Application Note

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

Evaluation of Asymmetric Multiprocessing for Zynq System-on-Modules TE0720-02-2IF, TE0720-02-1CF, TE0720-02-1QF with Carrier Board TE0701-05

Jiří Kadlec, Zdeněk Pohl
kadlec@utia.cas.cz , xpohl@utia.cas.cz
phone: +420 2 6605 2216
UTIA AV CR, v.v.i.

Revision history:

Rev.	Date	Author	Description
1	21.11.2015	Jiří Kadlec	Evaluation package for Xilinx SDK 2015.2 Vivado Lab Edition 2015.2 on Win7 64bit 4 grades of (8xSIMD) EdkDSP IP cores Designs with 4x (8xSIMD) accelerators Designs with 1x (8xSIMD) accelerator Designs with 1x (8xSIMD) accelerator with ILA Included BOOT.BIN files Packages for Zynq SoM modules: TE0720-02-2IF, TE0720-02-1CF TE0720-02-1QF (automotive)

Acknowledgements:

This work has been partially supported by the ARTEMIS JU project EMC2 "Embedded Multi-Core Systems for Mixed Criticality Applications in Dynamic and Changeable Real-Time Environments", project number ARTEMIS JU 621429 and 7H14005 (Ministry of Education Youth and Sports of the Czech Republic). See **http://www.emc2-project.eu/**.

Table of contents

Evaluation of Asymmetric Multiprocessing for Zynq System-on-Modules TE0720-02-2IF, TE0720-02-1CF, TE0720-02-1QF with Carrier Board TE0701-05 1

1. Summary 4

1.1 Key features 4

1.2 What is included 5

2. Demonstrator AMP with EdkDSP accelerators 8

2.1 Description of EdkDSP accelerators and evaluation designs 8

2.2 Resources used by the designs 10

2.3 Asymmetric multiprocessing and use of external DDR3 memory 11

2.4 Re-programmability of EdkDSP accelerator IP cores 11

2.5 Debug of the AMP system with EdkDSP accelerators in the evaluation package 11

3. Installation of AMP with EdkDSP platform 12

3.1 Import of precompiled projects and SW into Xilinx SDK 2015.2 12

3.2 Asymmetric Multiprocessing Demo 14

3.3 Performance of ARM NEON and EdkDSP accelerator 24

3.4 Evaluation of the EdkDSP C compiler 25

3.5 Asymmetric Multiprocessing Demo with Boot from SD Card 30

3.6 Asymmetric Multiprocessing Demo with single EdkDSP accelerator 32

3.7 AMP demo with single EdkDSP accelerator with In-circuit Logic Analyser (ILA) 33

3.8 Debug of Asymmetric Multiprocessing Demo with single EdkDSP accelerator with In-circuit Logic Analyser (ILA) 36

4. References 47

5. Evaluation version of the AMP demo on ZYNQ with (8xSIMD) EdkDSP accelerator IP. Designed in Vivado 2015.2. 48

6. AMP projects with evaluation version of (8xSIMD) EdkDSP accelerator IP for the Artemis EMC2 project partners. 49

7. AMP release version on ZYNQ with (8xSIMD) EdkDSP accelerator IP. Designs in Vivado 2015.2. 51

Disclaimer 53

Table of figures

Figure 1: Simplified architecture of AMP with four EdkDSP accelerators 6
Figure 2: AMP evaluation design with 4 EdkDSP accelerators in Xilinx Vivado 2015.2 IP Integrator 7
Figure 3: EdkDSP accelerator IP in Xilinx Vivado 2015.2 IP Integrator 7
Figure 4: Trenz TE0720-02-2IF system on module on the TE0701-05 carrier board with Imageon FMC board 8
Figure 5: Asymmetric multiprocessing demo running on ZYNQ with ARM, MicroBlaze and 4 (8xSIMD) EdkDSP floating point accelerators. (Vivado 2015.2 evaluation design) 9
Figure 6: AMP design on ZYNQ, ARM A9, MicroBlaze and 1x (8xSIMD) EdkDSP, with FP division 10
Figure 7: AMP designs on ZYNQ, ARM A9, MicroBlaze and 4x (8xSIMD) EdkDSP, with FP division 10

Figure 8: Select the SDK Workspace 12
Figure 9: Import Existing Projects into Workspace 13
Figure 10: Select “Copy projects into workspace” and finish the import of all projects 14
Figure 11: All projects are compiled. See IP Blocks present in the design 15
Figure 12: The bitstream system.bit is selected by the tool 16
Figure 13: Setup your COM port. Select speed to 115200 baud, and Flow control ”None” 16
Figure 14: Select “Serial” in the category Session and click Open 17
Figure 15: Select “amp_cpu_0_1x8_all.elf” code for Debug on ARM 18
Figure 16: Click Yes to switch to the debug perspective 19
Figure 17: Run amp_cpu_0_1x8_all.elf from the debugger as free running 20
Figure 18: amp_cpu_0_1x8_all.elf will write to the terminal the initial text and it will wait for the MicroBlaze part of the AMP application 21
Figure 19: Select “edkdsp_fp12_4x8_all.elf code for remote debug on MicroBlaze 21
Figure 20: Debug view with ARM and MicroBlaze. ARM is running. Start MicroBlaze 22
Figure 21: Terminal output from the debugged AMP demo application 23
Figure 22: Measured performance in MFLOP/s for FIR filter computation 24
Figure 23: Measured performance in MFLOP/s for LMS filter computation 24
Figure 24: Select the Ubuntu_EdkDSP image in the VMware Player and click “Play” 25
Figure 25: Compilation of EdkDSP firmware in Ubuntu 27
Figure 26: Initial firmware files and C listening of the LMS filter firmware for the EdkDSP 28
Figure 27: Directory shared by the Ubuntu and by the SDK 2015.2 29
Figure 28: Modification of ARM code for boot of the AMP application from SD card 30
Figure 29: Select files for generation of BOOT.BIN image file for the SD card 31
Figure 30: The default bitstream top.bit is selected automatically 33
Figure 31: Change the default hw_platform_0 to the hw_platform_0_ila 34
Figure 32: AMP design with Vivado In-Circuit Logic Analyser (ILA) on ZYNQ, ARM A9, MicroBlaze and 1x (8xSIMD) EdkDSP, with FP division 35
Figure 33: Evaluation design with single (1x8 SIMD) EdkDSP and ILA 35
Figure 34: Vivado Lab Edition 2015.2 37
Figure 35: Select Open Target 38
Figure 36: Open Hardware Target, next 38
Figure 37: Local server default, next 39
Figure 38: List of hardware targets, default, next 39
Figure 39: Summary, click Finish 39
Figure 40: Vivado Lab edition is connected to the board via jtag (single shared USB cable) 40
Figure 41: Select file with definition of probes present in HW 40
Figure 42: Compilation results of EdkDSP CC compiler. FIR filter code with probe trigger call 41
Figure 43: Trigger conditions. Selection in Vivado Lab Edition 2015.2 42
Figure 44: FIR filter waveforms after the trigger in Vivado Lab Edition 2015.2 43
Figure 45: LMS filter waveforms after the trigger in Vivado Lab Edition 2015.2 44
Figure 46: LMS filter waveforms in separate window and analog/hold format for bce_op 45
Figure 47: Separate dashboard with display of temperature and voltage in Vivado Lab Edition 2015.2 46

1. Summary

1.1 Key features

This application note describes the asymmetric multiprocessing design (AMP) based on the Xilinx application note XAPP1093 [1]. The AMP design is ported from ISE 14.5 design flow to the Xilinx Vivado 2015.2 and SDK 2015.2 design flow. The ARM Cortex A9 processor [5] works together with the MicroBlaze processor, sharing the terminal and block ram. Both processors execute program from the same external DDR3 memory. The MicroBlaze processor is controlling 4 EdkDSP floating point accelerators. Each accelerator is organised as 8xSIMD reconfigurable data path, controlled by the PicoBlaze6 controller.

This evaluation package is provided by UTIA for these three system-on-modules [2] on the **TE0701-05** carrier board [3]:

- **TE0720-02-2IF** Xilinx **XC7Z020-2CLG484I** Zynq; temperature range -40 ... +85
- **TE0720-02-1CF** Xilinx **XC7Z020-1CLG484C** Zynq; temperature range -00 ... +70
- **TE0720-02-1QF** Xilinx **XA7Z020-1CLG484Q** Automotive Zynq; temperature range -40 ... +105

This application note explains how to install and use the demonstrator on Win7 64 bit PC.

