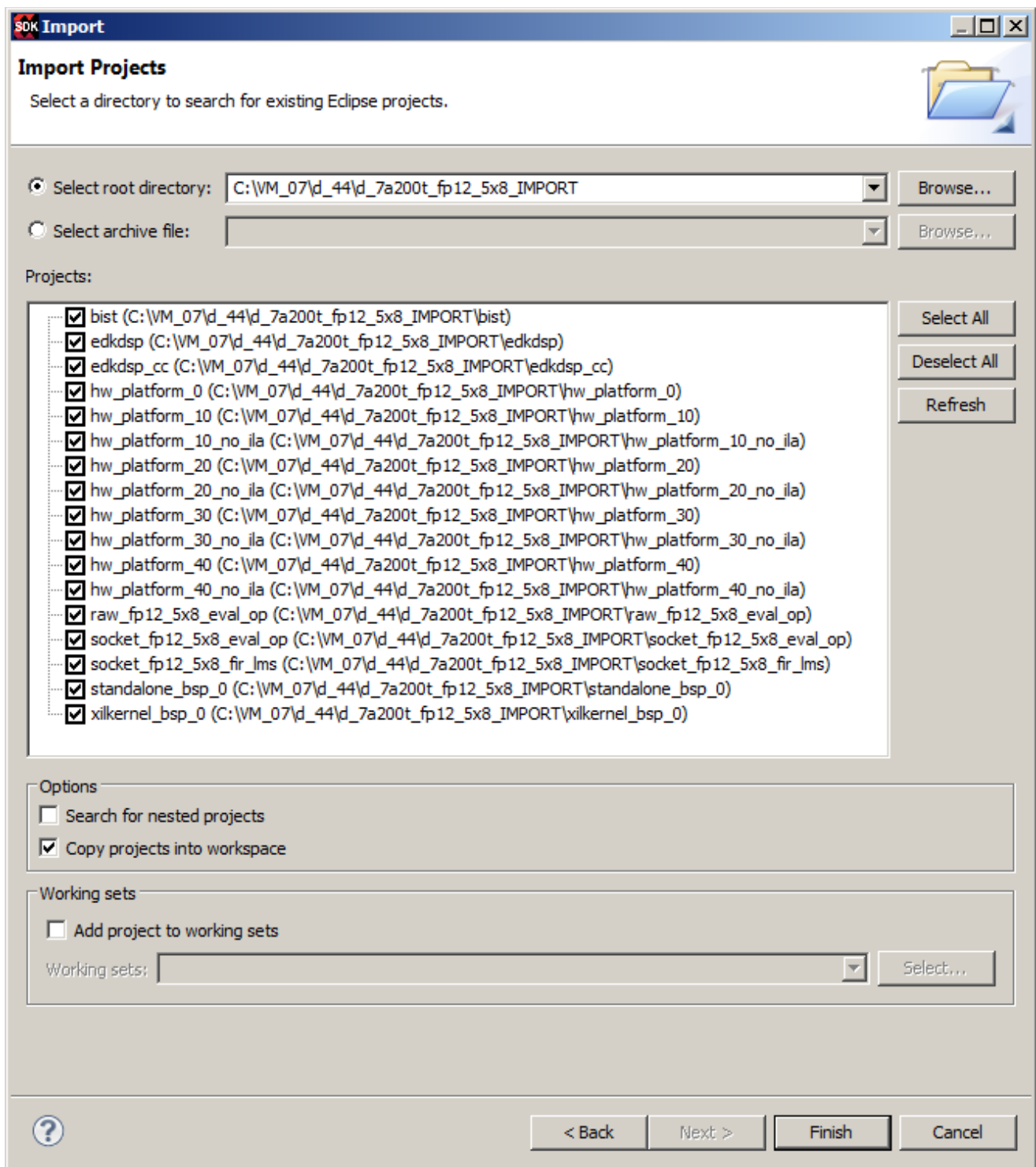


Figure 14: Select copy projects into workspace and finish the import of all projects

Process of compilation will start automatically. This first compilation of all SDK SW projects can take several minutes to finish. It should finish without errors. See Figure 15.

3.2 Evaluation of demo projects

The “bist_app” project in the “Project Explorer” window of the SDK 2014.4 is only slightly modified version of the Xilinx BIST SW application project. The RAM memory test is adjusted for the 128 KB RAM. See Figure 15.

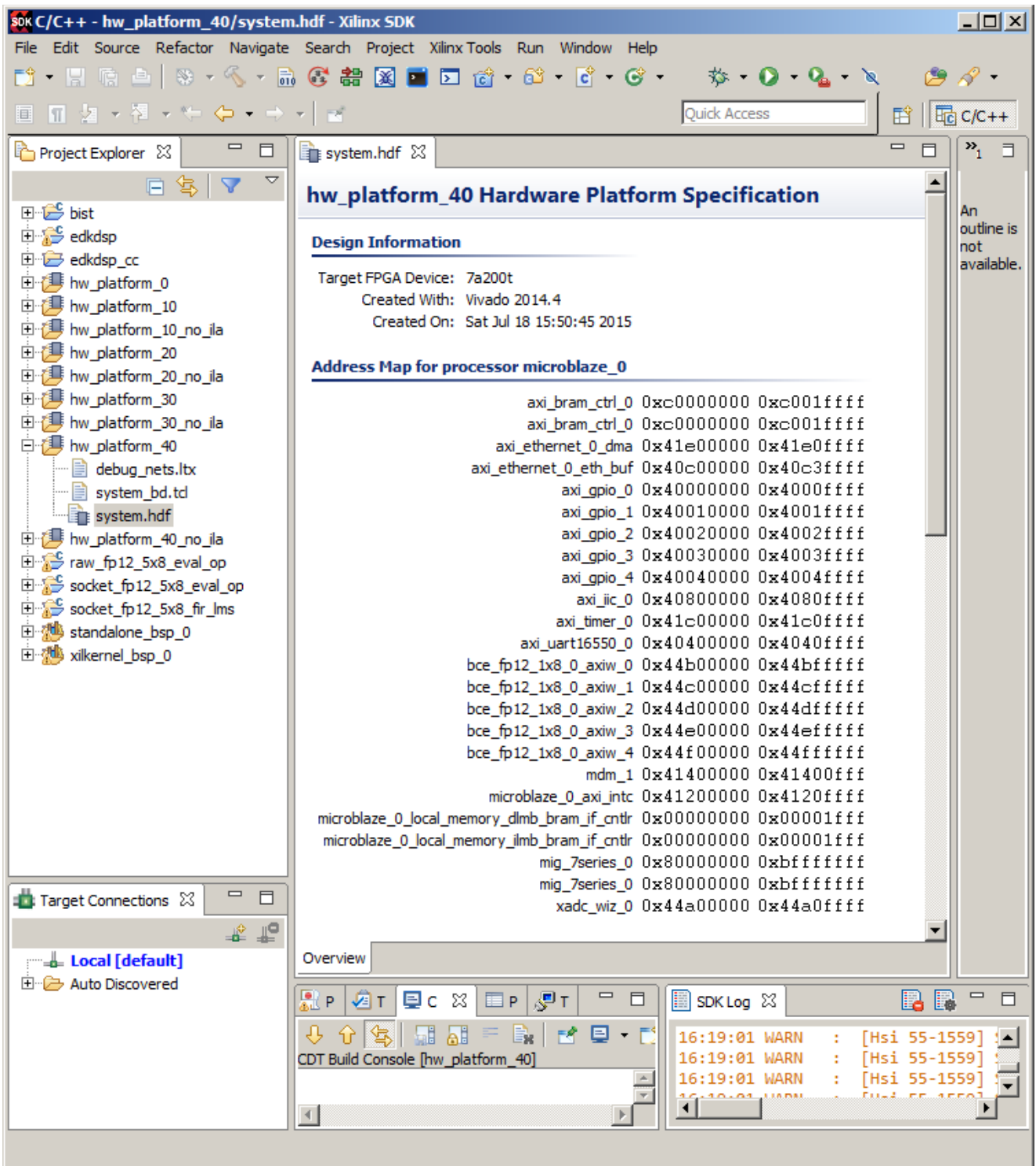


Figure 15: All projects are compiled. See IP blocks present in the design

The “edkdsp” project is extending the “bist_app” with tests of the EdkDSP accelerator, without Ethernet.

- The “raw_axi_bce_fp12_1x8_eval_op” project is extending the “edkdsp” with RAW version of the lwIP Ethernet www server GUI, the TFTP file server and the RAM based file system.
- The “socket_axi_bce_fp12_1x8_eval_op” project is extending the “edkdsp” with SOCKET version of the lwIP Ethernet www server GUI, the TFTP file server and the RAM based file system.
- The “socket_axi_bce_fp12_1x8_fir_lms” project is demonstrating the floating point FIR filter and LMS filter computation on a single (8xSIMD) EdkDSP accelerator with the SOCKET version of the lwIP Ethernet www server GUI, the SOCKET version of the TFTP file server and the RAM based file system.

Connect the jtag and serial line USB cables to your AC701 board. Switch ON the board.

3.3 Boot of the bitstream in Vivado 2014 Hardware Manager

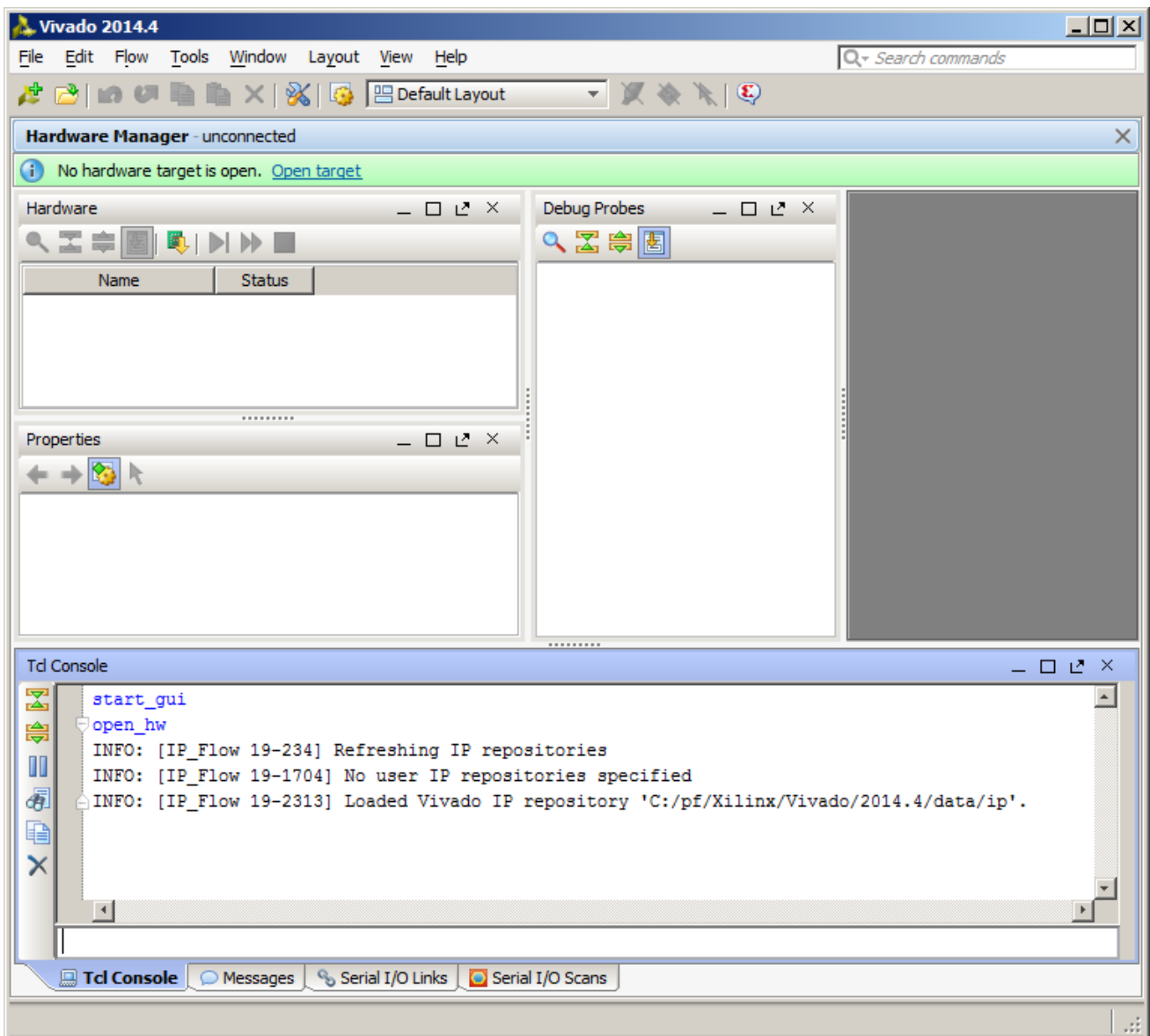


Figure 16: Select and open the Hardware Manager tool from the Vivado 2014.4 initial menu

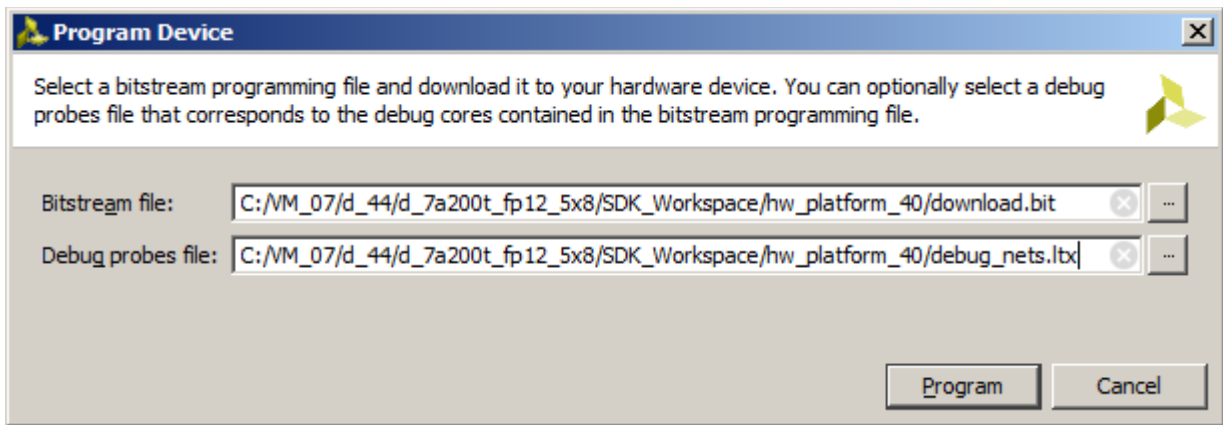


Figure 17: Select download.bit to program the Artix device and file defining the debug nets

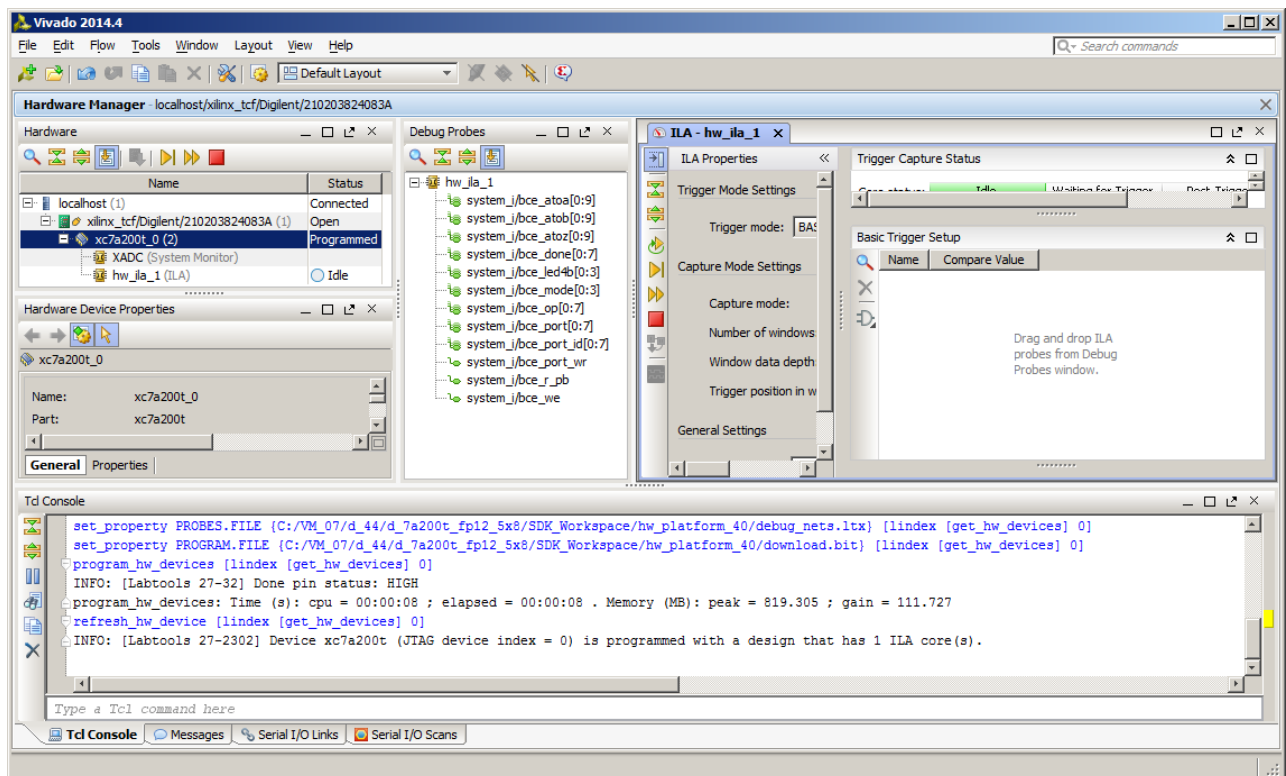


Figure 18: Board HW is booted and debug nets are identified

Open the Vivado 2014.4 Hardware manager tool. Select “open target”, confirm default server setup and auto-connect via USB/jtag to the board. See Figure 16. Select bitstream file and debug nets file. See Figure 17.

C:\VM_07\d_44\d_7a200t_fp12_5x8\SDK_Workspace\hw_platform_40\download.bit

C:\VM_07\d_44\d_7a200t_fp12_5x8\SDK_Workspace\hw_platform_40\debug_nets.ltx

The Artix part is configured by jtag and the Vivado 2014.4 Hardware Manager tool is prepared for ILA debugging using the debug probes. See Figure 18.

On PC, start PuTTY terminal. Set 9600 baud and “Flow control” to None. See Figure 19 and Figure 20.