These key features are demonstrated:

- Implementation of adaptive acoustic noise cancellation on 1 of 4 accelerators is computing the recursive adaptive LMS algorithm for identification of regression filter with 1984 coefficients in single precision floating point arithmetic with sustained performance
  - 775 MFLOP/s on 125 MHz MicroBlaze processor with single 125 MHz (8xSIMD) EdkDSP accelerator
  - 338 MFLOP/s on single 666 MHz ARM Cortex A9 (with the 32bit vector floating point NEON unit)
  - 160 MFLOP/s on single 666 MHz ARM Cortex A9 (with the HW floating point unit)
  - 10 MFLOP/s on single 125 MHz MicroBlaze processor (with the 32bit floating point HW unit)
- EdkDSP accelerators can be reprogrammed by firmware. Programming is possible in C with compilation by UTIA EDKDSP C compiler. Accelerators can contain two firmware programs. Accelerators can swap in the real time from one firmware to the other firmware in only few clock cycles in the runtime.
- The floating point applications are scheduled inside of the 8x SIMD EdkDSP accelerator by the Xilinx PicoBlaze6 processor [7]. Each firmware program has maximal size of 4096 (18 bit wide words).
- The alternative firmware can be downloaded to the EdkDSP accelerators in parallel with the execution of the current firmware. This is demonstrated by swap of the FIR-filter, room-response firmware to the firmware for adaptive LMS identification of filter coefficients in the enclosed acoustic noise cancellation demo.
- The 8xSIMD EdkDSP accelerator provides single-precision floating point results, which are bit-exact identical to the reference software implementation running on the MicroBlaze processor with the Xilinx HW 32bit floating point unit.
- The 125 MHz 8xSIMD EdkDSP accelerator is 2,3x faster than the 666 MHz ARM Cortex A9 (with NEON vector processing unit), 4,8x faster than the 666 MHz ARM Cortex A9 without code optimized for NEON and 78x faster than 125 MHz MicroBlaze with HW floating point unit. These data are measured for the presented case adaptive LMS filter with 1984 coefficients.
- The FIR filter with 1984 coefficients is computed by single 125 MHz (8xSIMD) EdkDSP accelerator with the floating point performance 1227 MFLOP/s.
- Peak performance of four 125 MHz (8xSIMD) EdkDSP accelerators implemented in this demo design is 8 GFLOP/s (single precision floating point).

1.2 What is included

The asymmetric multiprocessing on ZYNQ (AMP) with the EdkDSP platform evaluation package contains these deliverables for the Win 7 64 bit:

- Evaluation versions of AMP designs. Designs work with one ARM Cortex A9 processor core with NEON vector floating point unit, one MicroBlaze and one instance or four instances of the EdkDSP accelerators, each with 8xSIMD floating point data paths. Designs are compiled in Xilinx Vivado 2015.2 [10]. See Figure 1.
- Clocks: ARM 666 MHz; MicroBlaze 100 MHz and 125 MHz; EdkDSP accelerators 100 MHz and 125 MHz.
- UTIA is providing source code for the demo applications and SW projects for the Xilinx SDK 2015.2 [11]. These source code projects are compiled with the UTIA library libwal.a serving for the EdkDSP communication.
- Included evaluation versions of the UTIA EdkDSP accelerators have HW limitation of maximal number of performed vector operations.
- UTIA EdkDSPC C compiler is provided as 3 executable applications for Ubuntu. It can be executed in the VMware Workstation 12 Player on a 64bit Win7 PC.
- The firmware is also provided in format of binary files to enable testing of accelerators without C compiler.
- Partners of the Artemis EMC2 project [9] can get from UTIA the Vivado 2015.2 HW design projects with the evaluation versions of the EdkDSP accelerators (in the Vivado 2015.2 IP netlist format) for free. See chapter 6 for specification of deliverables for the EMC2 project partners and license details.
- Release of AMP designs with the commercial version of EdkDSP accelerators for Trenz TE0720-02-2IF, TE0720-02-1CF and automotive TE0720-02-1Q system-on-modules on Trenz TE0701-05 carrier board is offered by UTIA. It includes the Vivado 2015.2 HW design projects with the EdkDSP accelerator IP cores (in netlist format) with main limitations of the evaluation version removed. See sections 7 of this application note for specification of deliverables and license details.

The Vivado 2015.2 SoC design (see Figure 2) serves for evaluation of four different EdkDSP floating point accelerator IP cores **bce_fp12_1x8_0_axiw_v1_[10|20|30|40]** (see Figure 3). Four grades **[10|20|30|40]** of the EdkDSP accelerator IP differ in HW-supported vector floating point computing capabilities:

- **bce_fp12_1x8_0_axiw_v1_10** is area optimized and supports local vector data transfers (HW supported 8xSIMD transfers inside of the accelerator IP) and vector floating point operations FPADD, FPSUB in 8xSIMD data paths.
- **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 8xSIMD 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.
- **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 8xSIMD 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).
- **bce_fp12_1x8_0_axiw_v1_40** supports identical operations as bce_fp12_1x8_0_axiw_v1_30 plus an additional HW support of dot product. It is computed in 8xSIMD data paths with HW-supported wind-up into single scalar result. This result is propagated into all 8 SIMD data planes.

All **bce_fp12_1x8_0_axiw_v1_[10|20|30|40]** accelerators IP cores 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 SIMD data plains.

All **bce_fp12_1x8_0_axiw_v1_[10|20|30|40]** accelerators IP cores (see Figure 3) are suitable for applications like adaptive normalised NLMS filters, Square-root-free versions of adaptive RLS QR filters and Adaptive RLS LATTICE filters.