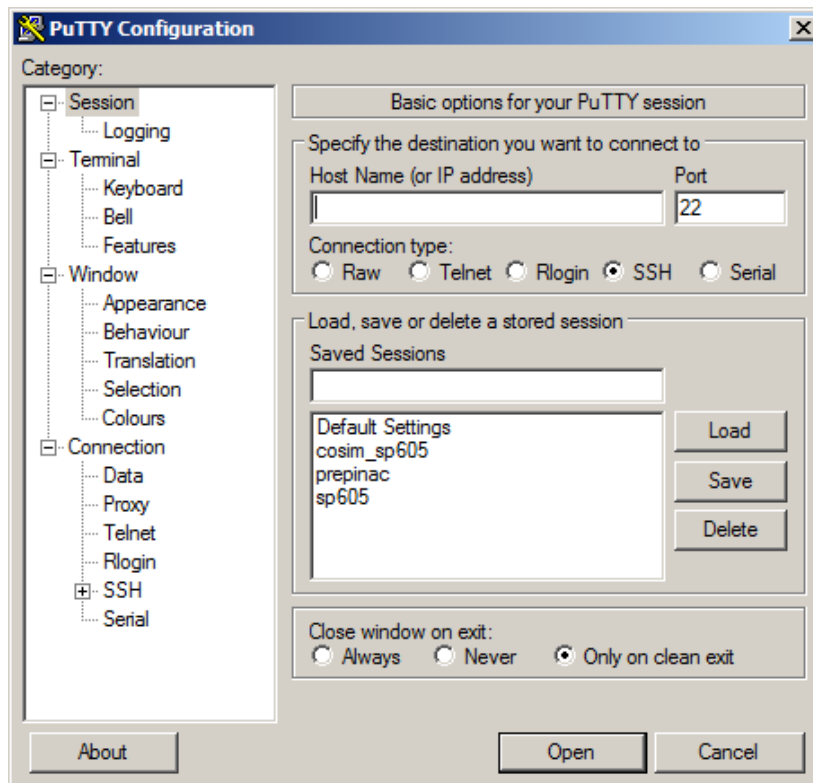


Figure 19: Open PuTTY terminal

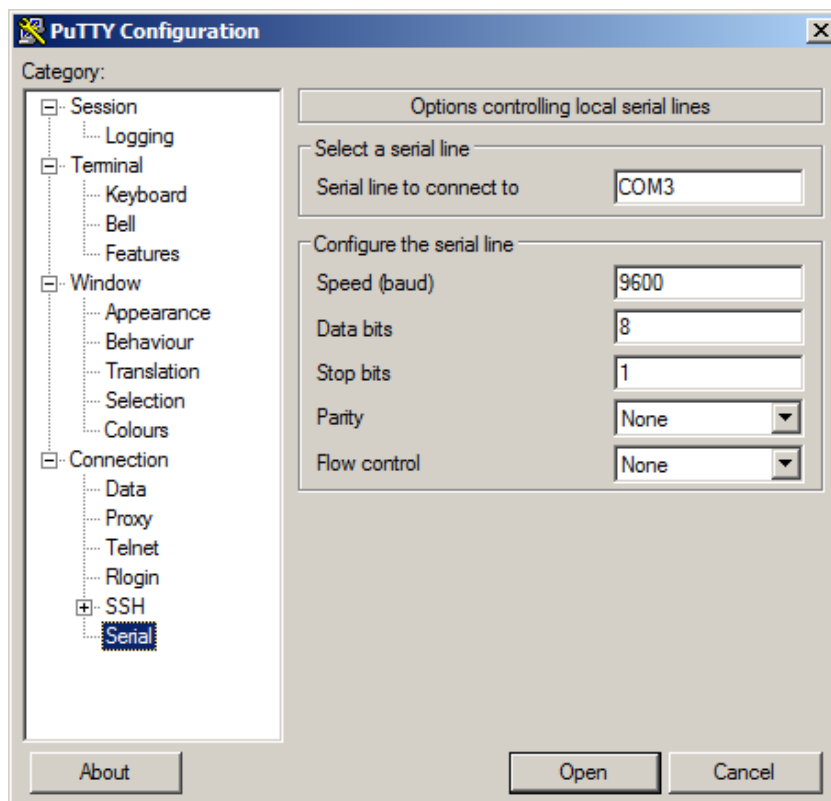


Figure 20: Select “Serial”, select your COM port, set speed to 9600 and flow control to None

3.4 Ethernet point to point connection with PC

The SDK SW projects included in this evaluation package demonstrate integration of the UTIA EdkDSP accelerator IPs cores in MicroBlaze SoC the Xilinx 1Gb Ethernet controller. The connection to the Ethernet is based on two versions of the LwIP SW:

- Raw versions of SDK SW projects use raw version of the LwIP 14.1 library without real-time OS.
- Socket versions of SW projects use the socket version of LwIP 14.1 on top of the Xilinx XilKernel.

Set your PC Ethernet connection to point-to-point with the fixed IP address: 192.168.1.100
All included evaluation designs are setting the IP address of the AC701 board to: 192.168.1.10
This setting enables the direct point to point Ethernet connection with PC.

3.5 Boot of the SW application

The SW `bist_app.elf` application from the “`bist_app`” project can be downloaded to the DDR3 memory and started. Select the “`bist_app`” project in the project navigator.

In SDK, select:

Run -> Run Configuration -> Xilinx C/C++ ELF

Click on the “New launch configuration” in the Run configuration screen and the `bist_app.elf` project executable is ready for download to DDR3 via the jtag cable. Click on “Run” button to download the executable. See Figure 21. Click on the “Program” button.

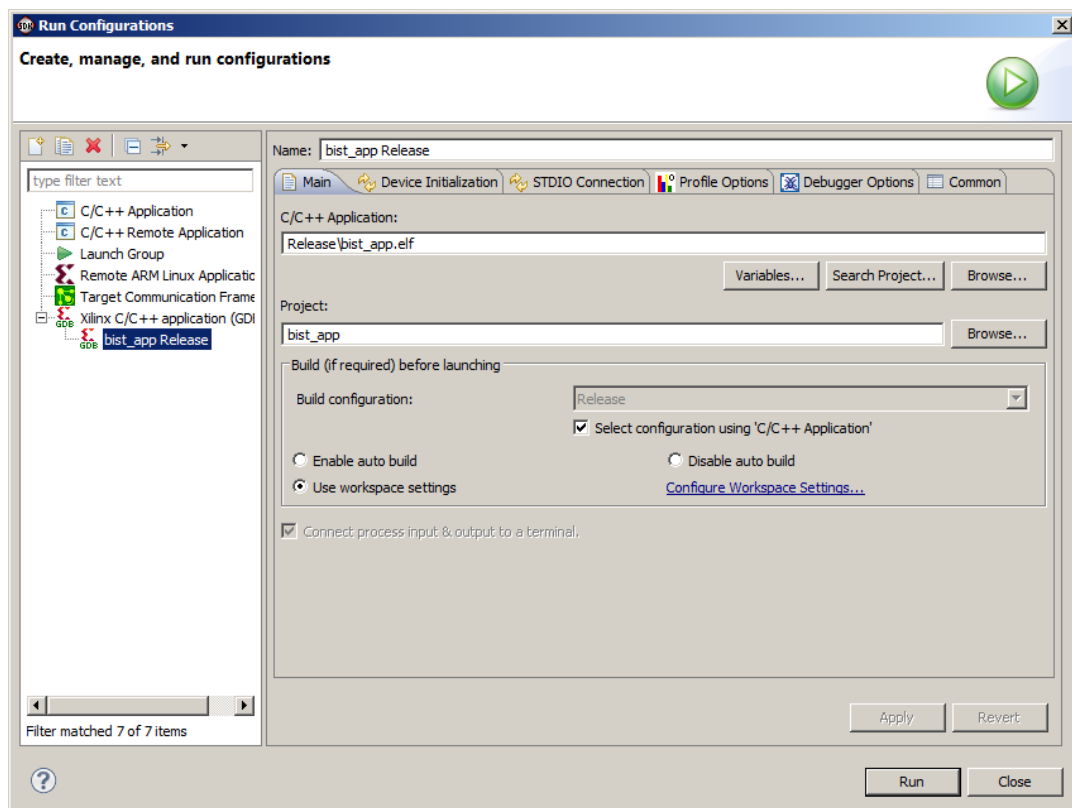


Figure 21: Select “`bist_app.elf`” code

Run the application `bist_app.elf` by clicking on Run. See Figure 22.

```
COM3 - PuTTY
*****
*****
**      Xilinx Artix-7 FPGA AC701 Evaluation Kit      **
*****
*****
Choose Feature to Test:
1: UART Test
2: LED Test
3: IIC Test
4: TIMER Test
5: ROTARY Test
6: SWITCH Test
7: LCD Test
8: DDR3 External Memory Test
9: BRAM Internal Memory Test
A: ETHERNET Loopback Test
B: BUTTON Test
0: Exit
2

*****
*****
**      AC701 - GPIO LED Test      **
*****
*****
Watch the LEDs
Press any key to return to main menu
Choose Feature to Test:
1: UART Test
2: LED Test
3: IIC Test
4: TIMER Test
5: ROTARY Test
6: SWITCH Test
7: LCD Test
8: DDR3 External Memory Test
9: BRAM Internal Memory Test
A: ETHERNET Loopback Test
B: BUTTON Test
0: Exit
0
Good-bye!
```

Figure 22: Run `bist_app.elf` and select tests from the terminal keyboard (PC)

The Xilinx `bist_app` demo serves for test of standard MicroBlaze peripherals. Stop and terminate MicroBlaze program execution from the SDK. This is done by click on the red square icon on top of the SDK Debugger console and next on the X icon to close the debugger session in SDK.

Download again the bitstream from Vivado 2014.4 Hardware Manager. Select the `edkdsp` SW project for download in SDK. Run it. See the extended menu enabling tests of the EdkDSP accelerator. See Figure 23.

```
COM3 - PuTTY
*****
**      AC701 - GPIO LED Test      **
*****

Watch the LEDs
Press any key to return to main menu
Choose Feature to Test:
1: UART Test
2: LED Test
3: IIC Test
4: TIMER Test
5: ROTARY Test
6: SWITCH Test
7: LCD Test
8: DDR3 External Memory Test
9: BRAM Internal Memory Test
A: ETHERNET Loopback Test
B: BUTTON Test
0: Exit
0
Good-bye!

*****
**      Xilinx Artix-7 FPGA AC701 Evaluation Kit      **
*****

Choose Feature to Test:
1: UART Test
2: LED Test
3: IIC Test
4: TIMER Test
5: ROTARY Test
6: SWITCH Test
7: LCD Test
8: DDR3 External Memory Test
9: BRAM Internal Memory Test
A: ETHERNET Loopback Test
B: BUTTON Test
C: EdkDSP Eval Op
0: Exit
```

Figure 23: Run the edkdsp.elf application and select the EdkDSP Eval Op test

Select the “C” option from the terminal keyboard to run test of the EdkDSP accelerator. See Figure 24.

```

COM3 - PuTTY
1: UART Test
2: LED Test
3: IIC Test
4: TIMER Test
5: ROTARY Test
6: SWITCH Test
7: LCD Test
8: DDR3 External Memory Test
9: BRAM Internal Memory Test
A: ETHERNET Loopback Test
B: BUTTON Test
C: EdkDSP Eval Op
0: Exit
C
-- Entering main() --

Tests of vector operations.
MBO : (EdkDSP 8xSIMD) Capabilities1 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities2 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities3 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities4 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities5 = 13FFFF

ah=0 bh=0 zh=0
MBO : (EdkDSP 8xSIMD) VZ2A 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VB2A 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VZ2B 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VA2B 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VADD 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VADD_BZ2A 'worker1' .. OK
MBO : (EdkDSP 8xSIMD) VADD_AZ2B 'worker1' .. OK
MBO : (EdkDSP 8xSIMD) VSUB 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VSUB_BZ2A 'worker1' .. OK
MBO : (EdkDSP 8xSIMD) VSUB_AZ2B 'worker1' .. OK
MBO : (EdkDSP 8xSIMD) VMULT 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VMULT_BZ2A 'worker1' . OK
MBO : (EdkDSP 8xSIMD) VMULT_AZ2B 'worker1' . OK
MBO : (EdkDSP 8xSIMD) VPROD 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VMAC 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VMSUBAC 'worker1' .... OK
MBO : (EdkDSP 8xSIMD) VPROD_S8 'worker1' ... OK
MBO : (EdkDSP 8xSIMD) VDIV 'worker1' ..... OK
Press any key to return to main menu

```

Figure 24: The EdkDSP basic vector floating point operations have been tested

Press any key to return to menu. Type 0 to terminate program. Stop and terminate MicroBlaze program execution from the SDK.

Download again the bitstream from Vivado 2014.4 Hardware Manager. Select and run the **raw_axi_bce_fp12_1x8_eval_op** SW project . See Figure 25.


```

COM3 - PuTTY
Initializing MFS at 0x802598B8
Done.
Located index.html

-----lwIP RAW Mode Demo Application -----
Start PHY autonegotiation
Waiting for PHY to complete autonegotiation.
autonegotiation complete
auto-negotiated link speed: 1000
DHCP Timeout
Configuring default IP of 192.168.1.10
Board IP:      192.168.1.10
Netmask :     255.255.255.0
Gateway :     192.168.1.1
Initializing MFS at 0x802598B8
Done.
Located index.html

-----
Server      Port Connect With..
-----
tftp server  69 $ tftp -i 192.168.1.10 PUT <source-file>
http server  80 Point your web browser to http://192.168.1.10

```

Figure 25: Select “raw_axi_bce_fp12_eval_opl.elf application to test the lwIP services in RWW mode

The RAW version of the TFTP server and the RAW version of the HTTP server have been started on the Artix7 MicroBlaze processor. See Figure 25.

Open WWW browser (Internet Explorer) in your PC and connect it to the board address <http://192.168.1.10/> See Figure 27.

Support script files are downloaded to the PC from the Artix7 MicroBlaze DDR3-based file system and the interface page is started. See Figure 26.

The loaded scripts support two buttons in the GUI. See Figure 27. The “**Update Status**” button serves to get the DIP switches status. The “**Toggle LEDs**” button is toggling the led output on the board and starts the EdkDSP accelerator evaluation. See Figure 28.

Click on the “**Toggle LEDs**” button to evaluate elementary vector functions of the EdkDSP accelerator IP.

- The SW application is testing presence of an updated firmware in the RAM based file system of the board. If it is not present, the default firmware is used.
- The file FP1101.TXT is open for WR in the RAM based file system. It will store text messages from the tested EdkDSP accelerator.
- The capabilities of all 6 EdkDSP accelerators are displayed next. This information is based on the reply from the initialised accelerators.
- Test is performed.
- Finally the top directory of the RAM based file system is listed together with the information about used and free blocks.

See Figure 28.

```
COM3 - PuTTY
Initializing MFS at 0x802598B8
Done.
Located index.html

-----lwIP RAW Mode Demo Application -----
Start PHY autonegotiation
Waiting for PHY to complete autonegotiation.
autonegotiation complete
auto-negotiated link speed: 1000
DHCP Timeout
Configuring default IP of 192.168.1.10
Board IP:      192.168.1.10
Netmask :     255.255.255.0
Gateway :     192.168.1.1
Initializing MFS at 0x802598B8
Done.
Located index.html

          Server  Port Connect With..
-----
          tftp server  69 $ tftp -i 192.168.1.10 PUT <source-file>
          http server  80 Point your web browser to http://192.168.1.10

http GET: index.html
http GET: css/main.css
http GET: images/logo.gif
http GET: yui/yahoo.js
http GET: yui/dom.js
http GET: yui/event.js
http GET: js/main.js
http GET: yui/conn.js
http GET: yui/anim.js
http POST: switch state: 0
http POST: ledstatus: 0
```

Figure 26: The Java Script has been loaded from the FPGA RAM based file system to your browser

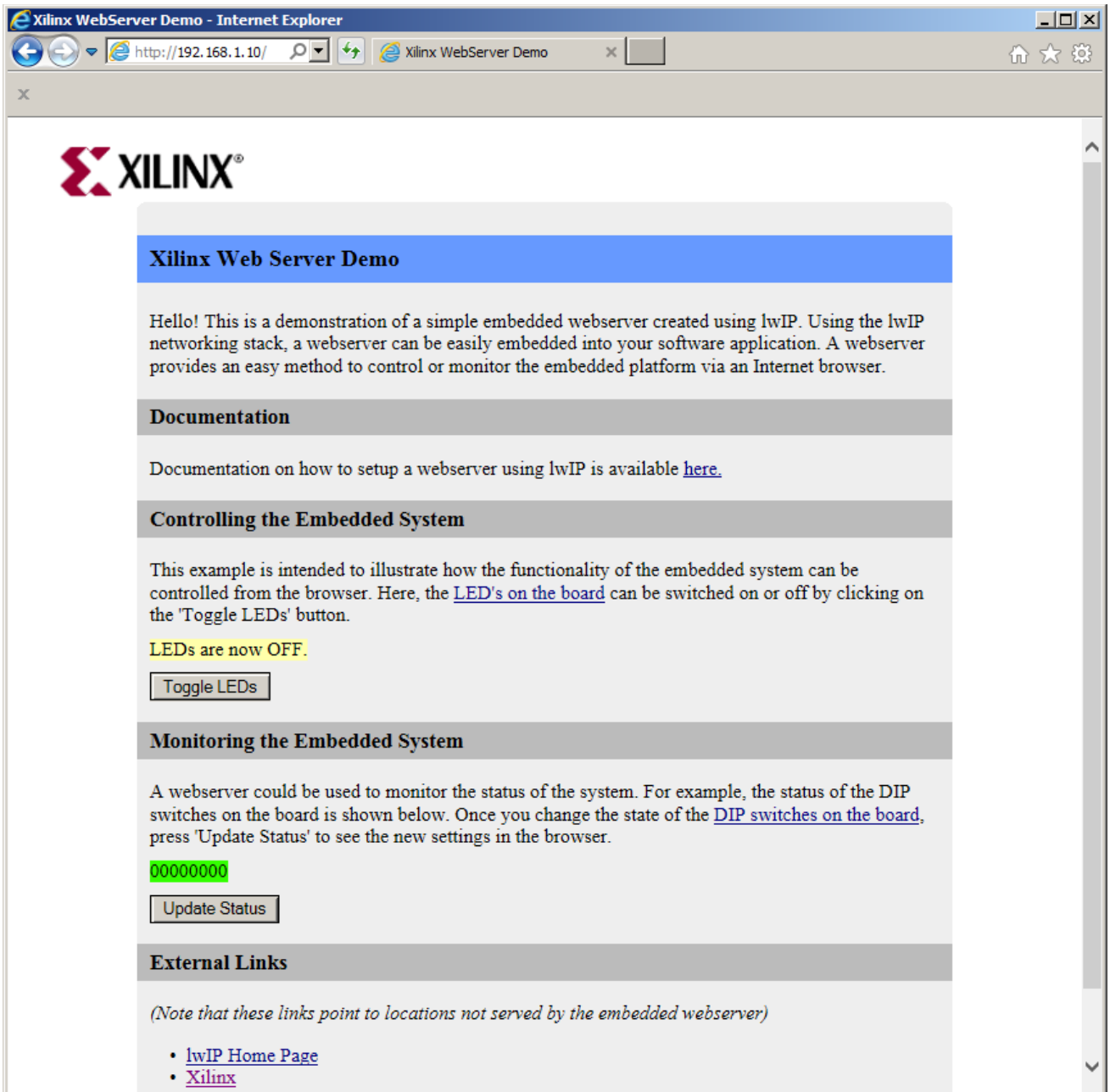


Figure 27: The demo www server GUI

```
COM3 - PuTTY
Tests of vector operations.
File FP1101P0.DEC not found.
Default firmware will be used.
File FP1101P1.DEC not found.
Default firmware will be used.
File FP1101.TXT created for wr
MB0 : (EdkDSP 8xSIMD) Capabilities1 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities2 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities3 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities4 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities5 = 13FFFF

ah=0 bh=0 zh=0
MB0 : (EdkDSP 8xSIMD) VZ2A 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VB2A 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VZ2B 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VA2B 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VADD 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VADD_BZ2A 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VADD_AZ2B 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VSUB 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VSUB_BZ2A 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VSUB_AZ2B 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VMULT 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMULT_BZ2A 'worker1' . OK
MB0 : (EdkDSP 8xSIMD) VMULT_AZ2B 'worker1' . OK
MB0 : (EdkDSP 8xSIMD) VPROD 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMAC 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMSUBAC 'worker1' .... OK
MB0 : (EdkDSP 8xSIMD) VPROD_S8 'worker1' ... OK
MB0 : (EdkDSP 8xSIMD) VDIV 'worker1' ..... OK
Blocks_used 238
Blocks_free 62
Directory css 00000003
Directory images 00000005
Directory js 00000003
Directory yui 00000007
index.html 00000c0b
FP1101.TXT 0000061e
http POST: ledstatus: FFFFFFFF
```

Figure 28: Test of basic operations of EdkDSP IP core

Close the web browser. Stop and terminate MicroBlaze program execution from the SDK.