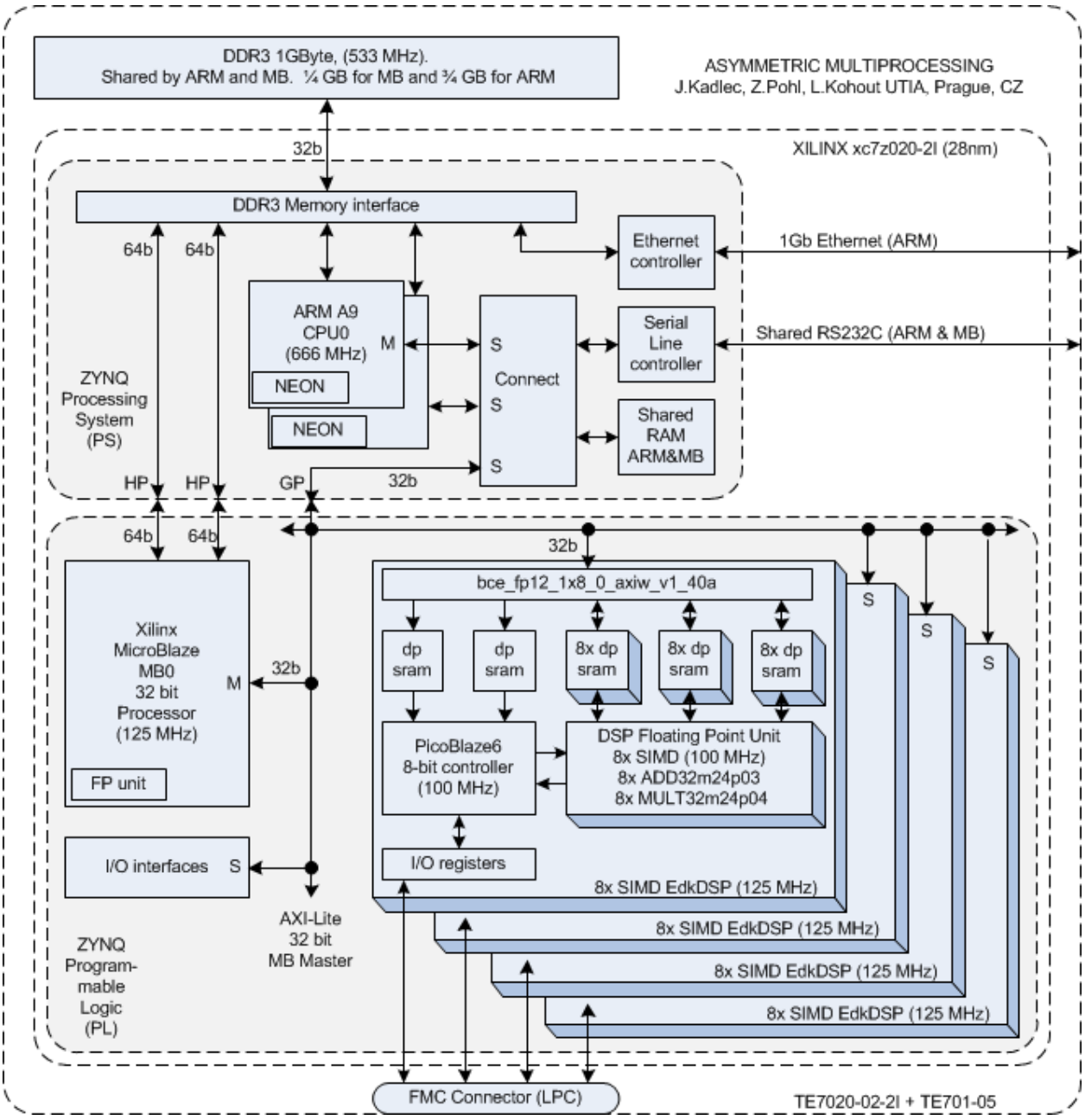

Figure 1: Simplified architecture of AMP with four EdkDSP accelerators

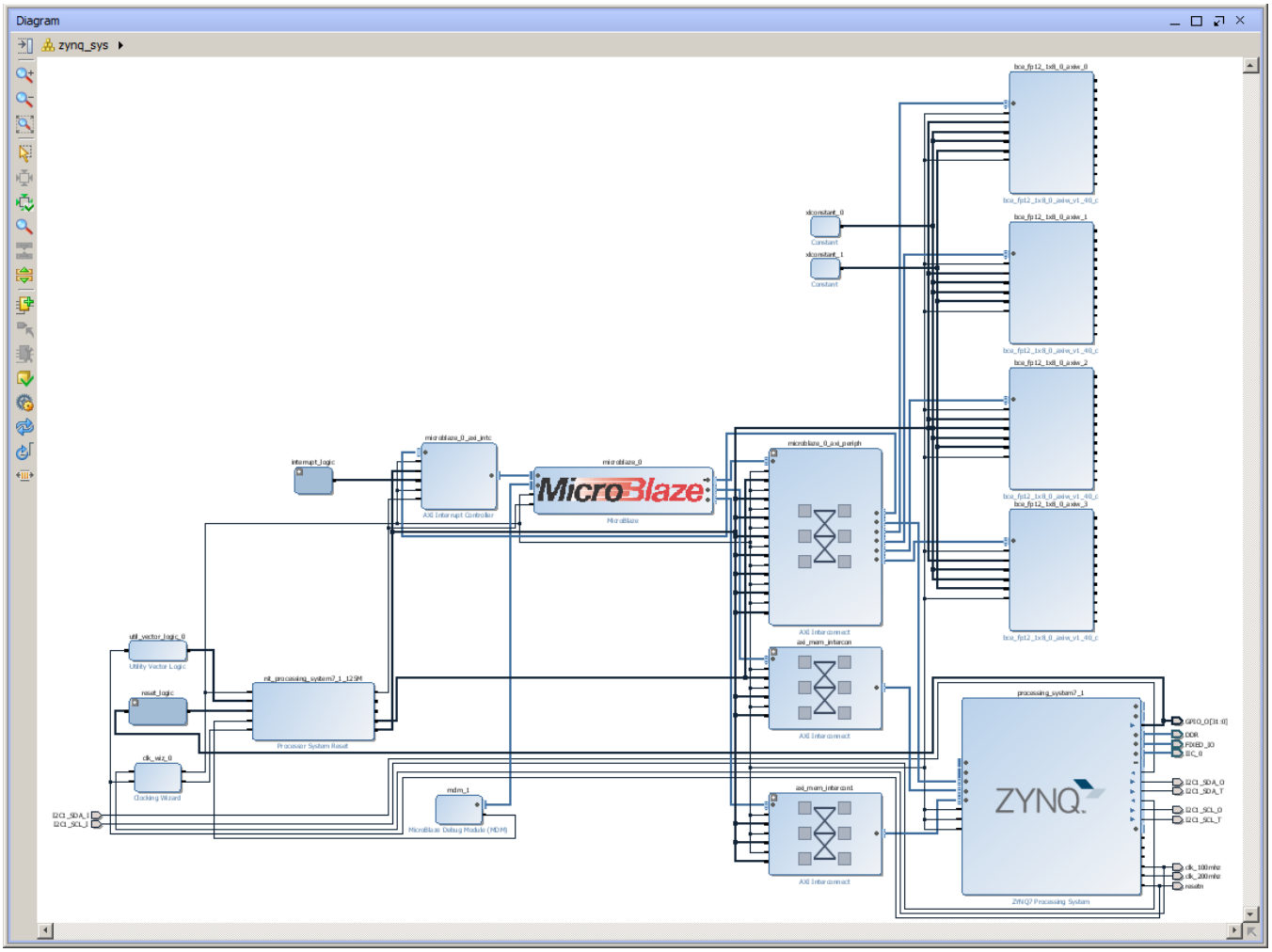

Figure 2: AMP evaluation design with 4 EdkDSP accelerators in Xilinx Vivado 2015.2 IP Integrator

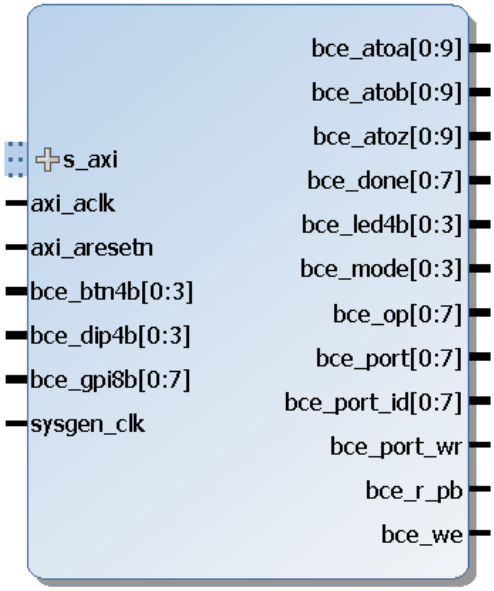

Figure 3: EdkDSP accelerator IP in Xilinx Vivado 2015.2 IP Integrator

2. Demonstrator AMP with EdkDSP accelerators

2.1 Description of EdkDSP accelerators and evaluation designs

This application note describes how to set-up and use HW designs running in an asymmetric multiprocessing architecture formed by one ARM Cortex A9 processor with NEON vector accelerator and one MicroBlaze processor with one or four (8xSIMD) EdkDSP accelerators IPs cores on the Trenz TE0720-02-2I system on module and TE0701-05 carrier board (see Figure 4 and Figure 5).

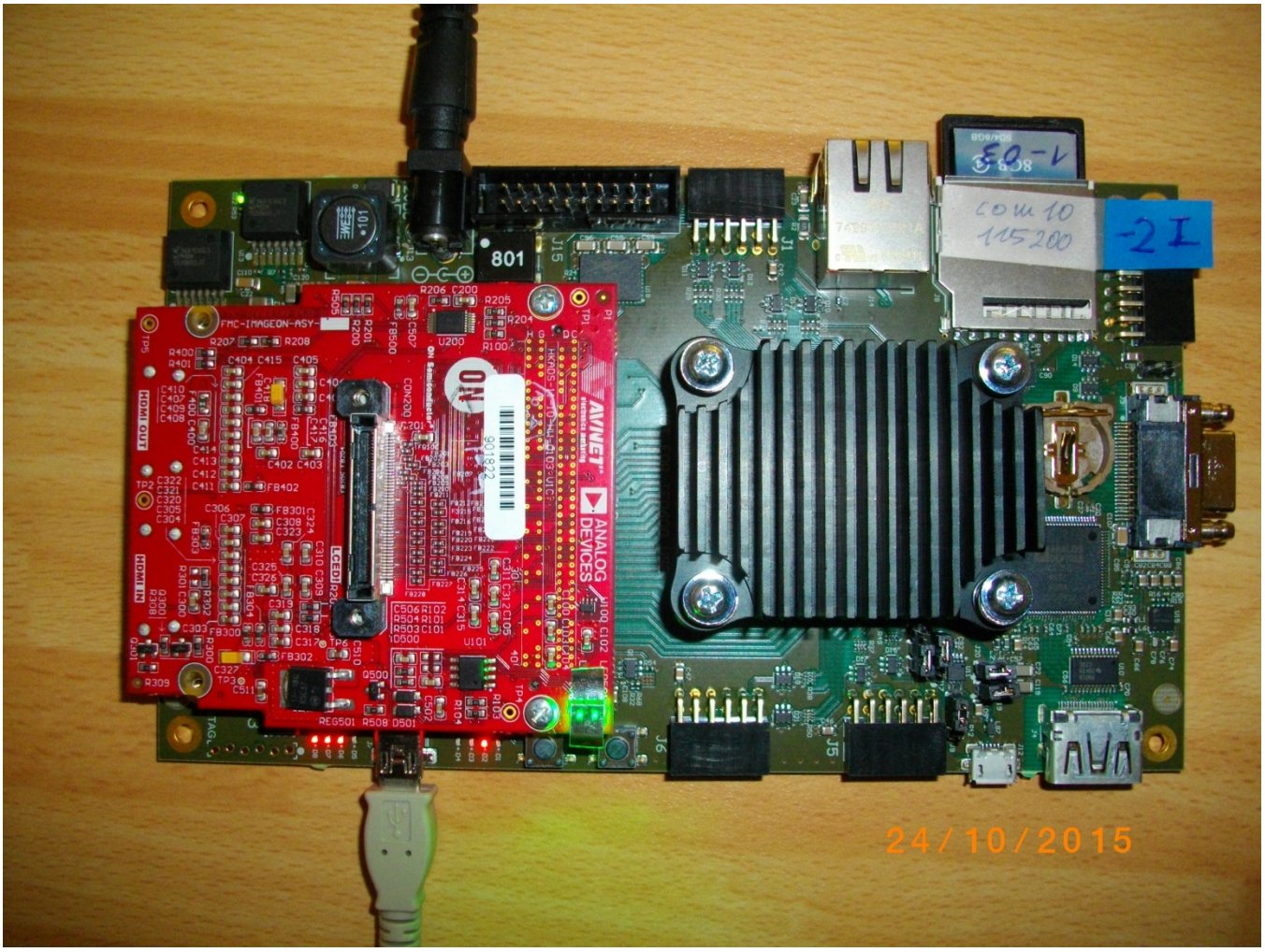

Figure 4: Trenz TE0720-02-2IF system on module on the TE0701-05 carrier board with Imageon FMC board

Figure 5: Asymmetric multiprocessing demo running on ZYNQ with ARM, MicroBlaze and 4 (8xSIMD) EdkDSP floating point accelerators. (Vivado 2015.2 evaluation design)

Common properties of precompiled Vivado 2015.2 evaluation designs:

- The EdkDSP floating point accelerators IP cores are reconfigurable during runtime by change of firmware.
- Asymmetric multiprocessing of ARM Cortex A9 and MicroBlaze system with shared external DDR3.
- HW evaluation designs have been compiled in Xilinx VIVADO 2015.2 [10].
- SW projects are compiled in Xilinx SDK 2015.2 [11].
- Project HW can be debugged with use of Xilinx Vivado Lab Edition 2015.2 [12].

Software Development Kit (SDK) 2015.2 [11] requires 64bit PC. It can be downloaded for free after registration.

Vivado Lab Edition 2015.2 [12] serves for programming and logic/serial IO debug of all 7 series, Zynq, and UltraScale devices. The Vivado Lab Edition 2015.2 requires no certificate or activation license key and supports 64- and 32-bit OS platforms (Win7 or Linux). It can be downloaded for free from [12].

2.2 Resources used by the designs

Resources used by designs are summarised in Figure 6 and Figure 7.

TE0720-02-2IF	fp32	fp32	fp32	fp32	fp32	Resources (complete PL)			EdkDSP performance	
EdkDSP vector op Clk: 125 MHz	Add Mul	Mac	Dot Prod	S8 Prod	div	FF %	Lut %	Bram no(of)	LMS Mflop/s	FIR Mflop/s
fp12_1x8_10	8x				1x	8,5	20,8	48(140)		
fp12_1x8_20	8x	8x			1x	9,5	22,4	48(140)		
fp12_1x8_30	8x	8x	8x		1x	10,5	24,5	48(140)		
fp12_1x8_40	8x	8x	8x	1x	1x	10,5	24,9	48(140)	**776**	**1228**
TE0720-02-1CF TE0720-02-1QF	fp32	fp32	fp32	fp32	fp32	Resources (complete PL)			EdkDSP performance	
EdkDSP vector op Clk: 100 MHz	Add Mul	Mac	Dot Prod	S8 Prod	div	FF %	Lut %	Bram no(of)	LMS Mflop/s	FIR Mflop/s
fp12_1x8_10	8x				1x	8,53	20,85	48(140)		
fp12_1x8_20	8x	8x			1x	9,51	22,45	48(140)		
fp12_1x8_30	8x	8x	8x		1x	10,2	24,32	48(140)		
fp12_1x8_40	8x	8x	8x	1x	1x	10,5	24,90	48(140)	**621**	**984**

Figure 6: AMP design on ZYNQ, ARM A9, MicroBlaze and 1x (8xSIMD) EdkDSP, with FP division

TE0720-02-2IF	fp32	fp32	fp32	fp32	fp32	Resources (complete PL)			EdkDSP performance	
EdkDSP vector op Clk: 125 MHz	Add Mul	Mac	Dot Prod	S8 Prod	div	FF %	Lut %	Bram no(of)	LMS Mflop/s	FIR Mflop/s
(4x) fp12_1x8_10	(4x) 8x				(4x) 1x	23,1	61,7	138(140)		
(4x) fp12_1x8_20	(4x) 8x	(4x) 8x			(4x) 1x	27,1	67,9	138(140)		
(4x) fp12_1x8_30	(4x) 8x	(4x) 8x	(4x) 8x		(4x) 1x	31,1	73,5	138(140)		
(4x) fp12_1x8_40	(4x) 8x	(4x) 8x	(4x) 8x	(4x) 1x	(4x) 1x	31,1	77,4	138(140)	**(4x) 775**	**(4x) 1227**
TE0720-02-1CF TE0720-02-1QF	fp32	fp32	fp32	fp32	fp32	Resources (complete PL)			EdkDSP performance	
EdkDSP vector op Clk: 100 MHz	Add Mul	Mac	Dot Prod	S8 Prod	div	FF %	Lut %	Bram no(of)	LMS Mflop/s	FIR Mflop/s
(4x) fp12_1x8_10	(4x) 8x				(4x) 1x	23,6	62,05	138(140)		
(4x) fp12_1x8_20	(4x) 8x	(4x) 8x			(4x) 1x	27,5	68,19	138(140)		
(4x) fp12_1x8_30	(4x) 8x	(4x) 8x	(4x) 8x		(4x) 1x	31,5	73,76	138(140)		
(4x) fp12_1x8_40	(4x) 8x	(4x) 8x	(4x) 8x	(4x) 1x	(4x) 1x	31,6	78,28	138(140)	**(4x) 622**	**(4x) 989**

Figure 7: AMP designs on ZYNQ, ARM A9, MicroBlaze and 4x (8xSIMD) EdkDSP, with FP division

2.3 Asymmetric multiprocessing and use of external DDR3 memory

Presented FPGA designs are running on the Trenz electronic TE0720-02-2I System on Module and the TE0701-05 carrier board. See Figure 3 and Figure 4. It is using the 1GB DDR3 memory with clock signal 533 MHz. The DDR3 is connected to Xilinx ZYNQ FPGA by 32 data path. The first ¾ of the DDR3 are reserved for the ARM A9 processor. The last ¼ is used by the MicroBlaze processor with the EdkDSP accelerators. The presented AMP demo is extending the Xilinx application note XAPP1093 [1] solution from the Xilinx ISE/EDK 14.5 flow to the Xilinx Vivado 2015.2 design flow. See Figure 1 for the architecture of the design.

2.4 Re-programmability of EdkDSP accelerator IP cores

Each of the four 125 MHz 8xSIMD EdkDSP floating point accelerator subsystems contains one reprogrammable Xilinx PicoBlaze6 8-bit processor and the floating point 8xSIMD DSP unit. The performance of the accelerator is application specific. In this demo, a single 8xSIMD EdkDSP unit is delivering sustained 1228 MFLOP/s in case of FIR filter with 1984 coefficients and 776 MFLOP/s in case of the adaptive LMS identification of filter with 1984 coefficients.

The Xilinx PicoBlaze6 processor has fixed configuration with size of the program memory 4096 (18 bit wide) words, 64 Bytes scratch pad RAM memory and the interrupt vector in the address 1023. Both PicoBlaze6 program memories are accessible by the MicroBlaze processor via AXI-lite bus. The PicoBlaze6 processor can execute program from each of these memories. The MicroBlaze application can write new firmware to the currently unused program memory, while the PicoBlaze6 is executing firmware from second program memory.

2.5 Debug of the AMP system with EdkDSP accelerators in the evaluation package

All EdkDSP accelerators can communicate with MicroBlaze program. The communication is using the Worker Abstraction Layer (WAL) library API. This API is used for support of writing of the debug information from the worker to the MicroBlaze terminal. MicroBlaze is using the terminal of the ARM A9 processing system, present in the ZYNQ processing system. ARM and MicroBlaze communicate via memory controller of the ZYNQ processing system. See Figure 1.

ARM and MicroBlaze can be both debugged simultaneously from the SDK 2015.2 debuggers integrated in the Xilinx SDK tool [11]. The PicoBlaze6 processors [7] 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.

Floating point data are accessed by the MicroBlaze processor via the dual ported block memories of accelerators. The MicroBlaze sides of the dual-ported accelerator memories are all mapped into the MicroBlaze memory space. The MicroBlaze processor can copy data from these dual ported memories to its global DDR3 workspace and display floating point data in the debugger.

The computation in the (8xSIMD) EdkDSP units can overlap with the communication with the DDR3. It is performed by MicroBlaze and supported by cache. A Ping-Pong swap of memory banks is used. The 8xSIMD EdkDSP firmware scheduling the parallel (8xSIMD) computation in some banks accelerator memories. The MicroBlaze program is communicating (sequentially) to/from DDR3 in another set of banks of the dual-ported accelerator memories. This process can be stopped, inspected and debugged by the MicroBlaze debugger from the SDK.

3. Installation of AMP with EdkDSP platform

3.1 Import of precompiled projects and SW into Xilinx SDK 2015.2

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 for the three system-on-module packages:

- TE0720-02-2IF system-on-module:
  - c:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8
  - c:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8_IMPORT
  - c:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8
  - c:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8_IMPORT
- TE0720-02-1CF system-on-module:
  - c:\VM_07\d_52\d_7z020_te7020-1\d_7z020_fp12_1x8
  - c:\VM_07\d_52\d_7z020_te7020-1\d_7z020_fp12_1x8_IMPORT
  - c:\VM_07\d_52\d_7z020_te7020-1\d_7z020_fp12_4x8
  - c:\VM_07\d_52\d_7z020_te7020-1\d_7z020_fp12_4x8_IMPORT
- TE0720-02-1QF system-on-module (automotive):
  - c:\VM_07\d_52\d_7z020_te7020-1Q\d_7z020_fp12_1x8
  - c:\VM_07\d_52\d_7z020_te7020-1Q\d_7z020_fp12_1x8_IMPORT
  - c:\VM_07\d_52\d_7z020_te7020-1Q\d_7z020_fp12_4x8
  - c:\VM_07\d_52\d_7z020_te7020-1Q\d_7z020_fp12_4x8_IMPORT

The installation and use is identical for all three SoM modules. Use of the package is described for TE0720-02-2IF system-on-module in detail. Other two packages can be used identically.

Start Xilinx SDK 2015.2 and select the directory for the SDK 2015.2 workspace. See Figure 8. Select **c:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\SDK_Workspace**.