Download again the bitstream from Vivado 2014.4.

Select the **socket_axi_bce_fp12_1x8_eval_op** project for download and run it from the SDK. See Figure 29.

```

COM3 - PuTTY
Initializing MFS at 0x80328C4C
Done.
Located index.html

-----lwIP Socket Mode Demo Application -----
Start PHY autonegotiation
Waiting for PHY to complete autonegotiation.
autonegotiation complete
auto-negotiated link speed: 1000
ERROR: DHCP request timed out
Configuring default IP of 192.168.1.10
Board IP: 192.168.1.10
Netmask : 255.255.255.0
Gateway : 192.168.1.1

          Server   Port Connect With..
-----
tftp server      69 $ tftp -i 192.168.1.10 PUT <source-file>
http server      80 Point your web browser to http://192.168.1.10

```

Figure 29: Start the `socket_axi_bce_fp12_1x8_eval_op.elf` demo application, working on top of the XilKernel OS

The SOCKET version of the TFTP server and the HTTP server run on the Artix7 MicroBlaze processor. Open www browser (Internet Explorer) and connect to the board address: <http://192.168.1.10/> See Figure 29.

Click on the **Toggle LEDs** button to toggle the led output on the board and to starts the EdkDSP accelerator evaluation. The SOCKET version of the server supports use of both buttons in parallel. See Figure 30.

```

COM3 - PuTTY
      http server      80 Point your web browser to http://192.168.1.10

http POST: switch state: 0
http POST: ledstatus: 0

Tests of vector operations.
File FP1101P0.DEC not found.
Default firmware will be used.
File FP1101P1.DEC not found.
Default firmware will be used.
File FP1101.TXT created for wr
MBO : (EdkDSP 8xSIMD) Capabilities1 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities2 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities3 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities4 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities5 = 13FFFF

ah=0 bh=0 zh=0
MBO : (EdkDSP 8xSIMD) VZ2A 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VB2A 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VZ2B 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VA2B 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VADD 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VADD_BZ2A 'worker1' .. OK
MBO : (EdkDSP 8xSIMD) VADD_AZ2B 'worker1' .. OK
MBO : (EdkDSP 8xSIMD) VSUB 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VSUB_BZ2A 'worker1' .. OK
MBO : (EdkDSP 8xSIMD) VSUB_AZ2B 'worker1' .. OK
MBO : (EdkDSP 8xSIMD) VMULT 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VMULT_BZ2A 'worker1' . OK
MBO : (EdkDSP 8xSIMD) VMULT_AZ2B 'worker1' . OK
MBO : (EdkDSP 8xSIMD) VPROD 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VMAC 'worker1' ..... OK
MBO : (EdkDSP 8xSIMD) VMSUBAC 'worker1' .... OK
MBO : (EdkDSP 8xSIMD) VPROD_S8 'worker1' ... OK
MBO : (EdkDSP 8xSIMD) VDIV 'worker1' ..... OK
Blocks_used 238
Blocks_free 62
Directory css 00000003
Directory images 00000005
Directory js 00000003
Directory yui 00000007
index.html 00000c0b
FP1101.TXT 0000061e
http POST: ledstatus: FFFFFFFF

```

Figure 30: Test of vector operations is started from the www browser GUI. It is served by the lwIP library working on top of the XilKernel

Close the web browser. Close the socket based application running on the Artix7 from the SDK.

Download again the bitstream from Vivado 2014.4. Select the socket_axi_bce_fp12_1x8_fir_lms project for download and run it. See Figure 31.


```
COM3 - PuTTY
Initializing MFS at 0x804B71F0
Done.
Located index.html

-----lwIP Socket Mode Demo Application -----
Start PHY autonegotiation
Waiting for PHY to complete autonegotiation.
autonegotiation complete
auto-negotiated link speed: 1000
ERROR: DHCP request timed out
Configuring default IP of 192.168.1.10
Board IP: 192.168.1.10
Netmask : 255.255.255.0
Gateway : 192.168.1.1

Server      Port Connect With..
-----
tftp server 69 $ tftp -i 192.168.1.10 PUT <source-file>
http server 80 Point your web browser to http://192.168.1.10
```

Figure 31: Start the `socket_axi_bce_fp12_1x8_fir_lms.elf` application

The SOCKET version of the TFTP and HTTP servers have been started on the Artix7 MicroBlaze processor. Open www browser (Internet Explorer) client and connect to the board address: <http://192.168.1.10/> Click on the **Toggle LEDs** button to toggle the led output on the board and starts the FIR and LMS filter computation on single (8xSIMS) EdkDSP accelerator. See Figure 32.

```

COM3 - PuTTY
Initializing MFS at 0x804B71F0
Done.
Located index.html

-----lwIP Socket Mode Demo Application -----
Start PHY autonegotiation
Waiting for PHY to complete autonegotiation.
autonegotiation complete
auto-negotiated link speed: 1000
ERROR: DHCP request timed out
Configuring default IP of 192.168.1.10
Board IP: 192.168.1.10
Netmask : 255.255.255.0
Gateway : 192.168.1.1

Server      Port Connect With..
-----
tftp server 69 $ tftp -i 192.168.1.10 PUT <source-file>
http server 80 Point your web browser to http://192.168.1.10

http POST: switch state: 0
http POST: ledstatus: 0

File FP1124P0.DEC not found.
Default firmware will be used.
File FP1124P1.DEC not found.
Default firmware will be used.
File FP1124.TXT created for wr
MBO : (EdkDSP 8xSIMD) Capabilities1 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities2 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities3 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities4 = 13FFFF
MBO : (EdkDSP 8xSIMD) Capabilities5 = 13FFFF
MBO : Generating far-end signal ...
MBO : (EdkDSP 8xSIMD) FIR filter ... 1111 MFLOPs
MBO : Adding near-end signal ...
MBO : (EdkDSP 8xSIMD) LMS filter ... 754 MFLOPs
MBO : LMS filter ...
Step 99 of 100: LMS acceleration 215x.
OK
Blocks_used 238
Blocks_free 62
Directory css 00000003
Directory images 00000005
Directory js 00000003
Directory yui 00000007
index.html 00000c0b
FP1124.TXT 000007d0
http POST: ledstatus: FFFFFFFF

```

Figure 32: The FIR and LMS computation is started from the web browser GUI

The performance for FIR and LMS is displayed and the speedup in comparison to the MicroBlaze is reported during the MicroBlaze verification run. The result from the EdkDSP is identical to the MicroBlaze result. Close browser. Stop the Artix7 application.

3.6 Use of the C compiler for the EdkDSP firmware with download from Ethernet

This section is describing the use of the UTIA EdkDSP C compiler to recompile the firmware for the PicoBlaze6 controller present in each of the five (8xSIMD) EdkDSP accelerator IP cores in the AC701 board.

In SDK Project Explorer, open the project **edkdsp_cc** and the subdirectory **edkdsp_cc/a**. See Figure 33. It contains C source code of the EdkDSP accelerator firmware and Ubuntu scripts for the compilation. The compiled versions of firmware are already present in the demonstrated applications in form of headers for the MicroBlaze applications. This helps to evaluate the EdkDSP accelerators without installation of the C compiler for the EdkDSP.

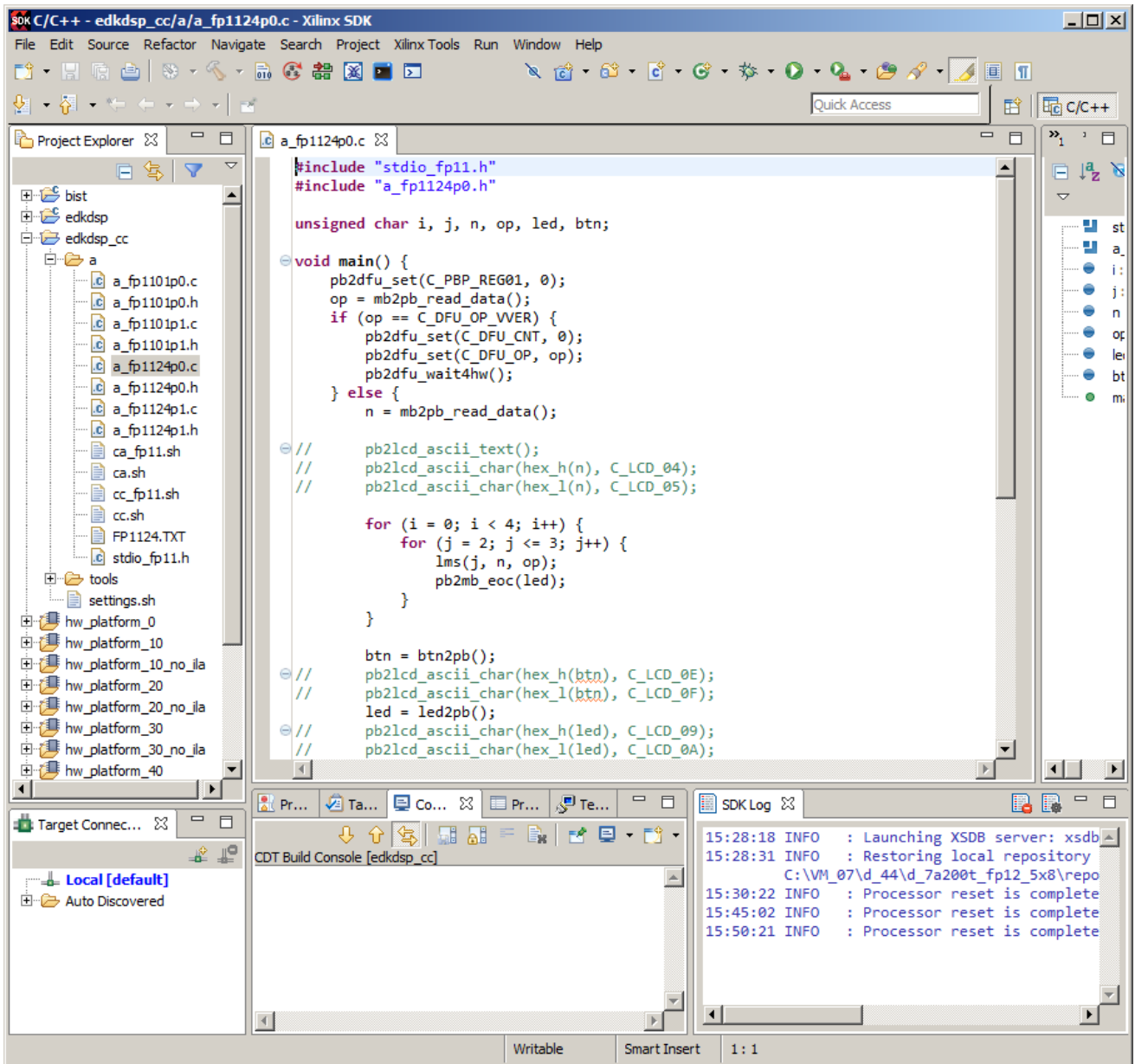


Figure 33: Evaluate the included C code for reprogramming of the EdkDSP accelerators

The UTIA EdkDSP C compiler is provided as implemented as several Ubuntu binary applications. The “VMware Workstation 12 player” software is used to run 32bit Ubuntu image on Windows 7 (64bit) PC. The “VMware player” 3.0.0 software can be also used to run 32bit Ubuntu image on Windows 7 (32bit) PC.

The Ubuntu image used in UTIA needs one DVD (4.7GB) for installation. That is why it is not included as part of the evaluation package. If you would need this image, write an email request to kadlec@utia.cas.cz to get it with correct Ubuntu image from UTIA (free of charge).

Install the VMware Player 3.0.0 or the VMware Workstation 12 player software on your PC. In VMware Player open the Ubuntu_EdkDSP package. See Figure 34.

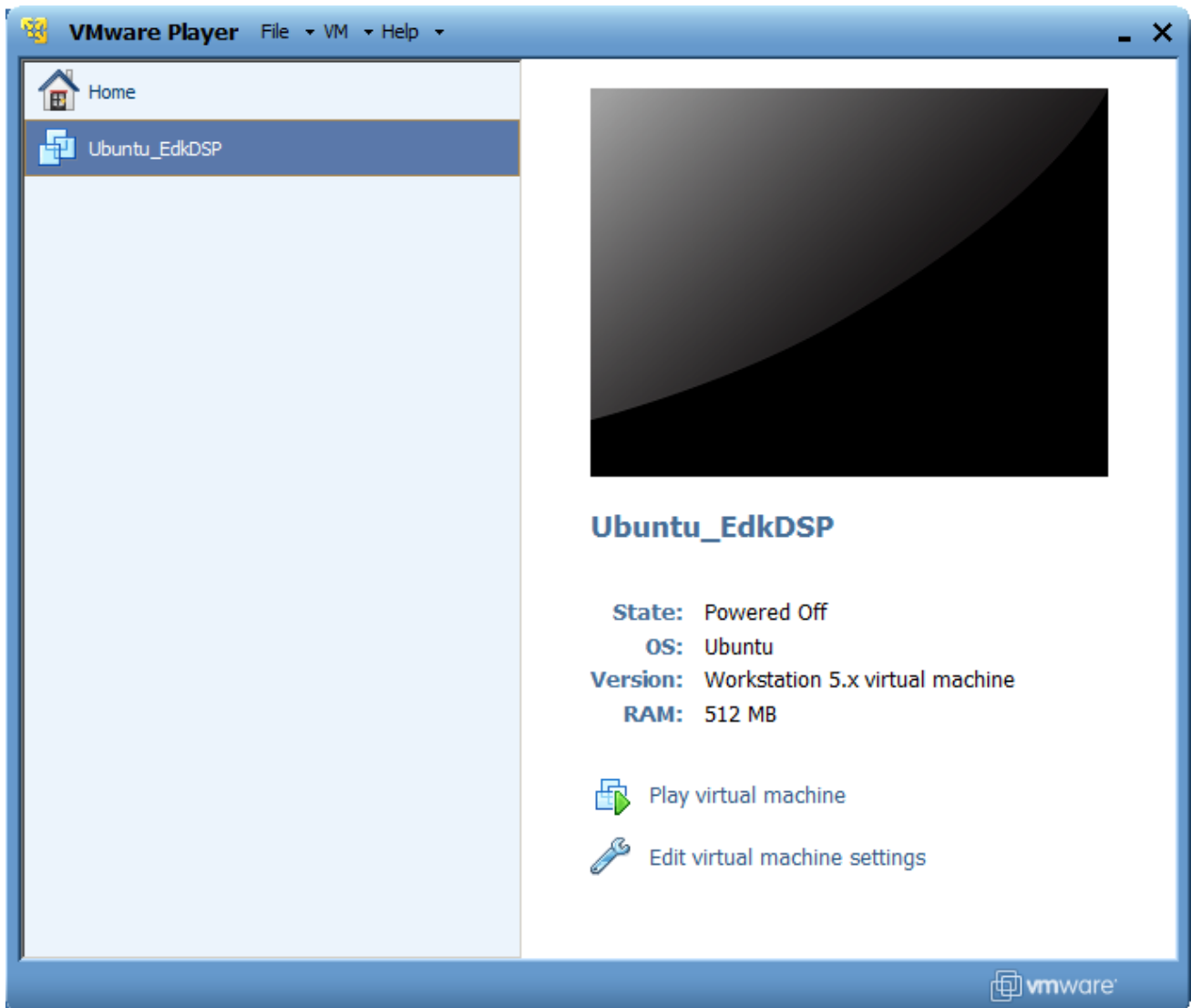


Figure 34: Start the VMware Player to run the C compiler for the EdkDSP accelerators as an Ubuntu binary user application

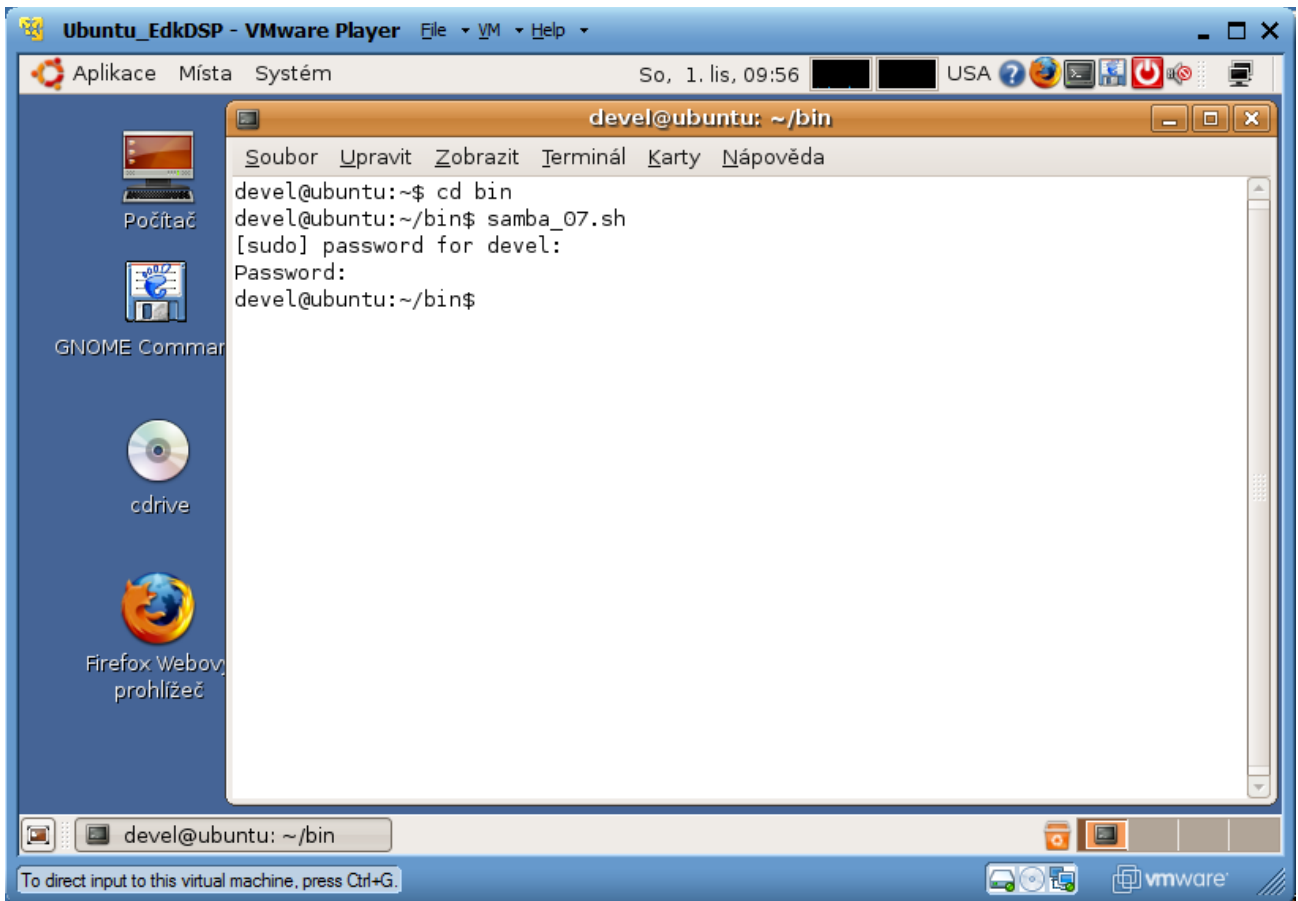


Figure 35: Mount the Windows 7 directory `c:\VM_07` as `/mnt/cdrive` in Ubuntu

Open the VMware Player and select the “Ubuntu_EdkDSP” image. The Ubuntu will start. Login as:

User: devel

Pswd: devuser

The PC directory `c:\VM_07` needs to be shared by Windows 7 with Ubuntu.

- In Windows 7, set the directory `c:\VM_07` and its subdirectories as shared for Read and Write.
- In Ubuntu, open terminal and mount the PC directory `c:\VM_07` to Ubuntu. The Windows 7 `c:\VM_07` directory is mounted to the Ubuntu OS as: **`/mnt/cdrive`**

This process is automated in script `samba_07.sh`. Script has to be updated to reflect the PC user name and to identify the virtual Ethernet connection of Win 7 PC and VMware Player for the virtual Ubuntu OS.

See Figure 35.

In Ubuntu terminal, change the directory to:

```
$ cd /mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/SDK_Workspace/edkdsp_cc
```

The EdkDSP C compiler utilities have to be on the Ubuntu PATH. This is done by sourcing the `settings.sh` script in this directory. Type in Ubuntu terminal. See Figure 36.

```
$ source settings.sh
```

In Ubuntu terminal, change the directory to the example directory “`./a`” (See Figure 36):

```
$ cd a
```

```
devel@ubuntu:/mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/SDK_Workspace/edkdsp_cc/a$
```

```
Ubuntu_EdkDSP - VMware Player  File - VM - Help -
Aplikace  Místa  Systém  So, 1. lis, 10:35  USA
devel@ubuntu: /mnt/cdrive/d_34_7a/d_7a200t_fp12_6x8/SDK_Workspace/edkdsp_cc/a
Soubor  Upravit  Zobrazit  Terminál  Karty  Nápověda
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$
devel@ubuntu:/$ cd /mnt/cdrive/d_34_7a/d_7a200t_fp12_6x8/SDK_Workspace/edkdsp_cc
devel@ubuntu:/mnt/cdrive/d_34_7a/d_7a200t_fp12_6x8/SDK_Workspace/edkdsp_cc$ ls
a  settings.sh  tools
devel@ubuntu:/mnt/cdrive/d_34_7a/d_7a200t_fp12_6x8/SDK_Workspace/edkdsp_cc$ source settings.sh
EdkDSP environment set to '/mnt/cdrive/d_34_7a/d_7a200t_fp12_6x8/SDK_Workspace/edkdsp_cc'
devel@ubuntu:/mnt/cdrive/d_34_7a/d_7a200t_fp12_6x8/SDK_Workspace/edkdsp_cc$ cd a
devel@ubuntu:/mnt/cdrive/d_34_7a/d_7a200t_fp12_6x8/SDK_Workspace/edkdsp_cc/a$ ls
a_fp1101p0.c  a_fp1101p1.c  a_fp1124p0.c  a_fp1124p1.c  ca_fp11.sh  cc_fp11.sh  stdio_fp11.h
a_fp1101p0.h  a_fp1101p1.h  a_fp1124p0.h  a_fp1124p1.h  ca.sh  cc.sh
devel@ubuntu:/mnt/cdrive/d_34_7a/d_7a200t_fp12_6x8/SDK_Workspace/edkdsp_cc/a$
```

Figure 36: Source the path to the EdkDSP C compiler tools

In SDK, open the C source code of the current firmware for the EdkDSP accelerator in the file **edkdsp_cc\a\fp1101p0.c**

See the original listing in Figure 37.

To demonstrate the compilation and new firmware download via Ethernet, we will change the message going from EdkDSP PicoBlaze processor to the MicroBlaze and to the FP1101.TXT log file from **“I=00”**; to **“Input=00”**.

Uncomment the four commented lines from **// pb2mb_Write ('n');** to **// pb2mb_Write ('t');**

See Figure 37.

Save the modifications in Win 7 SDK editor.

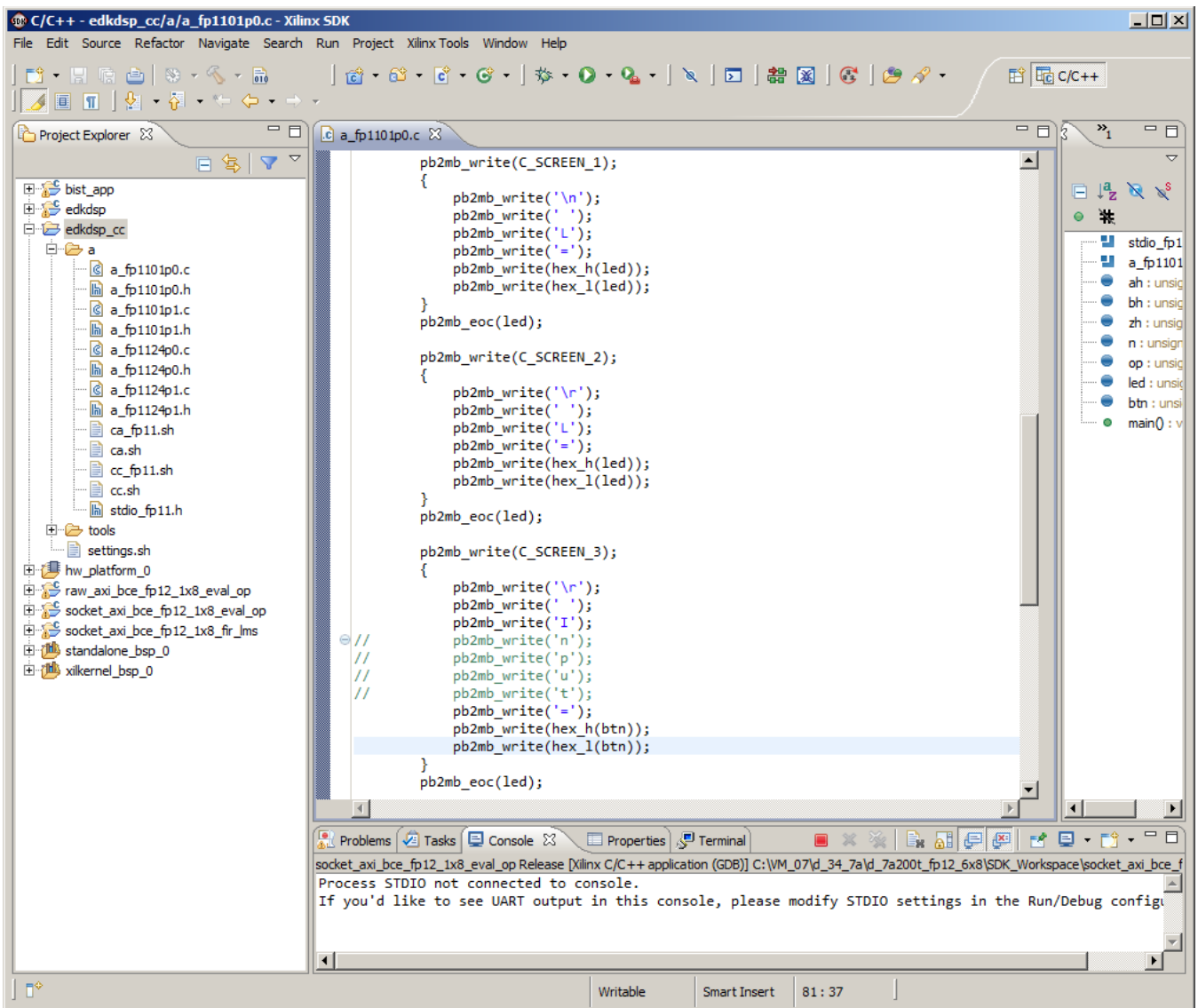


Figure 37: See the details of communication from the accelerator to MicroBlaze in the original code

We will demonstrate the complete process related to the compilation, download of results from Artix7 to the PC.

- Download again the bitstream from Vivado 2014.4 Hardware Manager.
- Start the application **socket_axi_bce_fp12_5x8_eval_op.elf**
- Open the www browser
- Start the demo run by clicking on the Toggle LEDs button.

See Figure 38 and Figure 39.

```

COM3 - PuTTY
-----
Server      Port Connect With..
-----
tftp server  69 $ tftp -i 192.168.1.10 PUT <source-file>
http server  80 Point your web browser to http://192.168.1.10

http POST: switch state: 0
http POST: ledstatus: 0

Tests of vector operations.
File FP1101P0.DEC not found.
Default firmware will be used.
File FP1101P1.DEC not found.
Default firmware will be used.
File FP1101.TXT created for wr
MB0 : (EdkDSP 8xSIMD) Capabilities1 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities2 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities3 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities4 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities5 = 13FFFF

ah=0 bh=0 zh=0
MB0 : (EdkDSP 8xSIMD) VZ2A 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VB2A 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VZ2B 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VA2B 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VADD 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VADD_BZ2A 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VADD_AZ2B 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VSUB 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VSUB_BZ2A 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VSUB_AZ2B 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VMULT 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMULT_BZ2A 'worker1' . OK
MB0 : (EdkDSP 8xSIMD) VMULT_AZ2B 'worker1' . OK
MB0 : (EdkDSP 8xSIMD) VPROD 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMAC 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMSUBAC 'worker1' .... OK
MB0 : (EdkDSP 8xSIMD) VPROD_S8 'worker1' ... OK
MB0 : (EdkDSP 8xSIMD) VDIV 'worker1' ..... OK

Blocks_used 238
Blocks_free 62
Directory css 00000003
Directory images 00000005
Directory js 00000003
Directory yui 00000007
index.html 00000c0b
FP1101.TXT 0000061e
http POST: ledstatus: FFFFFFFF

```

Figure 38: Test has been performed and the tested EdkDSP accelerator created data file FP1101.TXT in the RAM file system located in the DDR3 of the AC701 board

Open the TFTP application on your PC as a TFTP client connected to the Artix7 host 192.168.1.10 with Port 69. See Figure 39.

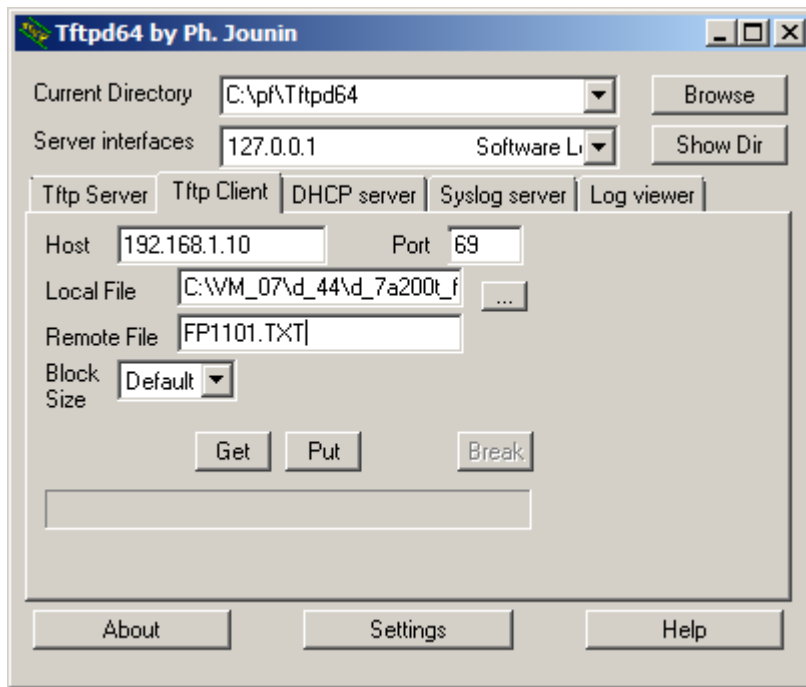


Figure 39: Start TFTP client and get the file FP1101.TXT from the Artix7 FPGA to PC via Ethernet
Select Local (PC) file to:

c:\VM_07\d_44\d_7a200t_fp12_5x8\SDK_Workspace\edkdsp_cc\a\FP1101.TXT

and the Remote File (Artix7 file system) to:

FP1101.TXT

Click on **“Get”** to download the file to PC from the board.

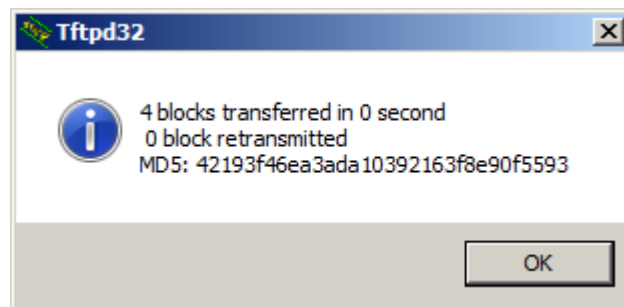


Figure 40: Ssize of FP1101.TXT received from the Artix7 FPGA to PC via Ethernet. Confirm OK
The EdkDSP firmware after the compilation is presented in Figure 39.

```

Screen1
L=00
Screen2
L=00
Screen3
I=00
Screen4
ah=00 bh=00 zh=00
...

```

Figure 41: Received FP1101.TXT file.

In SDK, Refresh the **edkdsp_cc/a** directory (by F5) to see the received **FP1101.TXT** file downloaded from the server running on the Artix7 FPGA. Notice that the input data are printed as **"I=00"**. Select the directory where you want to get the **FP1101.TXT** file.

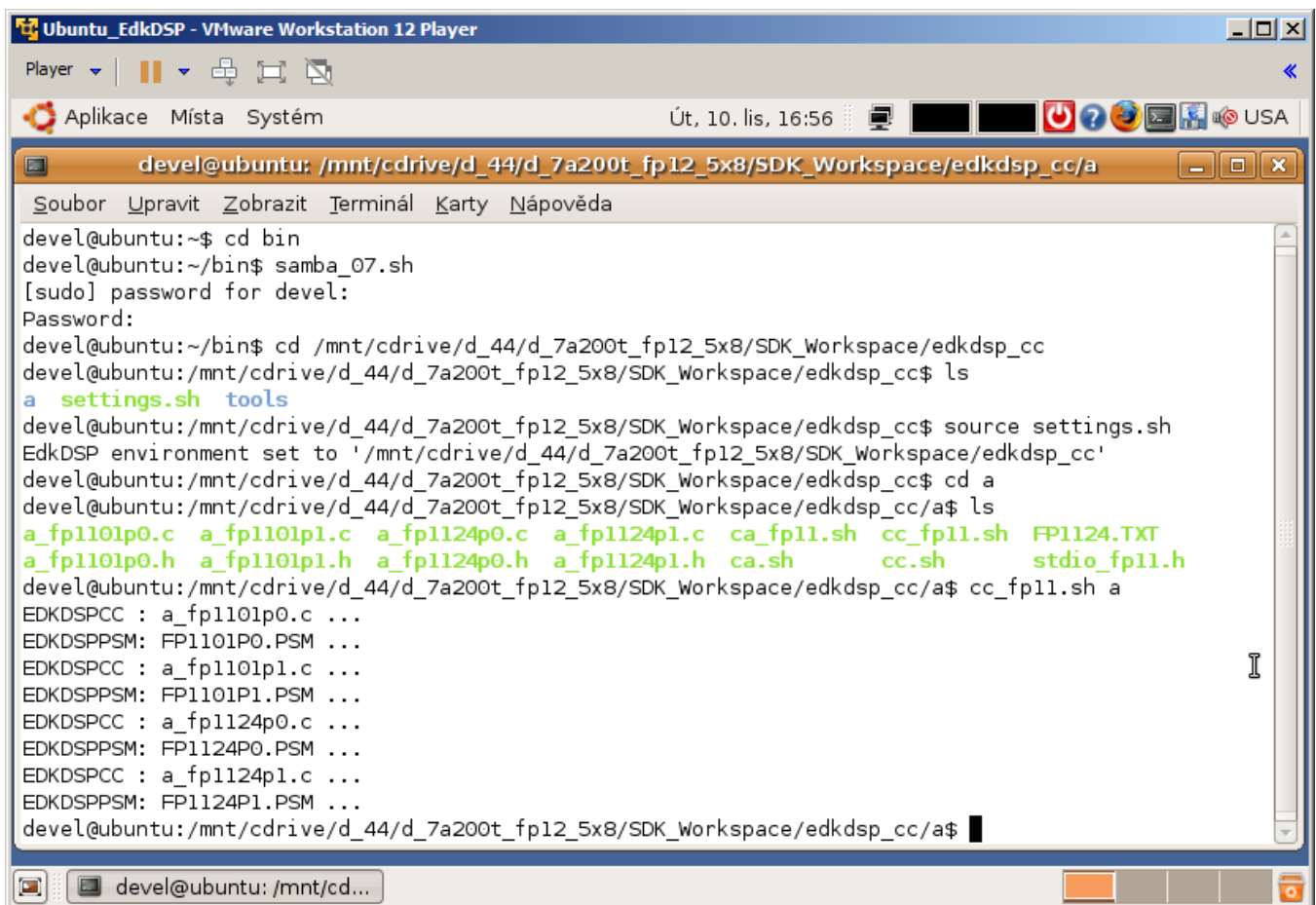


Figure 42: Compile the C code with uncommented lines to display Input=00 instead of I=00

Refresh the project explorer view by F5 in SDK. The uploaded log file **FP1101.TXT** can be open. See Figure 42. The PicoBlaze6 original firmware is writing **"I=00"** to the log file as expected.

cc

- Keep the application running on the Artix7 together with the browser GUI open.
- Compile the modified firmware source code by script **cc_fp11.sh** with parameter **a**.