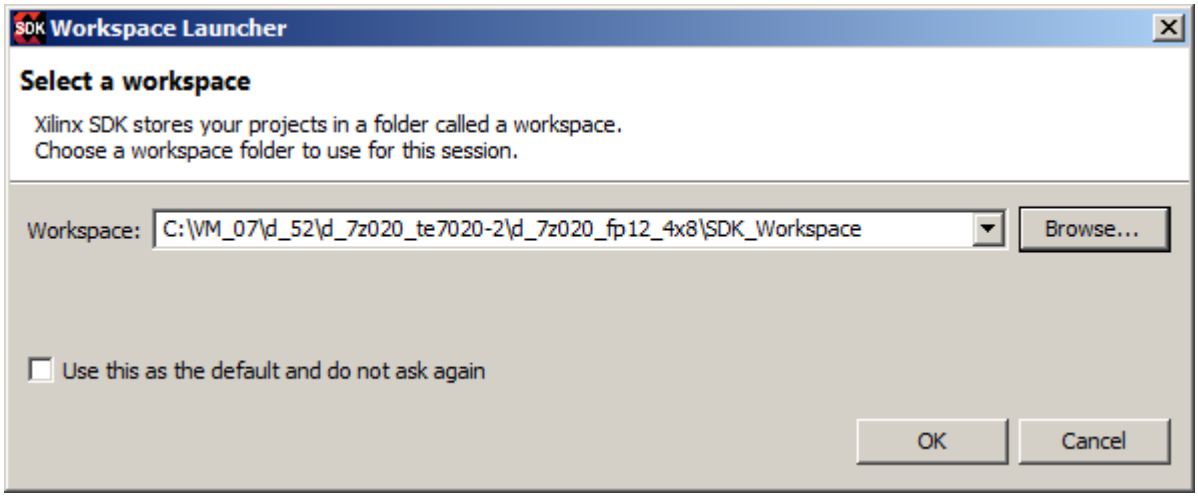

Figure 8: Select the SDK Workspace

HW and SW projects can be imported into SDK now. Select:
File -> Import -> General -> Existing Projects into Workspace
Click on Next button. See Figure 9.

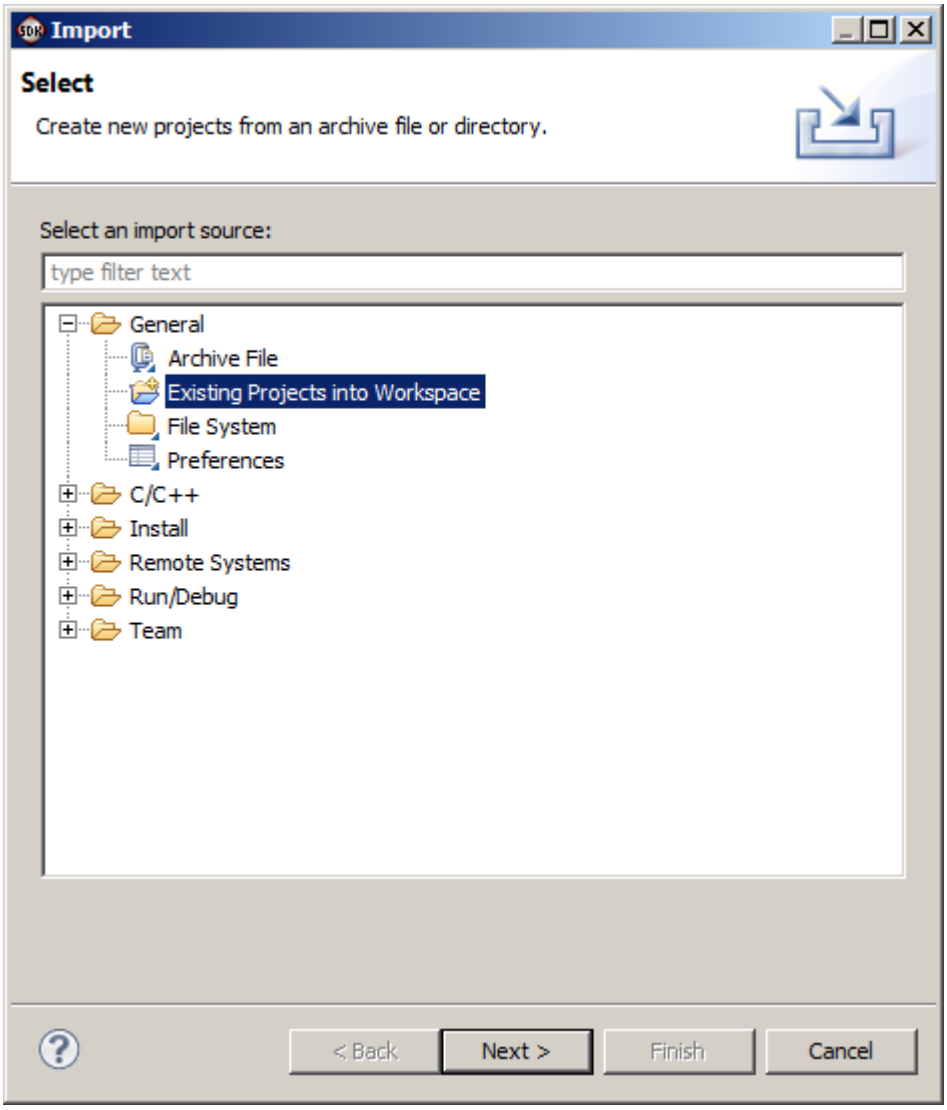

Figure 9: Import Existing Projects into Workspace

Type directory with projects to be imported. See Figure 10.

c:\VM_07\d_34_7z_te7020\d_7z020_fp12_4x8_IMPORT

Set the "**Copy projects into workspace**" check box.
Click on Finish button. See Figure 10.

All SW projects are imported into SDK workspace from the directory
c:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8_IMPORT

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 11.

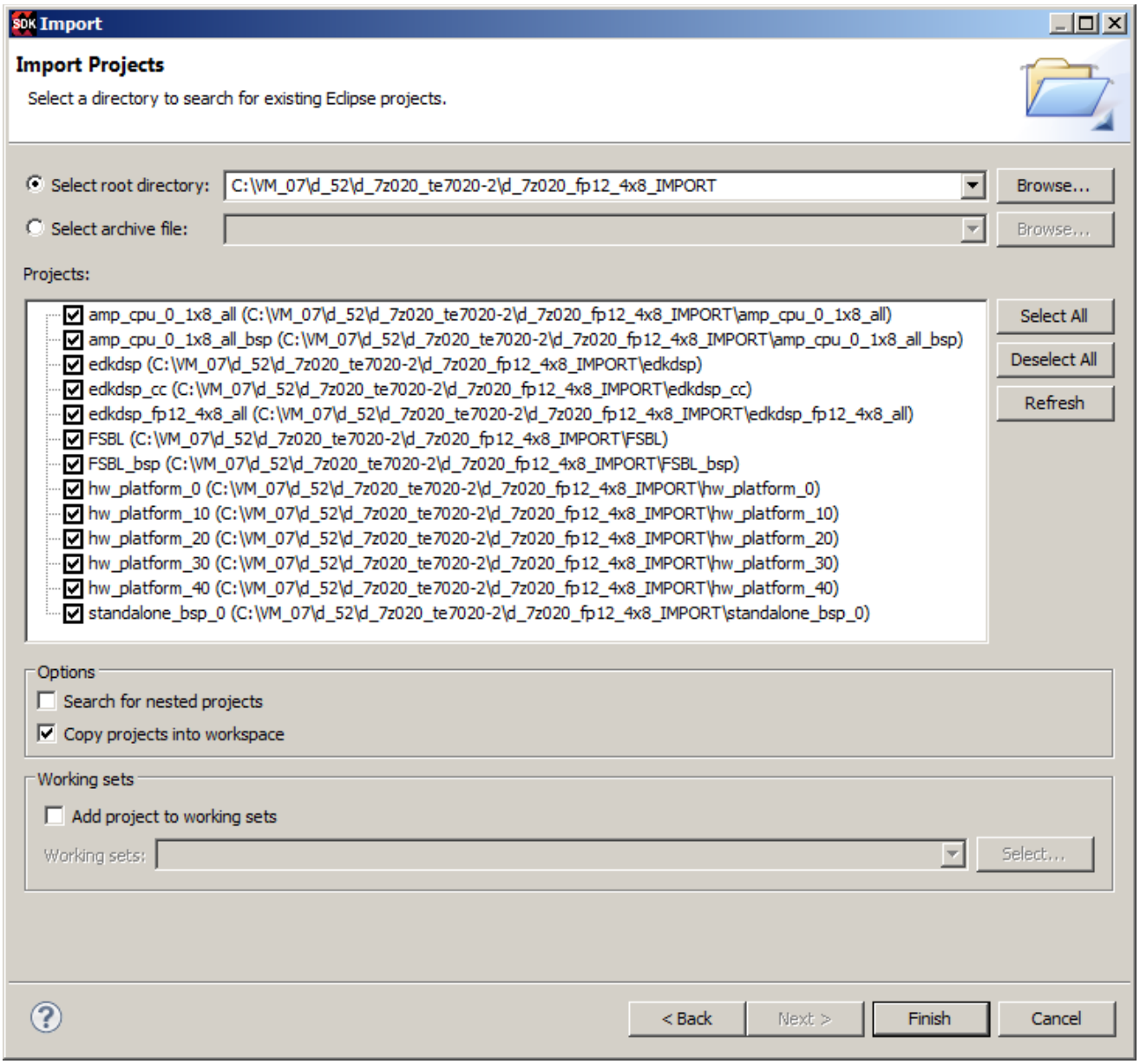

Figure 10: Select "Copy projects into workspace" and finish the import of all projects

3.2 Asymmetric Multiprocessing Demo

The "**amp_cpu_0_1x8_all**" project in the "**Project Explorer**" window of the SDK 2015.2 [7] will be used to program ARM core part of the AMP demo. The "**edkdsp_fp12_4x8_all**" project will be used to program the MicroBlaze core and the four EdkDSP accelerators of the AMP demo. The "**edkdsp**" project contains the same SW content as the "**edkdsp_fp12_4x8_all**" project and it includes in addition the precompiled **libwal.a** library for the MicroBlaze processor. The "**edkdsp_cc**" directory contains C firmware for the PicoBlaze6 controller and binary utilities for compilations of code for the EdkDSP accelerators. The UTIA EDKDSP C compiler can be executed under the VMware player [9] as an Ubuntu binary application. Inspect also the list of IP blocks and related driver versions present in the evaluation AMP design. The MicroBlaze processor has access to some ZYNQ peripheral devices [5] in the AMP design. See Figure 11.

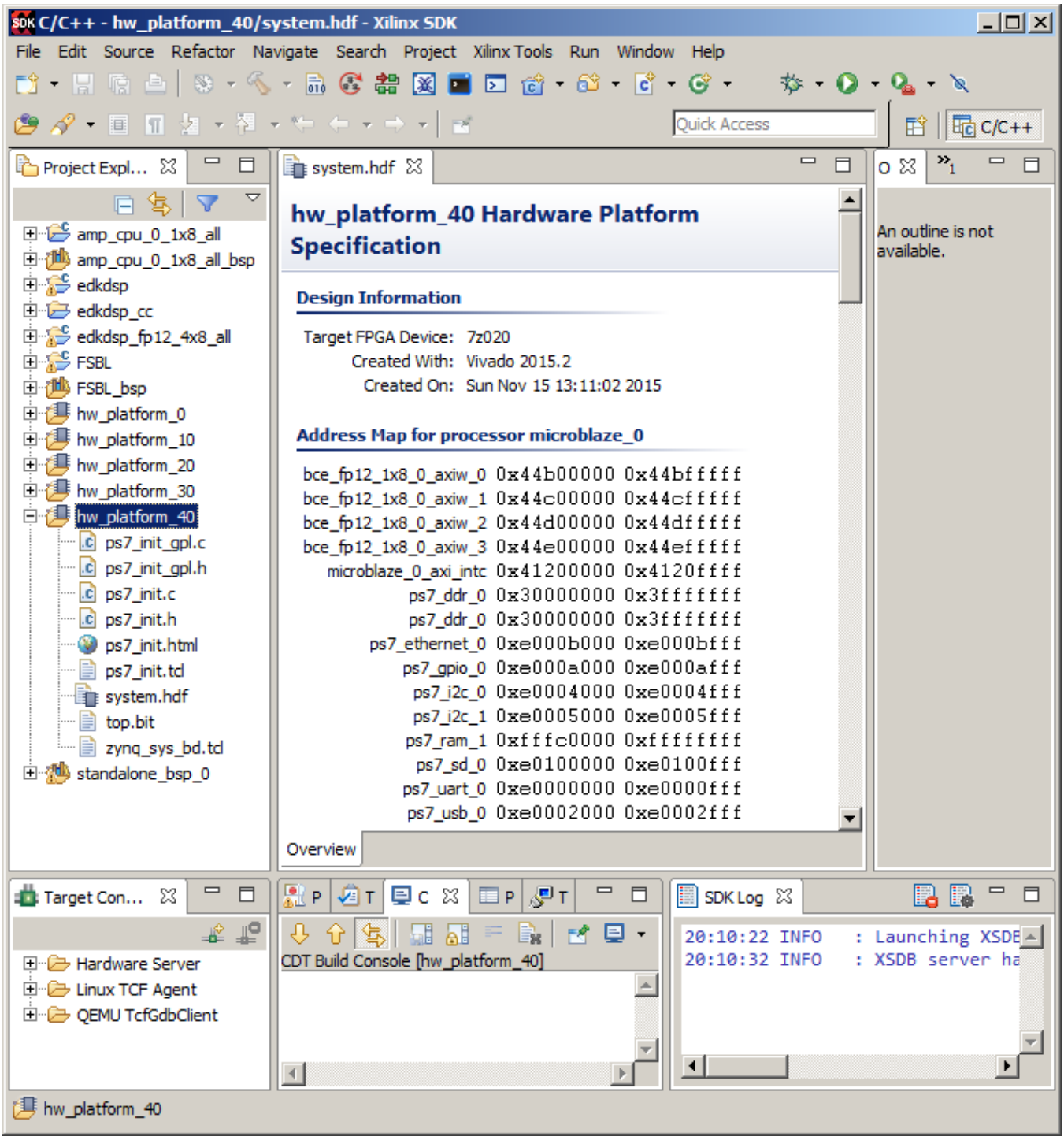

Figure 11: All projects are compiled. See IP Blocks present in the design

Connect the USB cable to the Trenz electronic TE0720-02-2I System on Module [2] via the TE0701-05 carrier board [3]. In SDK, program the board. See Figure 12.

In SDK, select: **Xilinx Tools -> Program FPGA**. It is pointing (by default) to:
c:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8
Click on the "**Program**" button.

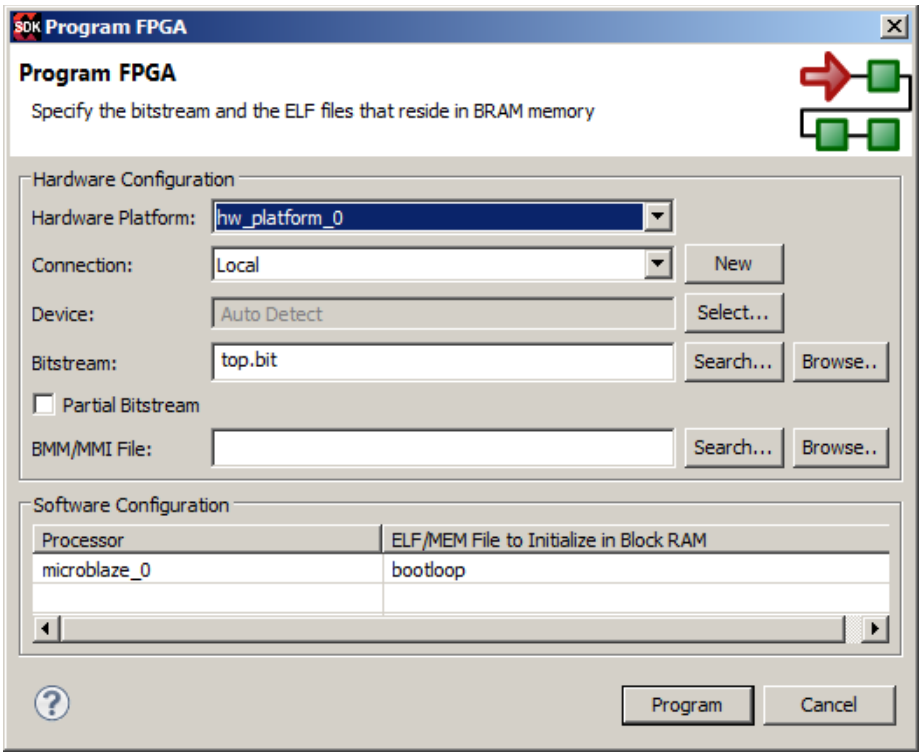

Figure 12: The bitstream system.bit is selected by the tool

The TE0720-02-2I module on TE0701-5 carrier board is programmed with the top.bit file now. On PC, start the Putty terminal. Set 115200 baud and "Flow control" to None. See Figure 13 and Figure 14.

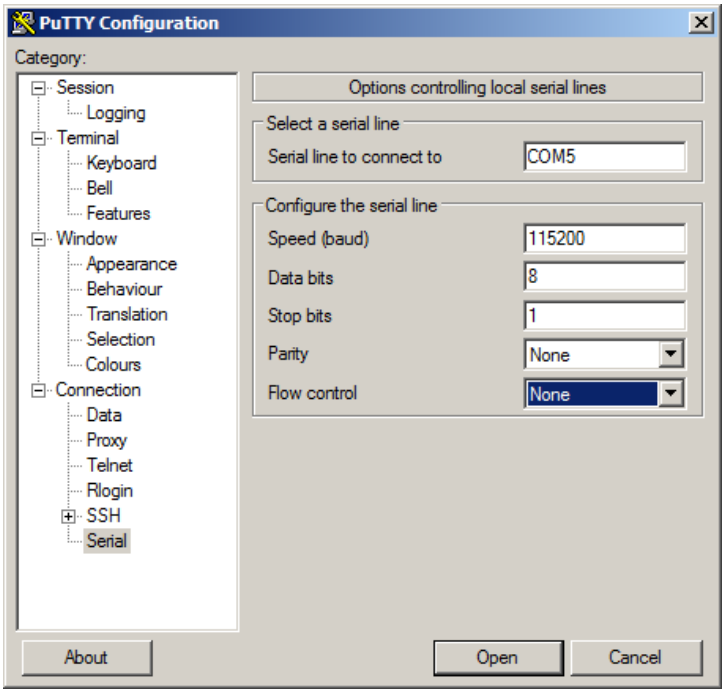

Figure 13: Setup your COM port. Select speed to 115200 baud, and Flow control "None"