\$ cc_fp11.sh a

Four C firmware programs will be compiled to header files with the firmware binary code. See Figure 43:

- a_fp1101p0.c is compiled to FP1101P0.DEC
- a_fp1101p1.c is compiled to FP1101P1.DEC
- a_fp1124p0.c is compiled to FP1124P0.DEC
- a_fp1124p1.c is compiled to FP1124P0.DEC

This compiled firmware can be uploaded from PC to the running demo application in the Artix7 chip.

Upload the compiled firmware from PC to the Artix7 File system. See Figure 44 - Figure 47.

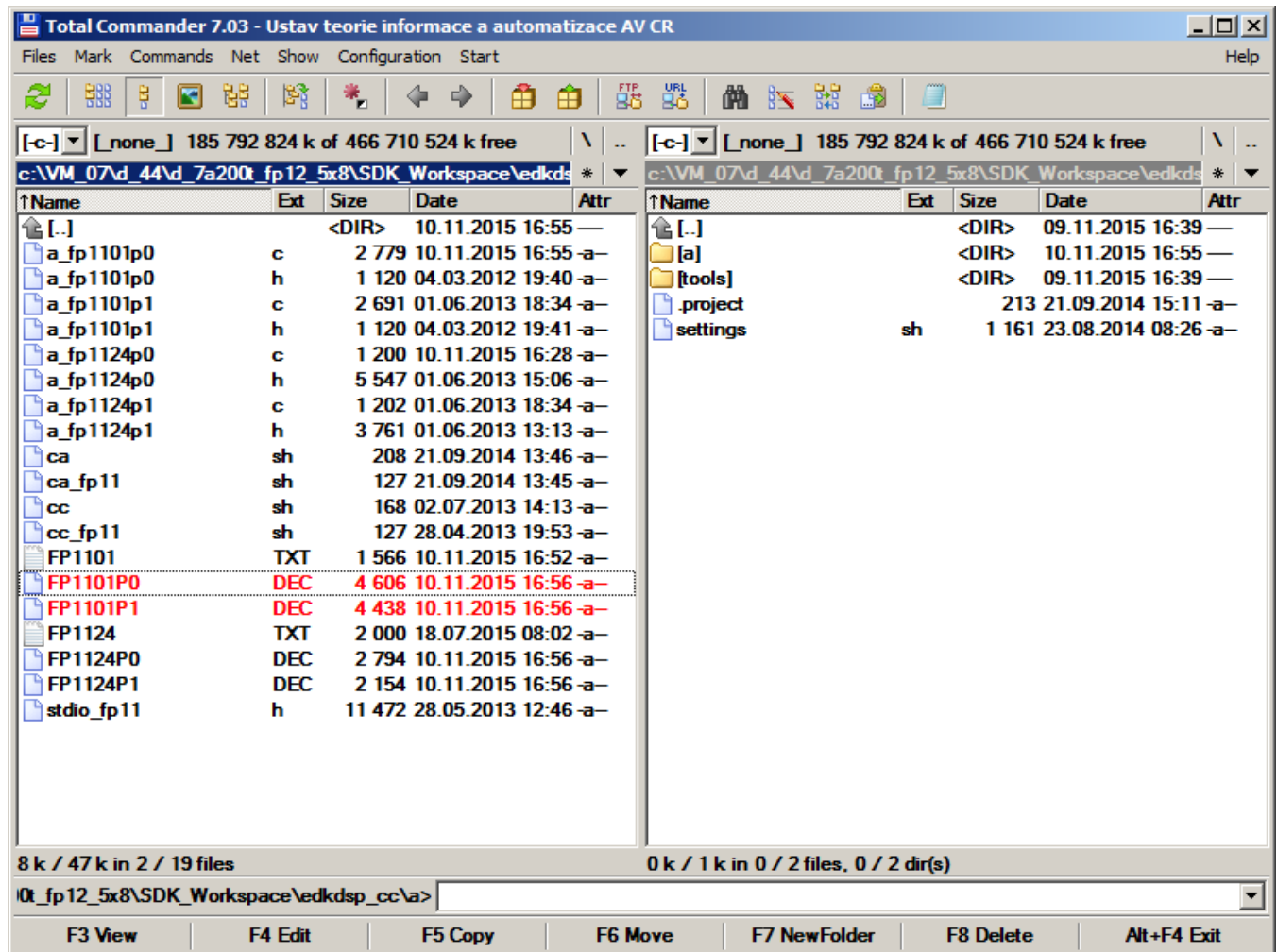


Figure 43: Select compiled binaries in Total Commander or in another file explorer

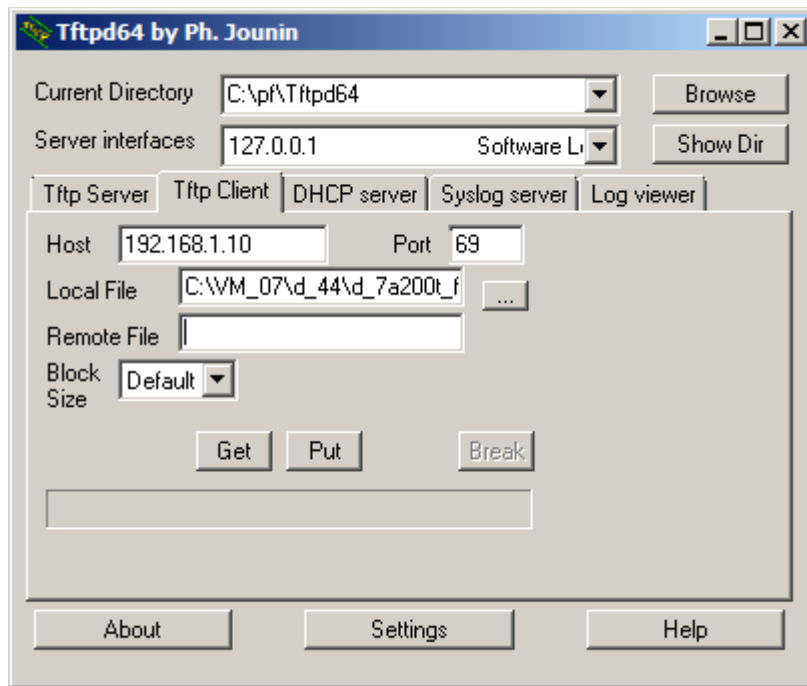


Figure 44: Drag and drop the firmware files to the TFTP client for the download to the Artix7 FPGA

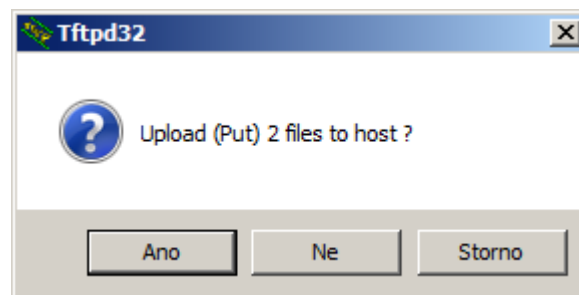


Figure 45: Confirm Ano (yes in Czech...)

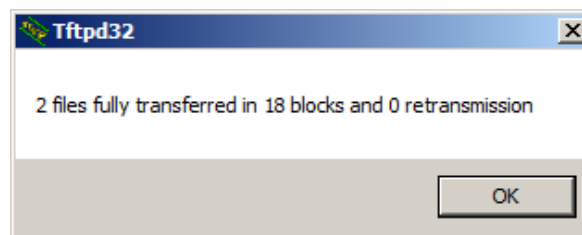


Figure 46: The TFTP server is indicating number of files and blocks transferred to Artix7 file system. Confirm OK

The transfer of each new firmware file is also reported in the console. See Figure 47.

In the www browser user interface, start new test of the EdkDSP accelerator IP by clicking on the "Toggle LEDs" button. See Figure 47. The new firmware files have been found, and the firmware of the tested EdkDSP accelerator IP core has been updated. Tests have been performed and the log file **FP1100.TXT** is stored in the Artix7 RAM based file system.


```

COM3 - PuTTY
Directory css 00000003
Directory images 00000005
Directory js 00000003
Directory yui 00000007
index.html 00000c0b
FP1101.TXT 0000061e
http POST: ledstatus: FFFFFFFF
TFTP RRQ (read request): FP1101.TXT
TFTP WRQ (write request): FP1101P0.DEC
TFTP WRQ (write request): FP1101P1.DEC

Tests of vector operations.
Updating firmware FP1101P0.DEC
Updating firmware FP1101P1.DEC
File FP1101.TXT created for wr
MB0 : (EdkDSP 8xSIMD) Capabilities1 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities2 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities3 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities4 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities5 = 13FFFF

ah=0 bh=0 zh=0
MB0 : (EdkDSP 8xSIMD) VZ2A 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VB2A 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VZ2B 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VA2B 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VADD 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VADD_BZ2A 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VADD_AZ2B 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VSUB 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VSUB_BZ2A 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VSUB_AZ2B 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VMULT 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMULT_BZ2A 'worker1' . OK
MB0 : (EdkDSP 8xSIMD) VMULT_AZ2B 'worker1' . OK
MB0 : (EdkDSP 8xSIMD) VPROD 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMAC 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMSUBAC 'worker1' .... OK
MB0 : (EdkDSP 8xSIMD) VPROD_S8 'worker1' ... OK
MB0 : (EdkDSP 8xSIMD) VDIV 'worker1' ..... OK
Blocks_used 238
Blocks_free 62
Directory css 00000003
Directory images 00000005
Directory js 00000003
Directory yui 00000007
index.html 00000c0b
FP1101.TXT 00000666
http POST: ledstatus: 0

```

Figure 47: The TFTP server is indicating number of blocks uploaded to Artix7 file system

Download the **FP1101.TXT** file to PC with the TFTP client application, and see its content in the SDK. The messages from the tested EdkDSP accelerator have been modified to **"Input=00"**. See Figure 50.

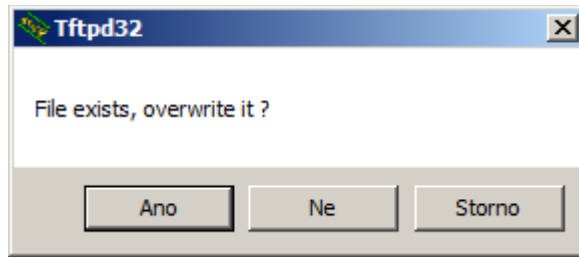


Figure 48: Confirm Ano (yes in Czech...)

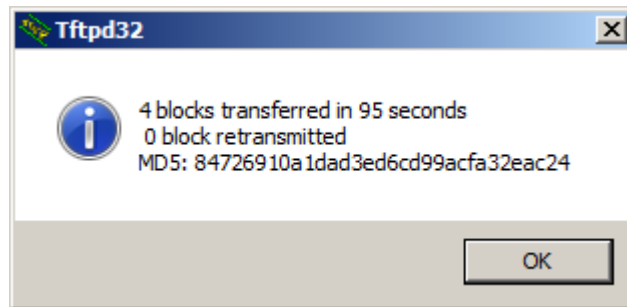


Figure 49: FP1101.TXT received from the Artix7 FPGA to PC via Ethernet. Confirm OK

```
Screen1  
L=00  
Screen2  
L=00  
Screen3  
Input=00  
Screen4  
ah=00 bh=00 zh=00  
...
```

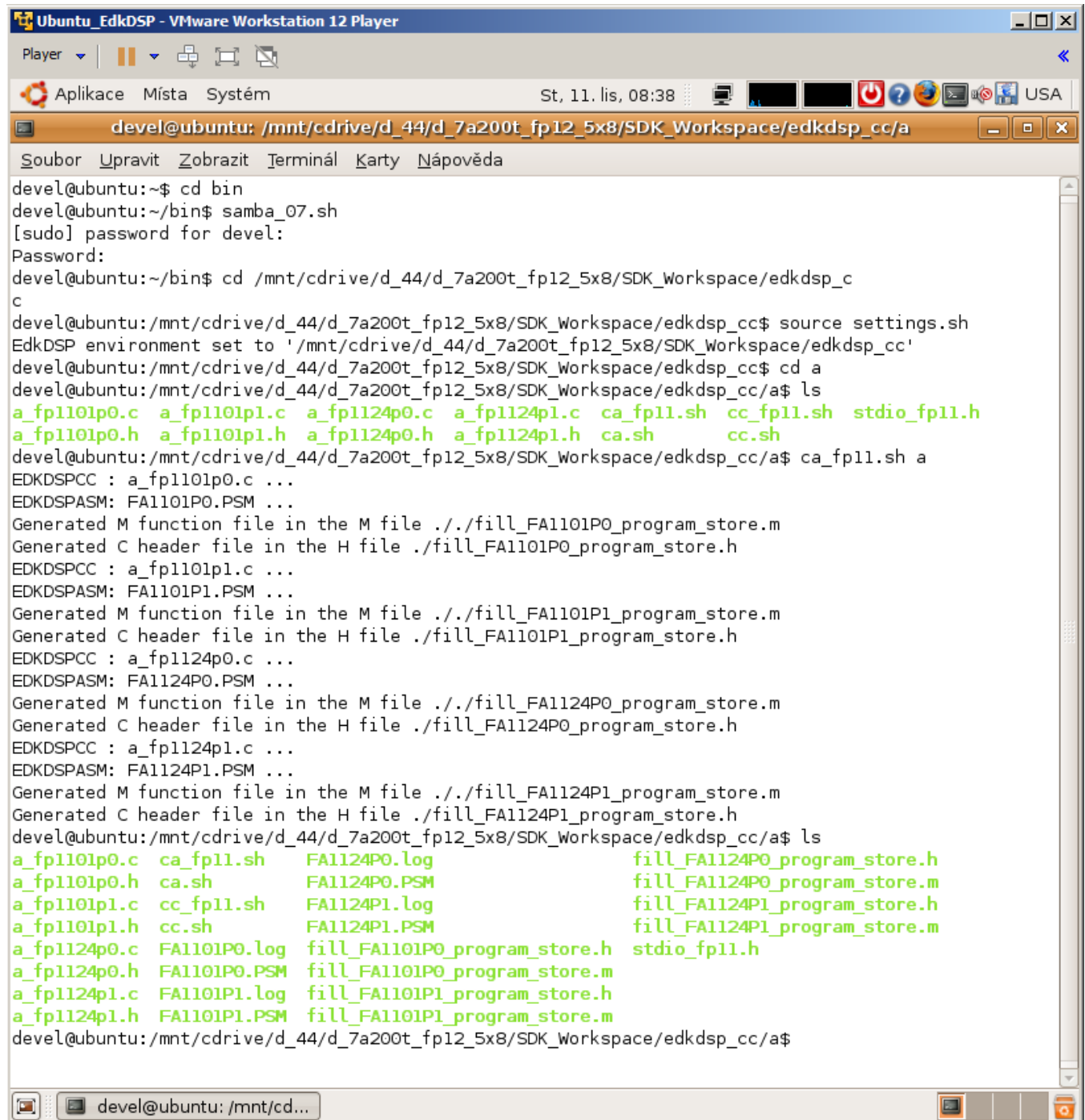
Figure 50: The console output indicates that 2 firmware files have been found.

Close the browser application. Stop the application on the Artix7 MicroBlaze processor from the SDK.

We have demonstrated the upload of re-compiled firmware from PC to the Artix7 and to the EdkDSP accelerator IP core and download of resulting log file from the Artix7 file system to the PC via Ethernet.

3.7 Use of the C compiler for the EdkDSP firmware without Ethernet

This section is describing the use of the UTIA EdkDSP C compiler to recompile the firmware for the PicoBlaze6 controller present in each of the 5 (8xSIMD) EdkDSP accelerators in the AC701 board for simple application without internet connectivity. The **edkdsp** project in the SDK project explorer will be used as an example.



```
Ubuntu_EdkDSP - VMware Workstation 12 Player
Player
Aplikace Místa Systém St, 11. lis, 08:38
devel@ubuntu: /mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/edkdsp_cc/a
Soubor Upravit Zobrazit Terminál Karty Nápověda
devel@ubuntu:~$ cd bin
devel@ubuntu:~/bin$ samba_07.sh
[sudo] password for devel:
Password:
devel@ubuntu:~/bin$ cd /mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/edkdsp_c
c
devel@ubuntu:/mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/edkdsp_cc$ source settings.sh
EdkDSP environment set to '/mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/edkdsp_cc'
devel@ubuntu:/mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/edkdsp_cc$ cd a
devel@ubuntu:/mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/edkdsp_cc/a$ ls
a_fp1101p0.c a_fp1101p1.c a_fp1124p0.c a_fp1124p1.c ca_fp11.sh cc_fp11.sh stdio_fp11.h
a_fp1101p0.h a_fp1101p1.h a_fp1124p0.h a_fp1124p1.h ca.sh cc.sh
devel@ubuntu:/mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/edkdsp_cc/a$ ca_fp11.sh a
EDKDSPCC : a_fp1101p0.c ...
EDKDSPASM: FA1101P0.PSM ...
Generated M function file in the M file ./fill_FA1101P0_program_store.m
Generated C header file in the H file ./fill_FA1101P0_program_store.h
EDKDSPCC : a_fp1101p1.c ...
EDKDSPASM: FA1101P1.PSM ...
Generated M function file in the M file ./fill_FA1101P1_program_store.m
Generated C header file in the H file ./fill_FA1101P1_program_store.h
EDKDSPCC : a_fp1124p0.c ...
EDKDSPASM: FA1124P0.PSM ...
Generated M function file in the M file ./fill_FA1124P0_program_store.m
Generated C header file in the H file ./fill_FA1124P0_program_store.h
EDKDSPCC : a_fp1124p1.c ...
EDKDSPASM: FA1124P1.PSM ...
Generated M function file in the M file ./fill_FA1124P1_program_store.m
Generated C header file in the H file ./fill_FA1124P1_program_store.h
devel@ubuntu:/mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/edkdsp_cc/a$ ls
a_fp1101p0.c ca_fp11.sh FA1124P0.log fill_FA1124P0_program_store.h
a_fp1101p0.h ca.sh FA1124P0.PSM fill_FA1124P0_program_store.m
a_fp1101p1.c cc_fp11.sh FA1124P1.log fill_FA1124P1_program_store.h
a_fp1101p1.h cc.sh FA1124P1.PSM fill_FA1124P1_program_store.m
a_fp1124p0.c FA1101P0.log fill_FA1101P0_program_store.h stdio_fp11.h
a_fp1124p0.h FA1101P0.PSM fill_FA1101P0_program_store.m
a_fp1124p1.c FA1101P1.log fill_FA1101P1_program_store.h
a_fp1124p1.h FA1101P1.PSM fill_FA1101P1_program_store.m
devel@ubuntu:/mnt/cdrive/d_44/d_7a200t_fp12_5x8/SDK_Workspace/edkdsp_cc/a$
```

Figure 51: Compile C source code for the accelerator by the EDKDSPCC compiler

The firmware C source code examples can be compiled by the script **ca_fp11.sh** with parameter **a**. Type in the Ubuntu terminal (See Figure 51):

```
$ ca_fp11.sh a
```

Compile C source code. It will create the assembler source code and firmware binary in format of C .h header files. These headers can be used for inclusion into the EdkDSP demo project (without the TFTP file server). This will compile and assemble all four C firmware programs to header files with the firmware binary code:

a_fp1101p0.c is compiled to **fill_FA1101P0_program_store.h**

a_fp1101p1.c is compiled to **fill_FA1101P1_program_store.h**

a_fp1124p0.c is compiled to **fill_FA1124P0_program_store.h**

a_fp1124p1.c is compiled to **fill_FA1124P1_program_store.h**

Copy and paste the compiled headers into the src directory of the MicroBlaze project **edkdsp** of the SDK 2014.4. See Figure 52 - Figure 54.

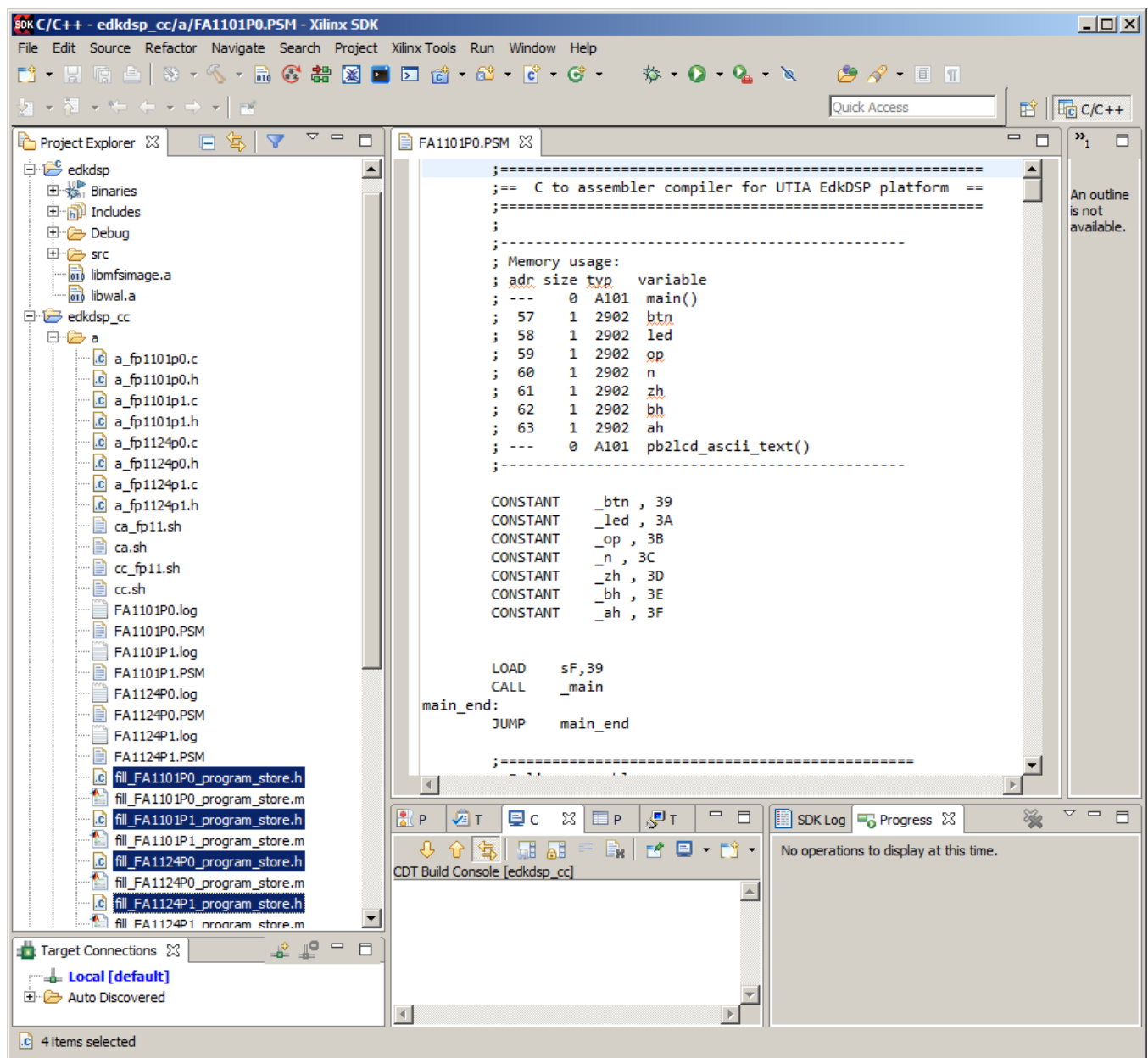


Figure 52: Select firmware header files and Ctrl-C Ctrl-V them to the edkdsp/src directory

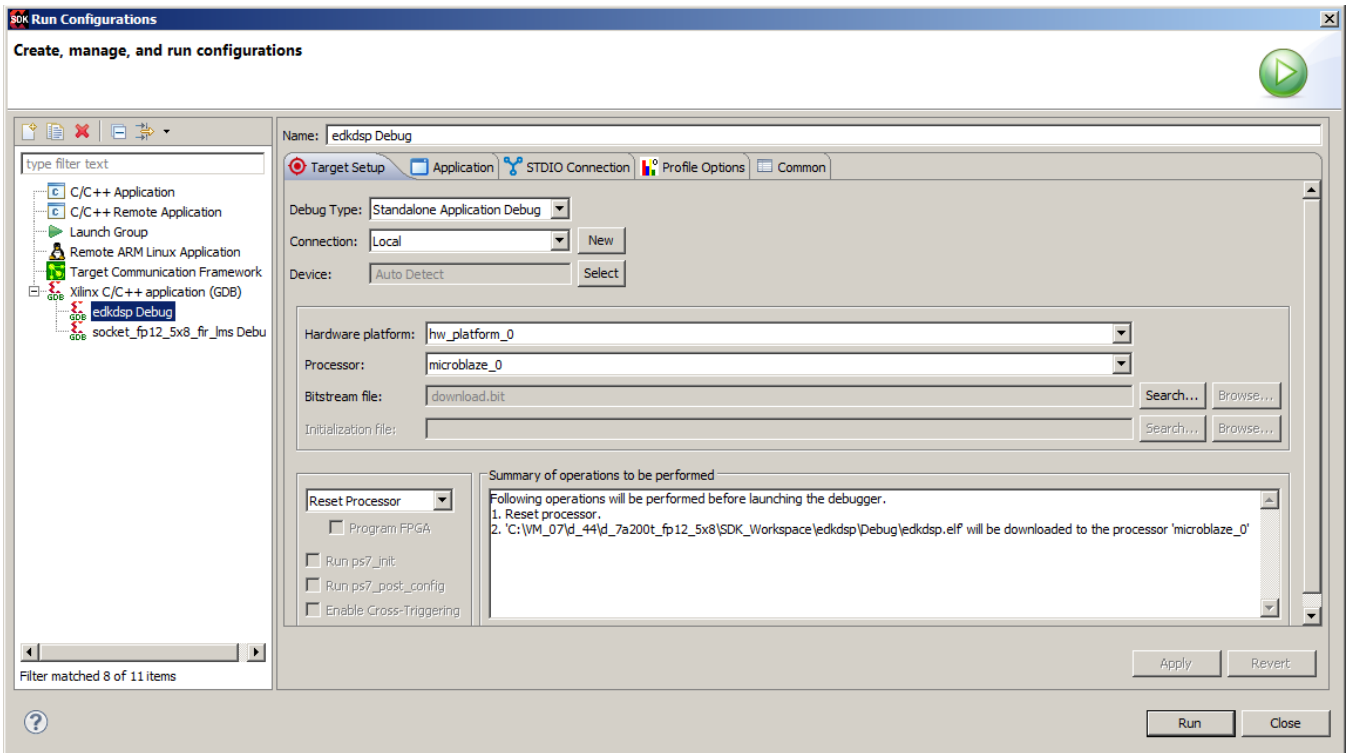


Figure 53: Confirm to overwrite multiple files

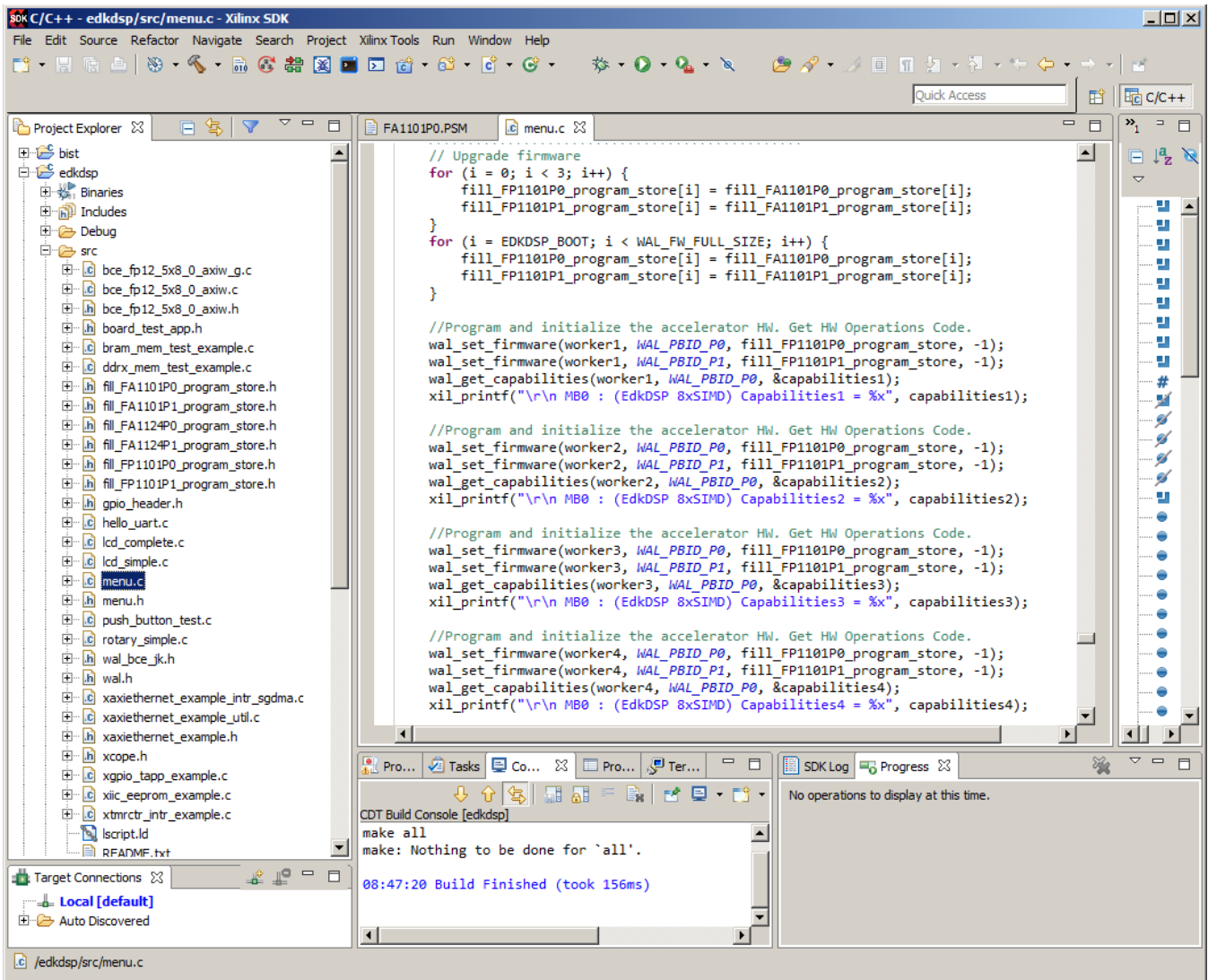


Figure 54: See the updated edkdsp/src directory and section of the Microblaze source code, where the recompiled modified firmware is updated and EdkDSP accelerators are programmed

Notice also the listing of the firmware in the assembler in Figure 52.

Figure 54 is presenting the firmware update section of the C code in the Microblaze edkdsp project.

In SDK, recompile the edkdsp project, to reflect the change of the firmware in header files.

To test the new firmware, take these steps:

- Recompile **edkdsp.elf** application (clear the current project, to take into account the new, changed firmware in MicroBlaze header files)
- Download again the bitstream from Vivado 2014.4 Hardware Manager.

See Figure 55.

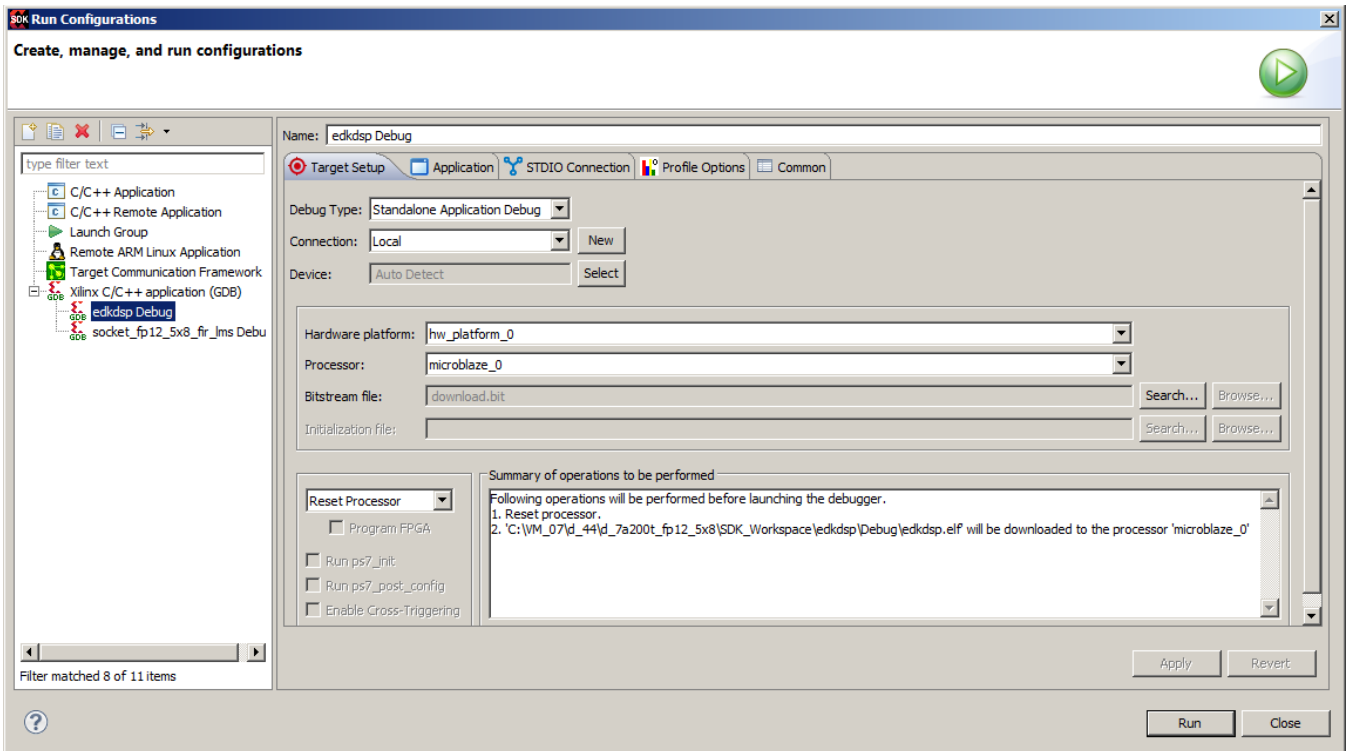


Figure 55: Recompile edkdsp project, download the .bit file and run the edkdsp.elf on Artix

Figure 56 is presenting the initial menu of the **edkdsp** application.

- Type C to select test of the EdkDSP operation
- See results of the test of the EdkDSP accelerator with modified firmware.
- Type 0 to exit from the **edkdsp** simple menu.
- Close the debug session from SDK console.

```
COM3 - PuTTY
*****
*****
**      Xilinx Artix-7 FPGA AC701 Evaluation Kit      **
*****
*****
Choose Feature to Test:
1: UART Test
2: LED Test
3: IIC Test
4: TIMER Test
5: ROTARY Test
6: SWITCH Test
7: LCD Test
8: DDR3 External Memory Test
9: BRAM Internal Memory Test
A: ETHERNET Loopback Test
B: BUTTON Test
C: EdkDSP Eval Op
0: Exit
C
-- Entering main() --

Tests of vector operations.
MB0 : (EdkDSP 8xSIMD) Capabilities1 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities2 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities3 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities4 = 13FFFF
MB0 : (EdkDSP 8xSIMD) Capabilities5 = 13FFFF

ah=0  bh=0  zh=0
MB0 : (EdkDSP 8xSIMD) VZ2A 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VB2A 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VZ2B 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VA2B 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VADD 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VADD_BZ2A 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VADD_AZ2B 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VSUB 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VSUB_BZ2A 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VSUB_AZ2B 'worker1' .. OK
MB0 : (EdkDSP 8xSIMD) VMULT 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMULT_BZ2A 'worker1' . OK
MB0 : (EdkDSP 8xSIMD) VMULT_AZ2B 'worker1' . OK
MB0 : (EdkDSP 8xSIMD) VPROD 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMAC 'worker1' ..... OK
MB0 : (EdkDSP 8xSIMD) VMSUBAC 'worker1' .... OK
MB0 : (EdkDSP 8xSIMD) VPROD_S8 'worker1' ... OK
MB0 : (EdkDSP 8xSIMD) VDIV 'worker1' ..... OK
Press any key to return to main menu
```

Figure 56: Test the EdkDSP accelerator with the new firmware from the menu (type C)

3.8 Use of ILA for debug of the EdkDSP accelerator IP

The Vivado 2014.4 Hardware Manager ILA trigger can be triggered from the EdkDSP firmware running on the PicoBlaze6 firmware inside of the (8xSIMD) EdkDSP unit. This firmware can be modified and recompiled in the SDK 2013.4.