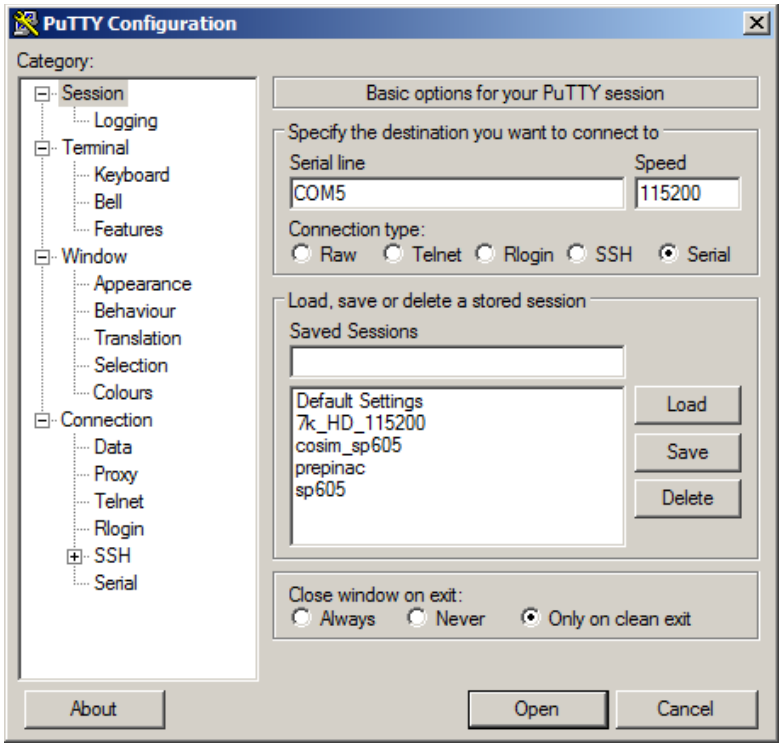

Figure 14: Select "Serial" in the category Session and click Open

The ARM part of the AMP application has to be downloaded to the DDR3 memory, now.

Select the "**amp_cpu_0_1x8_all**" project by clicking on it in the SDK Project Explorer Window.

In SDK, select:
Run -> Debug Configuration ->Xilinx C/C++application (GDB)

Click on the "**New launch configuration**" in the Debug configuration screen. The "**amp_cpu_0_1x8_all**" project is open and the ARM executable **Debug\amp_cpu_0_1x8_all.elf** is ready for download to DDR3 on the TE0720-02-2I module on TE0701-05 carrier board via the jtag cable. See Figure 15.

Click on "**Debug**" button to download the executable. See Figure 15.

Click **Yes** in the perspective switch dialog window. See Figure 16.

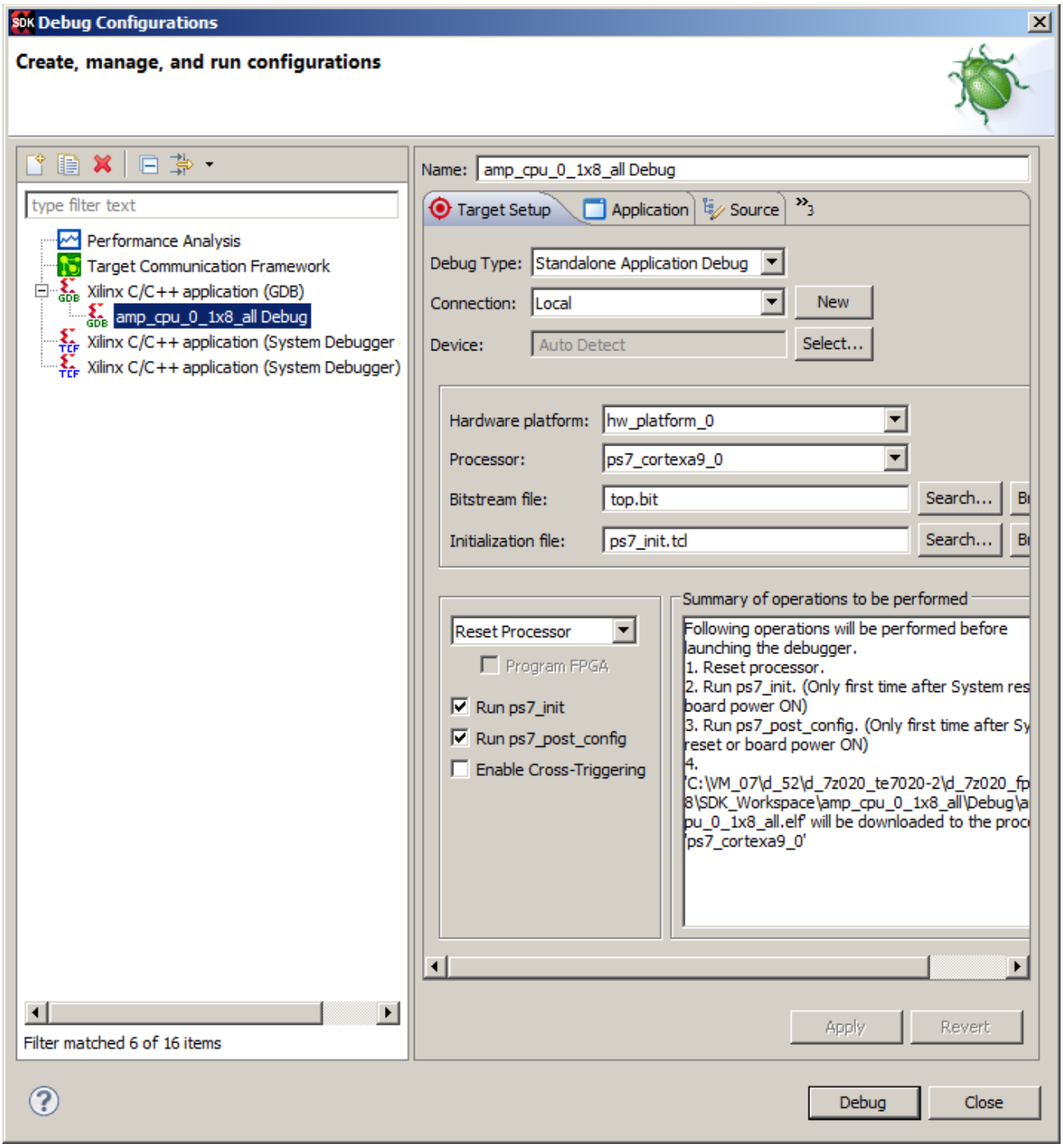

Figure 15: Select "amp_cpu_0_1x8_all.elf" code for Debug on ARM

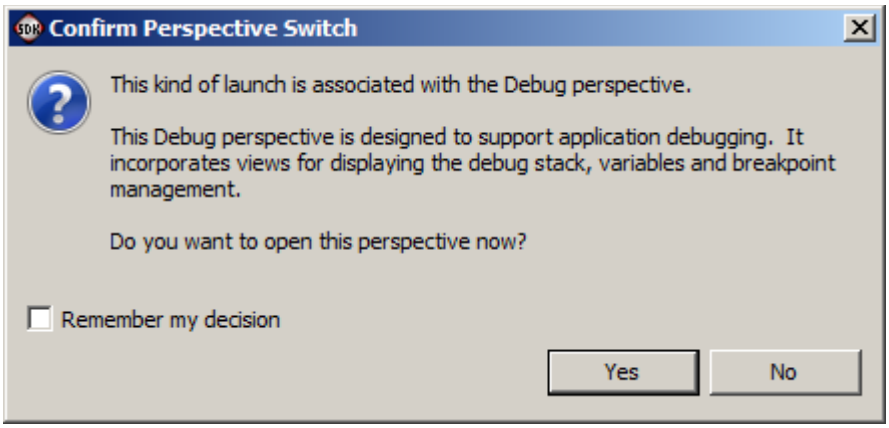

Figure 16: Click Yes to switch to the debug perspective

The debug perspective is opened and **Debug\amp_cpu_0_1x8_all** can be debugged or started on the ARM core. See Figure 17. Start the Resume button (F8) of the program from the debugger. It starts to run, with output to the terminal window. The timer is started with 3ns resolution. CPU0: on terminal is indicating the output from the Core_0 of the dual core Cortex A9 of the ZYNQ. The ARM processor is running and waiting in a pooling loop for handshake with MicroBlaze. See Figure 18.

The ARM application **amp_cpu_0_1x8_all.elf** has prepared the initial waiting loop code for the MicroBlaze processor at the address 0x30000000 in the DDR3. The MicroBlaze has been released from reset by the ARM application. MicroBlaze is running the initial loop code at the address 0x30000000 now.

The MicroBlaze application **edkdsp_fp12_1x8_all.elf** will be loaded to the DDR3 memory in next steps.

The SDK has to connect to the running MicroBlaze via jtag. The MicroBlaze processor will be stopped under the jtag control. The **edkdsp_fp12_1x8_all.elf** code will be downloaded to DDR3 and the MicroBlaze will be started again by jtag from the second debugger instance.

Select the "**edkdsp_fp12_1x8_all**" project by clicking on it in the SDK Project Explorer Window.