In SDK, open the `edkdsp_cc/a/a_fp1124p1.c` file. See section of the modified C code FIR firmware. Code includes the additional call to the `pb2dfu_set()` function. It will be used for triggering of the ILA in this specified point of computation of the EdkDSP accelerator.

```
...
pb2dfu_set(0x20, 1); // To provide the trigger (0x01 on port 0x20) for the ILA
for (i = 0; i < 4; i++) {
    for (j = 2; j <= 3; j++) {
        fir(j, n, op);
        pb2mb_eoc(led);
    }
}
...
```

In Vivado, in the ILA configuration page, change the trigger condition to:

`(bce_port_wr ==1) AND (bce_port_id[0:7]==0x20) AND (bce_port[0:7]==0x01)`

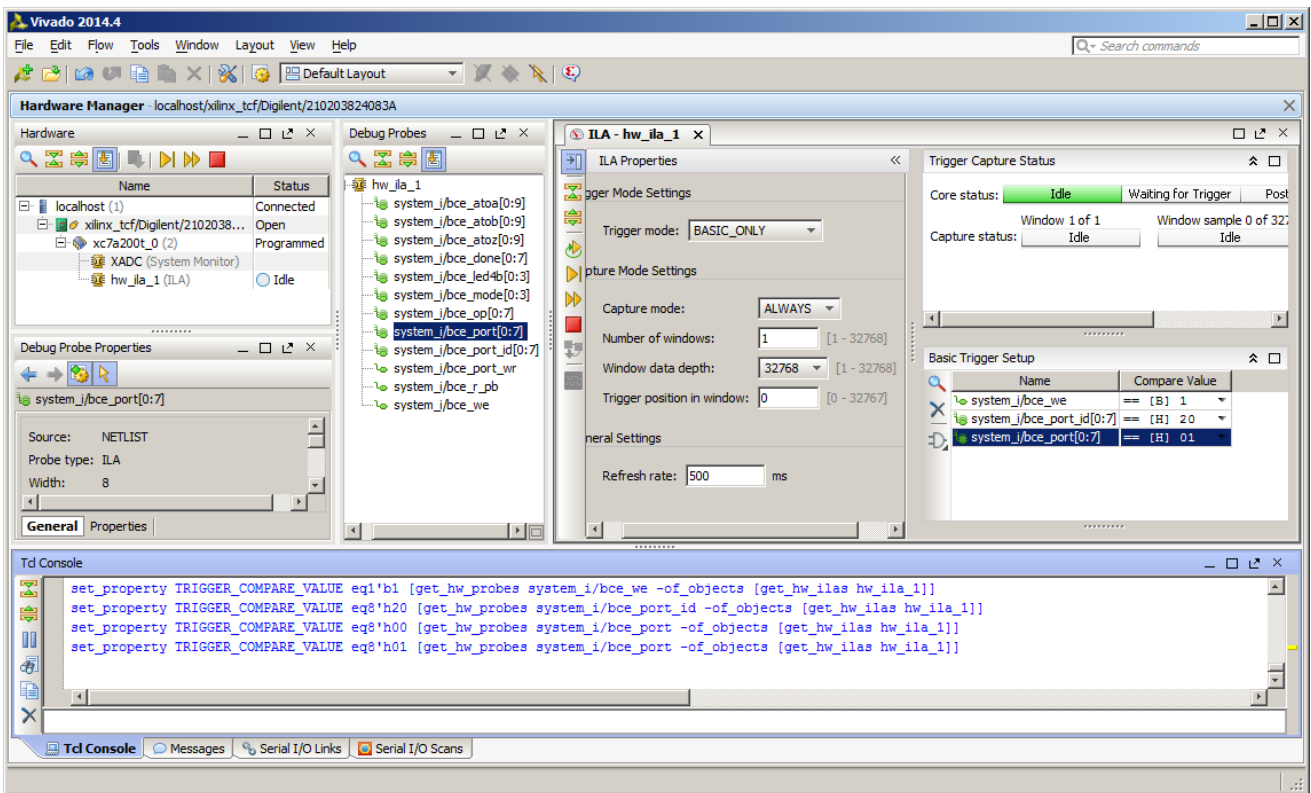


Figure 57: Vivado 2013.4 ILA display of the FIR filter computation

In Vivado HW Manager, arm the ILA by pressing **Run Trigger** button in **Hardware** window. See Figure 57. ILA will wait for this function call and it will trigger 32K samples of all debug signals with the sampling rate 125 MHz and provide the detailed trace of the initial 32k samples of the FIR filter computation.

The red trigger is corresponding to the event. We can zoom in the data and define additional markers. Selected markers indicate single elementary step of the FIR filter. It takes 308 clock cycles (125 MHz = 8ns clock period) to compute the vector product of two floating point vectors (coefficients and data), both with length 250*8=2000 elements and to update the data vector (circular buffer). The ILA provides sufficient level of visibility and debug capabilities for the developer of the (8xSIMD) EdkDSP IP and its firmware. See Figure 58.

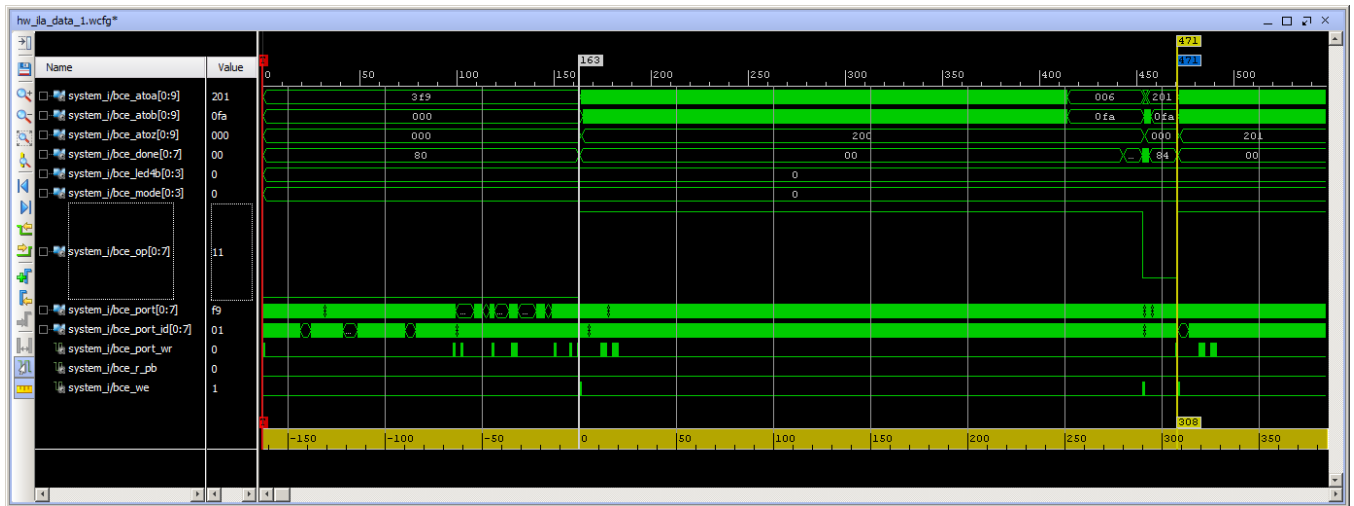


Figure 58: Vivado 2014.4 HW Manager ILA display of the FIR filter computation

In SDK, see also `edkdsp_cc/a/a_fp1124p0.c` code implementing the LMS filter on the identical EdkDSP HW.

```

...
pb2dfu_set(0x20, 0); // To provide dedicated trigger (0x00 on port 0x20) for the ILA
for (i = 0; i < 4; i++) {
    for (j = 2; j <= 3; j++) {
        lms(j, n, op);
        pb2mb_eoc(led);
    }
}
...

```

In Vivado hardware manager select new trigger condition for the ILA:

```
(bce_port_wr == 1) AND (bce_port_id[0:7] == 0x20) AND (bce_port[0:7] == 0x00)
```

In Vivado, arm the ILA by pressing **Run Trigger** button in **Hardware** window. This will trigger 32K samples of all debug signals with the sampling rate 125 MHz and provide detailed trace of the initial 32k samples of the LMS filter computation. See Figure 44.

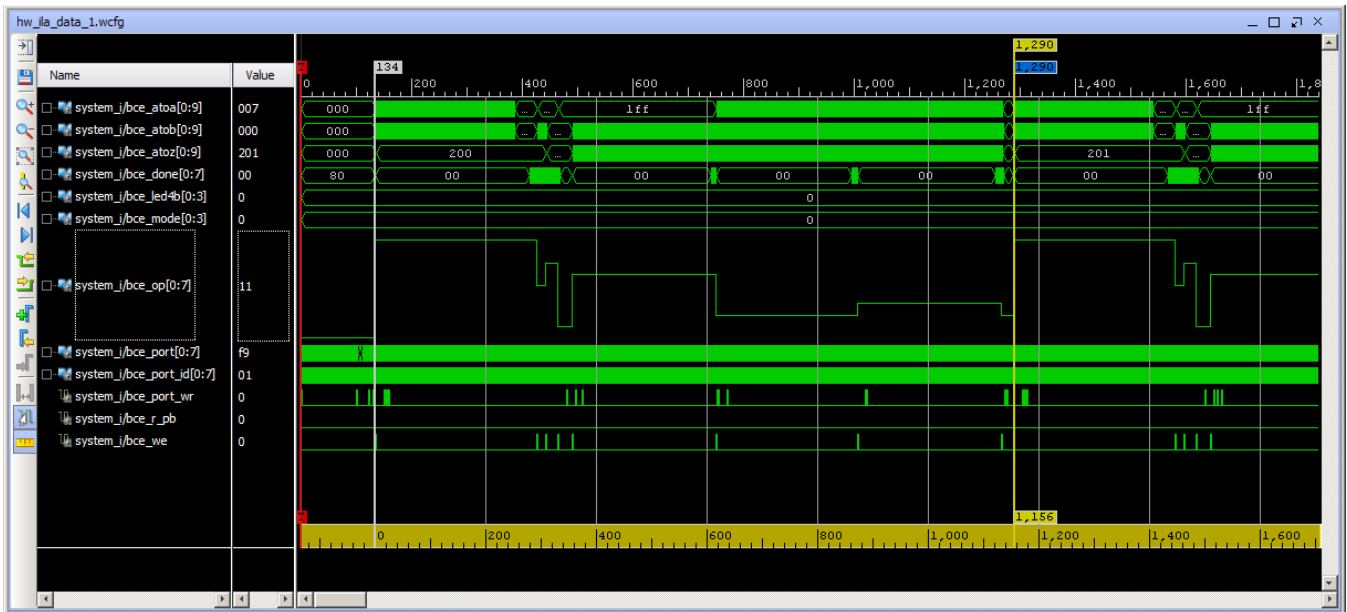


Figure 59: Vivado 2014.4 HW Manager ILA display of adaptive LMS filter computation

The red trigger is corresponding to the event. We can zoom in the data and define additional markers. Selected markers to indicate single elementary step of the LMS filter. It takes 1156 clock cycles (125 MHz = 8ns clock period) to compute the vector product of two floating point vectors (coefficients and data), both with length $250 \times 8 = 2000$ elements, update the data vector (circular buffer), compute the prediction error and adapt the coefficients of the LMS filter. See Figure 59.

The bce_op[0:7] debug signal is displayed in the analogue/hold mode and indicate the sequence of vector operations issued by the PicoBlaze6 firmware, while implementing the LMS single step on the (8xSIMD) EdkDSP vector unit IP.

3.9 Display of Temperature, Debug and Verification

Temperature of the Artix chip can be measured and displayed by the Vivado HW manager support in a separate dashboard. See Figure 60.

MicroBlaze code can be compiled with `-O0`, ... , `-O3` optimisations and executed in debugger in combination with ILA triggering. The `-O0` option provides lower performance MicroBlaze, but the corresponding code includes no transformations. This makes the debugging of C code easier. This helps in debugging of the interactions of MicroBlaze with the EdkDSP accelerator IP. Complete blocks of floating point data can be inspected and verified in the MicroBlaze debugger.

The EdkDSP accelerator code is deterministic and all operations can be emulated in the MicroBlaze C code, including the exact sequence of all floating point operations. The floating point unit cores of the MicroBlaze for the ADD and MULT provide bit-exact identical results to the floating point units used in the (8xSIMD) EdkDSP vector unit.

This determinism secures, that the MicroBlaze code provides bit-exact identical results to the (8xSIMD) EdkDSP IP core. This is used for verification of algorithms.

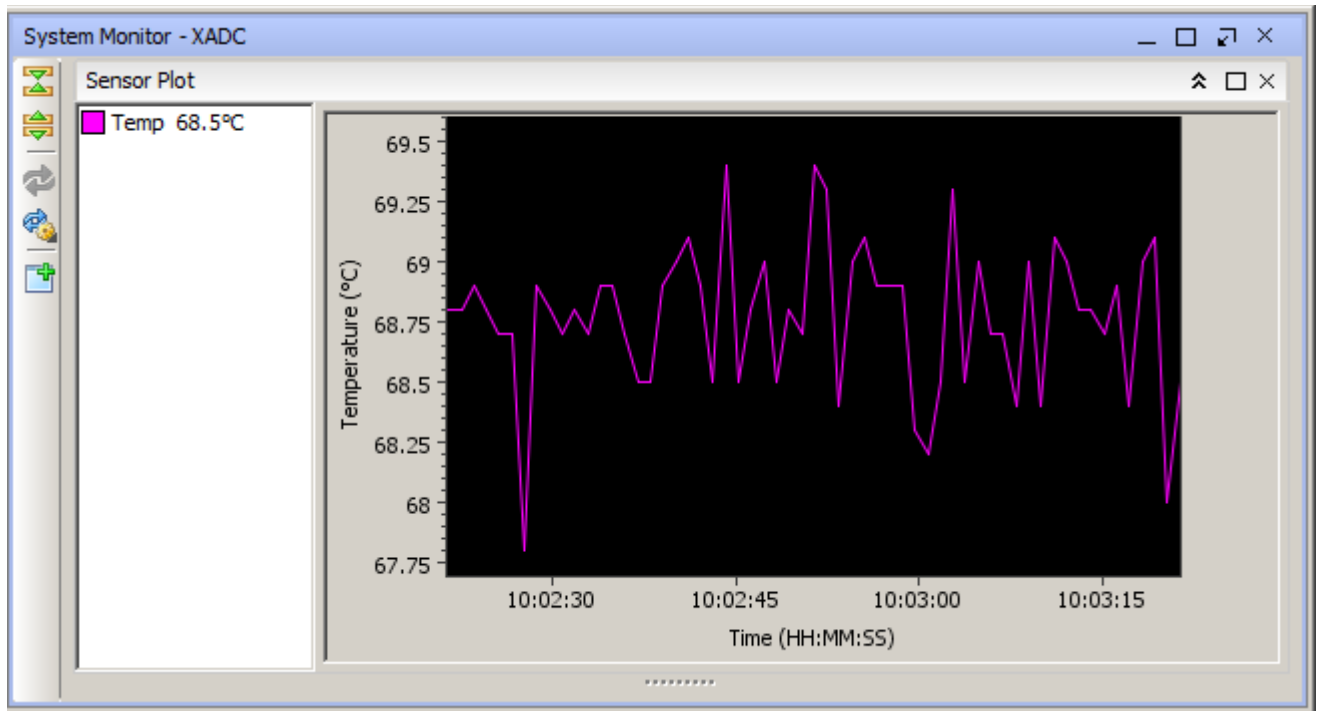


Figure 60: Dashboard display: Temperature of the chip. No cooling at all.

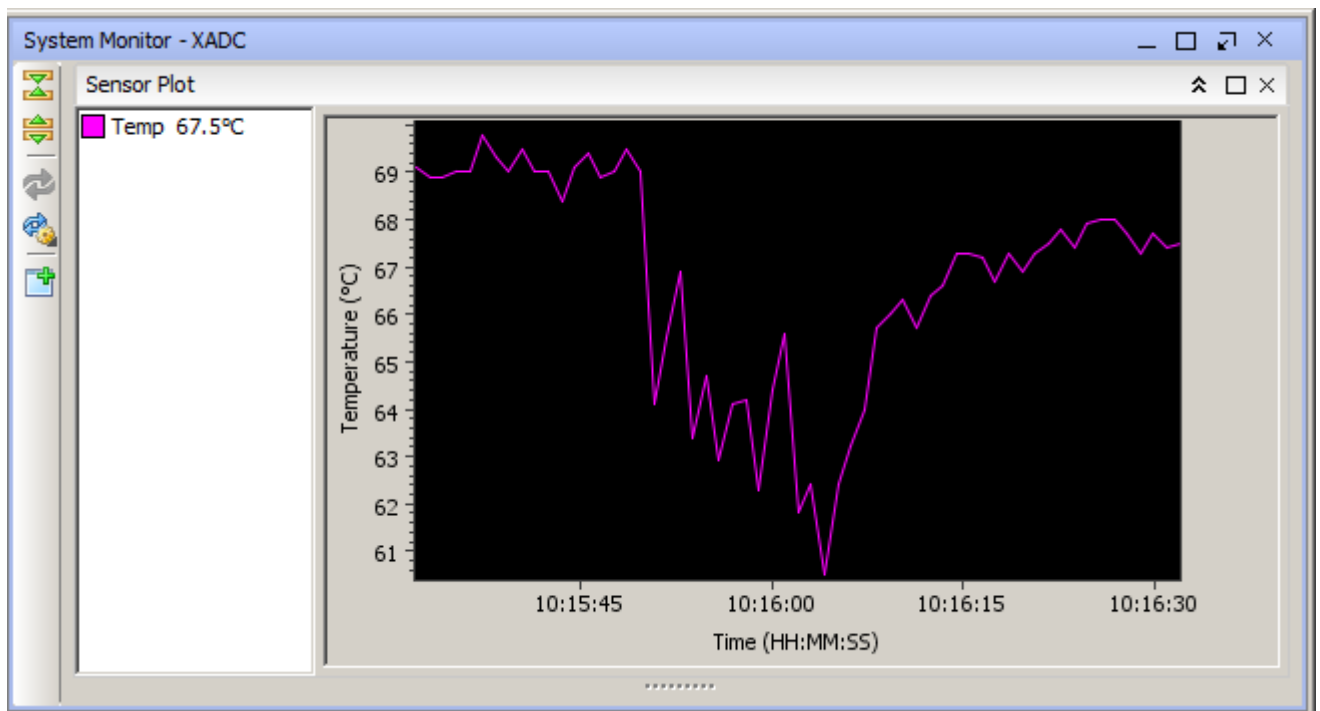


Figure 61: Dashboard display: Temperature of the chip. Active cooling OFF-ON-OFF

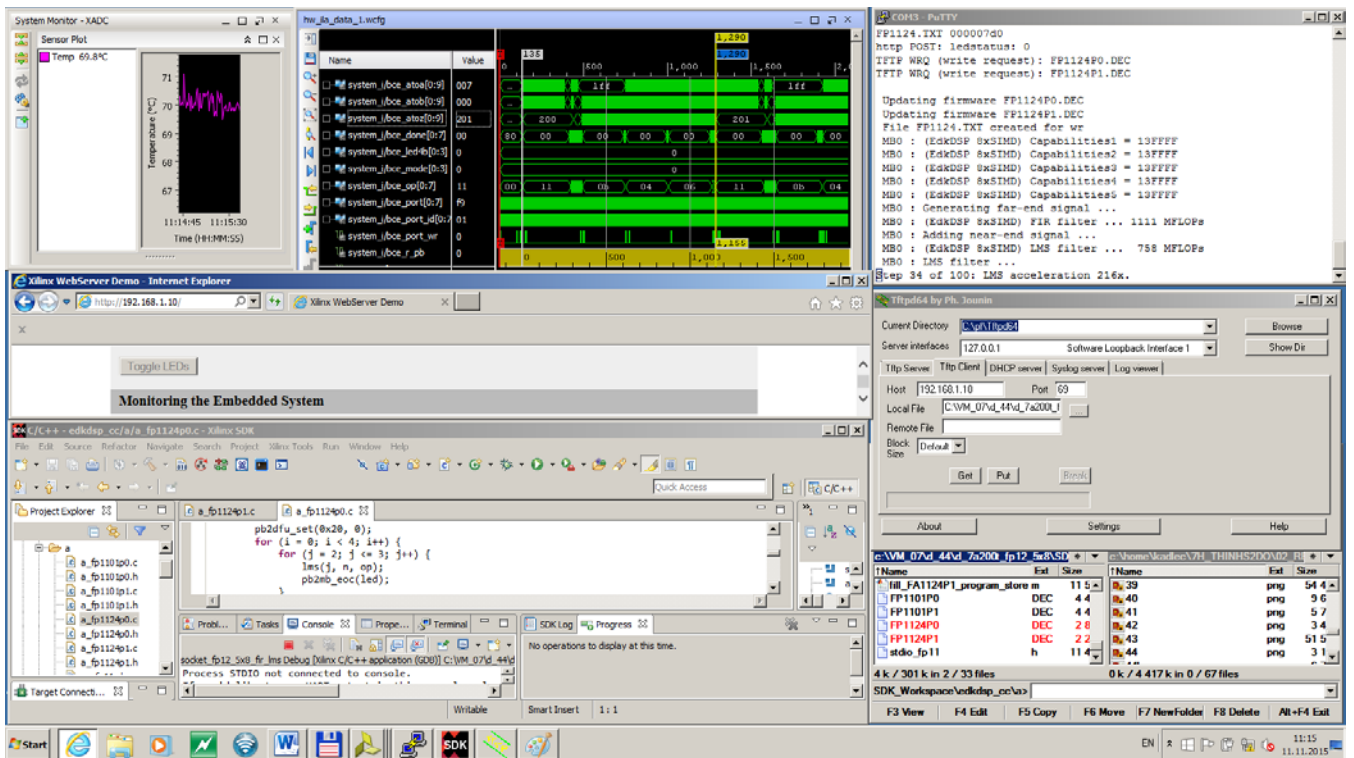


Figure 62: PC screen snapshot. Demonstrates debug of EdkDSP IP core with ILA

Figure 62 is presenting the PC debug screen. See also Figure 1.
This is brief list of applications open on the screen:

Top

- Vivado 2014.4 HW Manager chip temperature measurement
- Vivado 2014.4 HW Manager ILA probes detecting the LMS filter computation
- Board terminal

Middle

- Microsoft Internet Explorer (11.0) GUI served by the demonstrator
- TFTP client used for transfer of recompiled EdkDSP accelerator IP firmware to the board

Bottom

- SDK 2014.4 with open C source code of the EdkDSP IP firmware
- Total Commander for managing of files on PC

4. References

- [1] AC701 Evaluation Board, for the Artix-7 FPGA User Guide UG952 (v1.3) April 7, 2015 UG952 http://www.xilinx.com/support/documentation/boards_and_kits/ac701/ug952-ac701-a7-eval-bd.pdf
- [2] AC701 Built - In Self Test, Flash Application, ver 10.0, 11th December 2014 http://www.xilinx.com/support/documentation/boards_and_kits/ac701/2014_4/xtp194-ac701-bist-c-2014-4.pdf
- [3] LightWeight IP (lwIP) Application Examples, Author: Anirudha Sarangi and Stephen MacMahon; XAPP1026 (v3.2); October 28, 2012.
- [4] LightWeight IP Application Examples, Author: Anirudha Sarangi, Stephen MacMahon, and Upender Cherukupaly; XAPP1026 (v5.1); November 21, 2014. http://www.xilinx.com/support/documentation/application_notes/xapp1026.pdf
- [5] PicoBlaze 8-bit Embedded Microcontroller User Guide for Extended Spartan 3 and Virtex5 FPGAs; Introducing PicoBlaze for Spartan-6, Virtex-6, and 7 Series FPGAs. UG129 June 22, 2011. http://www.xilinx.com/support/documentation/ip_documentation/ug129.pdf
- [6] Eniac JU project THINGS2DO “Thin but Great Silicon 2 Design Objects”, project number ENIAC JU 621221. <http://things2do.space.com.ro/>
<http://sp.utia.cz/index.php?ids=projects/things2do>

5. Evaluation version of Vivado 2014.4 Artix7 designs

The enclosed **Evaluation version of precompiled Vivado 2014.4 Artix7 designs with evaluation versions of UTIA (8xSIMD) EdkDSP accelerator cores** can be downloaded from UTIA [www](http://www.utia.cas.cz) pages free of charge and used for evaluation together with the five UTIA (8xSIMD) EdkDSP accelerators.

The evaluation package includes one DVD or the www download package with these deliverables:

8 precompiled designs with UTIA (8xSIMD) EdkDSP accelerators for Xilinx Artix7 AC701 board [1], [2] compiled in Xilinx Vivado 2014.4. The UTIA (8xSIMD) EdkDSP accelerators are compiled with HW limit on number of vector operations. The termination of the nonexclusive, non-transferable evaluation license is reported in advance by the demonstrator on the terminal.

The evaluation package includes SDK 2014.4 SW projects with C source code for MicroBlaze processor. SW projects support the family of UTIA (8xSIMD) EdkDSP accelerators for the Xilinx AC701 board [1], [2].

The evaluation package includes this compiled library:

libwal.a EdkDSP api (SDK 2014.4, MicroBlaze) for EdkDSP accelerators on AC701 board.
libmfsimage.a The library with file system supporting simple www server GUI.

The library **libwal.a** has no time restriction. The nonexclusive, non-transferable evaluation license is provided by UTIA only for the use with the family of UTIA EdkDSP accelerators designed for the Xilinx AC701 board. Source code of this library is owned by UTIA and it is not provided in this evaluation package.

The evaluation package includes these binary applications for Ubuntu:

edkdsppp EdkDSP C pre-processor binary for Ubuntu (x86 PC) under the VMware Player.
edkdspcc EdkDSP C compiler binary for Ubuntu (x86 PC) under the VMware Player.
edkdspasm EdkDSP ASM compiler binary for Ubuntu (x86 PC) under the VMware Player.
edkdsppsm EdkDSP ASM compiler binary for Ubuntu (x86 PC) under the VMware Player.

These binary applications have no time restriction. The user of the evaluation package has nonexclusive, non-transferable license from UTIA to use these utilities for compilation of the firmware for the Xilinx PicoBlaze6 processor inside of the UTIA EdkDSP accelerators in the 8 precompiled designs for the Xilinx AC701 board. The source code of these compilers is owned by UTIA and it is not provided in the evaluation package.

The evaluation package includes demonstration firmware in C source code for the Xilinx PicoBlaze6 processor for the family of UTIA EdkDSP accelerators for the Xilinx AC701 board.

The evaluation package also includes compiled versions of this firmware in form of header files .h. These compiled firmware files can be used for initial test of the UTIA EdkDSP accelerators on the Xilinx AC701 board without the need to install the UTIA compiler binaries and the Ubuntu (x86 PC) OS image under VMware Player 3.0.0 (on Win 7, 32bit PC) or under VMware Workstation 12 Player (on Win 7, 64bit PC).

On email request to kadlec@utia.cas.cz, UTIA will send DVD with the Ubuntu (x86 PC) image free of charge.

6. Release version of designs for THINGS2DO project partners

The release version of Vivado 2014.4 Artix7 designs with evaluation versions of UTIA (8xSIMD) EdkDSP accelerator cores for THINGS2DO [6] project partners can be ordered from UTIA AV CR, v.v.i., by email request for quotation to kadlec@utia.cas.cz. UTIA will provide quotation by email. After the confirmed order received by email to kadlec@utia.cas.cz, UTIA AV CR, v.v.i. will deliver (by standard mail to the THINGS2DO project partner) a printed version of this application note together with DVD with deliverables described in this section. UTIA AV CR, v.v.i., will also send to the THINGS2DO project partner (by email) and by the standard mail the invoice for:

Release version of Vivado 2014.4 Artix7 designs with evaluation versions of UTIA (8xSIMD) EdkDSP accelerator cores for THINGS2DO [6] project partners (without VAT)

0,00 Eur

The package includes this application note and the EdkDSP DVD with these deliverables:

8 precompiled designs with UTIA (8xSIMD) EdkDSP accelerators for Xilinx AC701 board, compiled in Xilinx Vivado 2014.4. The UTIA (8xSIMD) EdkDSP accelerators are compiled with HW limit on number of vector operations. The termination of the evaluation license is reported in advance by the demonstrator on the terminal.

The Release version of Vivado 2014.4 Artix7 designs with evaluation versions of UTIA (8xSIMD) EdkDSP accelerator cores for THINGS2DO [6] project partners include all 8 Vivado 2014.4 design projects and the evaluation versions of the UTIA (8xSIMD) EdkDSP accelerators provided in form of netlisted IP cores generated in Xilinx VIVADO 2014.4:

bce_fp12_1x8_0_axiw_v1_10_c
bce_fp12_1x8_0_axiw_v1_20_c
bce_fp12_1x8_0_axiw_v1_30_c
bce_fp12_1x8_0_axiw_v1_40_c

These evaluation versions of UTIA (8xSIMS) EdkDSP netlist pcores are compiled with an HW limit on number of vector operations. THINGS2DO [6] project partners have nonexclusive, non-transferable license from UTIA to integrate these evaluation netlists into their own VIVADO 2014.4 designs and to compile them to unlimited number of bit-streams for designs on Xilinx Artix7 FPGAs. This nonexclusive, non-transferable license has no time restriction. The source code of the evaluation versions of (8xSIMS) EdkDSP accelerators is an IP owned by UTIA and it is not provided in the release package to the THINGS2DO project partners.

The package for the THINGS2DO [6] project partners includes the SDK 2014.4 SW projects in source code for MicroBlaze as described in this application note. Projects support the evaluation versions of the UTIA (8xSIMD) EdkDSP accelerators (in the netlist pcore format) for the Xilinx AC701 board.

The package for the THINGS2DO project partners includes the library:

libwal.a EdkDSP api (SDK 2014.4, MicroBlaze) for EdkDSP accelerators on AC701 board.
libmfsimage.a The library with file system supporting simple www server GUI.

The library **libwal.a** has no time restriction. The nonexclusive, non-transferable evaluation license is provided by UTIA only for the use with the family of UTIA EdkDSP accelerators designed for the Xilinx AC701 board. Source code of this library is owned by UTIA and it is not provided in this evaluation package.

The package for the THINGS2DO project partners includes these binary applications for Ubuntu:

edkdsppp EdkDSP C pre-processor binary for Ubuntu (x86 PC) under the VMware Player.
edkdspcc EdkDSP C compiler binary for Ubuntu (x86 PC) under the VMware Player.
edkdspasm EdkDSP ASM compiler binary for Ubuntu (x86 PC) under the VMware Player.
edkdsppsm EdkDSP ASM compiler binary for Ubuntu (x86 PC) under the VMware Player.

These binary applications have no time restriction. The THINGS2DO project partners have nonexclusive, non-transferable license from UTIA to use these utilities for compilation of the firmware for the Xilinx PicoBlaze6 processor inside of the family of UTIA EdkDSP accelerators for the Xilinx AC701 board. The source code of these compilers is owned by UTIA and it is not provided in the evaluation package.

The package includes demonstration firmware in C source code for the Xilinx PicoBlaze6 processor for the family of UTIA EdkDSP accelerators for the Xilinx AC701 board.

The package also includes compiled versions of this firmware in form of header files .h. These compiled firmware files can be used to evaluate the UTIA EdkDSP accelerators on the Xilinx AC701 board without the need to install the UTIA compiler binaries and the Ubuntu (x86 PC) OS image under VMware Player 3.0.0 (on Win 7, 32bit PC) or under VMware Workstation 12 Player (on Win 7, 64bit PC).

The release package deliverables also includes DVD with the Ubuntu (x86 PC) image (free of charge). This image is provided to ease the installation of the UTIA EdkDSP C compiler under VMware Player 3.0.0 (on Win 7, 32bit PC) or under VMware Workstation 12 Player (on Win 7, 64bit PC).

Any and all legal disputes that may arise from or in connection with the use, intended use of or license for the software provided hereunder shall be exclusively resolved under the regional jurisdiction relevant for UTIA AV CR, v. v. i. and shall be governed by the law of the Czech Republic.

7. Release version of Vivado 2014.4 Artix7 designs

The **release version of Vivado 2014.4 Artix7 designs with the release version of the UTIA (8xSIMD) EdkDSP accelerator cores** can be ordered from UTIA AV CR, v.v.i., by email request for quotation to kadlec@utia.cas.cz. UTIA will provide quotation by email. After the confirmed order received by email to kadlec@utia.cas.cz, UTIA AV CR, v.v.i. will deliver (by standard mail) to the customer the printed version of this application note together with 3 DVDs with deliverables described in this section. UTIA AV CR, v.v.i., will send to the customer (by email) and by the standard mail the invoice for:

Release version of Vivado 2014.4 Artix7 designs with the release version of UTIA (8xSIMD) EdkDSP accelerator cores (without VAT) **400,00 Eur**

The release package includes this application note and the EdkDSP DVD with these deliverables:

8 precompiled designs with UTIA (8xSIMD) EdkDSP accelerators for Xilinx AC701 board [2], compiled in Xilinx Vivado 2014.4. The UTIA (8xSIMD) EdkDSP accelerators included in these designs are compiled with **no HW limit on number of vector operations**. Therefore, all these precompiled designs of the release package run on AC701 without limitations of the evaluation package.

The release package includes all 8 Vivado 2014.4 design projects. The UTIA (8xSIMD) EdkDSP accelerators are provided in the form of netlist IP cores generated in Xilinx VIVADO 2014.4:

```
bce_fp12_1x8_0_axiw_v1_10_c
bce_fp12_1x8_0_axiw_v1_20_c
bce_fp12_1x8_0_axiw_v1_30_c
bce_fp12_1x8_0_axiw_v1_40_c
```

These UTIA (8xSIMS) EdkDSP netlist pcores have **no HW limit on number of vector operations**. The user of the release package has nonexclusive, non-transferable license from UTIA to integrate these netlists into its own VIVADO 2014.4 designs and to compile them to unlimited number of bit-streams. This nonexclusive, non-transferable license has no time restriction. The source code of the (8xSIMS) EdkDSP accelerators is an IP owned by UTIA and it is not provided in the release package to the customer.

The release package includes SDK 2014.4 SW projects in source code for MicroBlaze as described in this application note. Projects support the family of UTIA (8xSIMD) EdkDSP accelerators for Xilinx AC701 board [2].

The release package includes the library:

libwal.a EdkDSP api (SDK 2014.4, MicroBlaze) for EdkDSP accelerators on AC701 board.
libmfsimage.a The library with file system supporting simple www server GUI.

The library **libwal.a** has no time restriction. The nonexclusive, non-transferable evaluation license is provided by UTIA only for the use with the family of UTIA EdkDSP accelerators designed for the Xilinx AC701 board. Source code of this library is owned by UTIA and it is not provided in this release package.

The release package includes these binary applications for Ubuntu:

edkdsppp EdkDSP C pre-processor binary for Ubuntu (x86 PC) under the VMware Player.
edkdspcc EdkDSP C compiler binary for Ubuntu (x86 PC) under the VMware Player.
edkdspasm EdkDSP ASM compiler binary for Ubuntu (x86 PC) under the VMware Player.
edkdsppsm EdkDSP ASM compiler binary for Ubuntu (x86 PC) under the VMware Player.

These binary applications have no time restriction. The user of the evaluation package has nonexclusive, non-transferable license from UTIA to use these utilities for compilation of the firmware for the Xilinx PicoBlaze6 processor inside of the family of UTIA EdkDSP accelerators for the Xilinx AC701 board. The source code of these compilers is owned by UTIA and it is not provided in the release package.

The release package includes demonstration firmware in C source code for the Xilinx PicoBlaze6 processor for the family of UTIA EdkDSP accelerators for the Xilinx AC701 board.

The release package also includes compiled versions of this firmware in form of header files .h. These compiled firmware files can be downloaded into the UTIA EdkDSP accelerators for the Xilinx AC701 board without the need to install UTIA compiler binaries and the Ubuntu (x86 PC) OS under under VMware Player 3.0.0 (on Win 7, 32bit PC) or under VMware Workstation 12 Player (on Win 7, 64bit PC).

The release package deliverables also includes DVD with the Ubuntu (x86 PC) image for the VMware Player (free of charge). This image is provided to ease the installation of the UTIA EdkDSP C compiler under VMware Player 3.0.0 (on Win 7, 32bit PC) or under VMware Workstation 12 Player (on Win 7, 64bit PC).

Any and all legal disputes that may arise from or in connection with the use, intended use of or license for the software provided hereunder shall be exclusively resolved under the regional jurisdiction relevant for UTIA AV CR, v. v. i. and shall be governed by the law of the Czech Republic.

Disclaimer

This disclaimer is not a license and does not grant any rights to the materials distributed herewith. Except as otherwise provided in a valid license issued to you by UTIA AV CR v.v.i., and to the maximum extent permitted by applicable law:

(1) THIS APPLICATION NOTE AND RELATED MATERIALS LISTED IN THIS PACKAGE CONTENT ARE MADE AVAILABLE "AS IS" AND WITH ALL FAULTS, AND UTIA AV CR V.V.I. HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and

(2) UTIA AV CR v.v.i. shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under or in connection with these materials, including for any direct, or any indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or UTIA AV CR v.v.i. had been advised of the possibility of the same.

Critical Applications:

UTIA AV CR v.v.i. products are not designed or intended to be fail-safe, or for use in any application requiring fail-safe performance, such as life-support or safety devices or systems, Class III medical devices, nuclear facilities, applications related to the deployment of airbags, or any other applications that could lead to death, personal injury, or severe property or environmental damage (individually and collectively, "Critical Applications"). Customer assumes the sole risk and liability of any use of UTIA AV CR v.v.i. products in Critical Applications, subject only to applicable laws and regulations governing limitations on product liability.