In SDK, select:
Run -> Debug Configuration -> Xilinx C/C++application (GDB)

Click on the "**New launch configuration**" in the Run configuration screen. The "**edkdsp_fp12_1x8_all**" project is open and the MicroBlaze executable **Debug\edkdsp_fp12_1x8_all.elf** will be ready for download to the DDR3 on the TE0720-02-2I module on TE0701 carrier board via the jtag cable, to the address 0x30000000. See Figure 19.

This Microblaze debug session will need 3 adjustments to co-exist together with the first running ARM debug session. Select "No reset", unselect "Run ps7_init" and unselect "Run ps7_port_config". Click Apply to activate this selection. Click on **Apply** to apply both changes. See Figure 19.

Click on **Debug** to start the debuger.

The program **edkdsp_fp12_1x8_all.elf** will be downloaded to DDR3 from address 0x30000000 and this second remote MicroBlaze debug section will be opened in the SDK, together with the already running ARM debug section. See Figure 20.

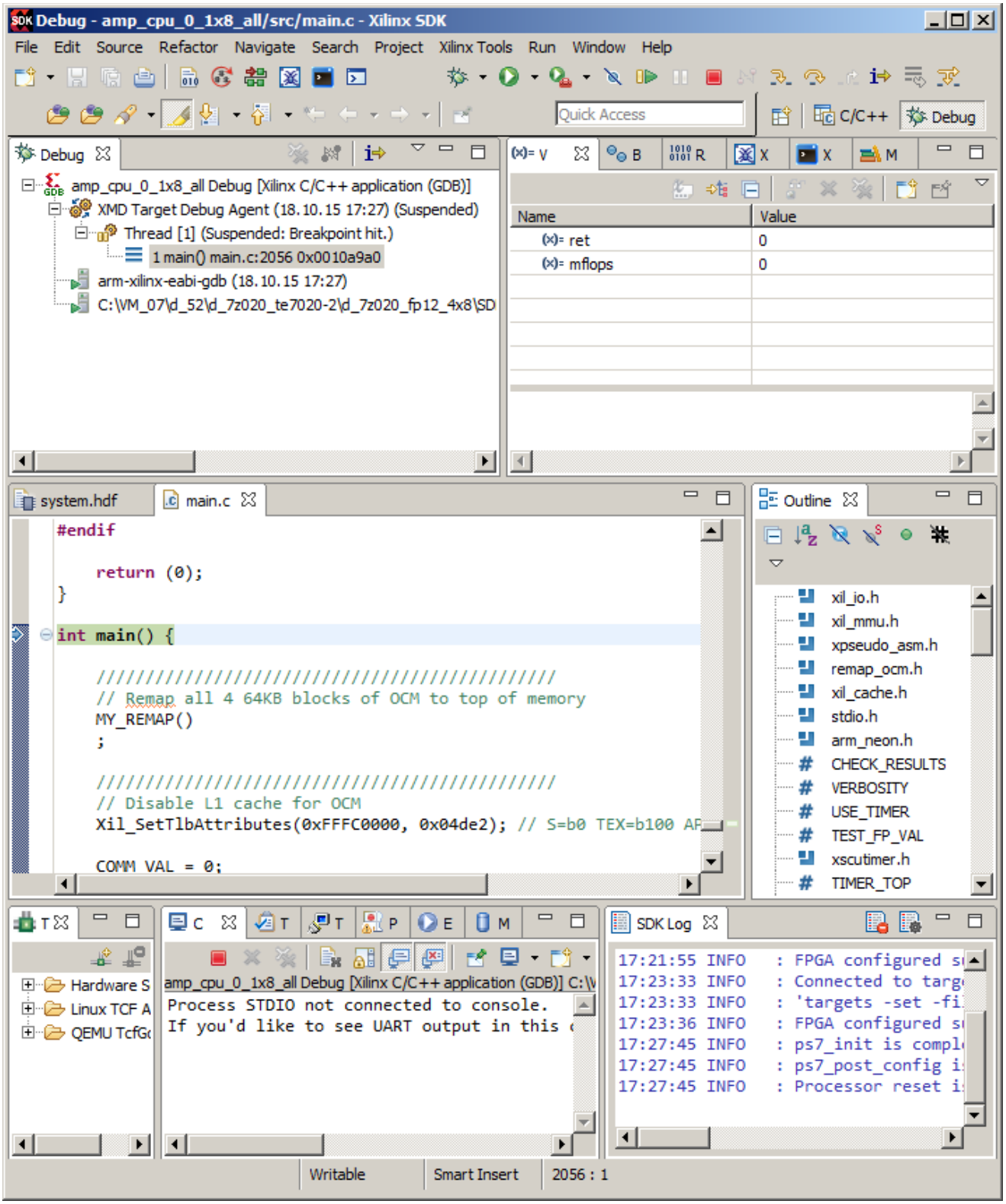

Figure 17: Run amp_cpu_0_1x8_all.elf from the debugger as free running

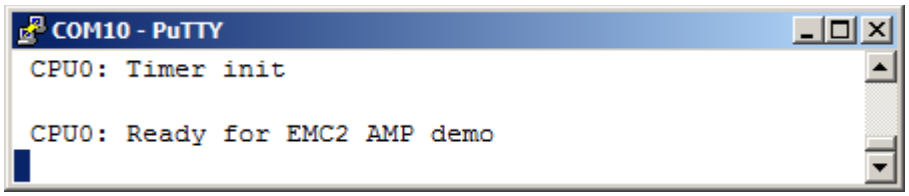

Figure 18: amp_cpu_0_1x8_all.elf will write to the terminal the initial text and it will wait for the MicroBlaze part of the AMP application

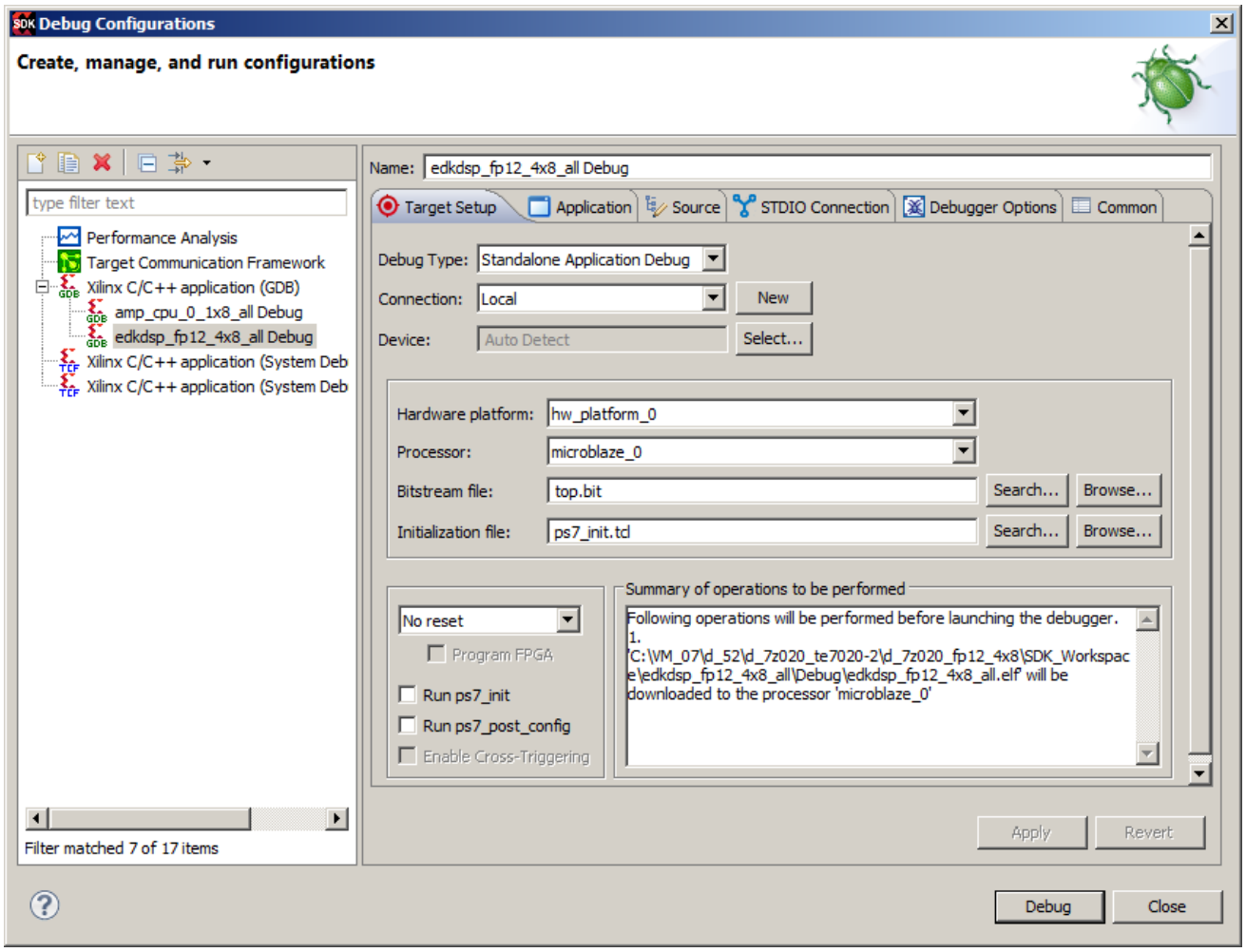

Figure 19: Select "edkdsp_fp12_4x8_all.elf code for remote debug on MicroBlaze

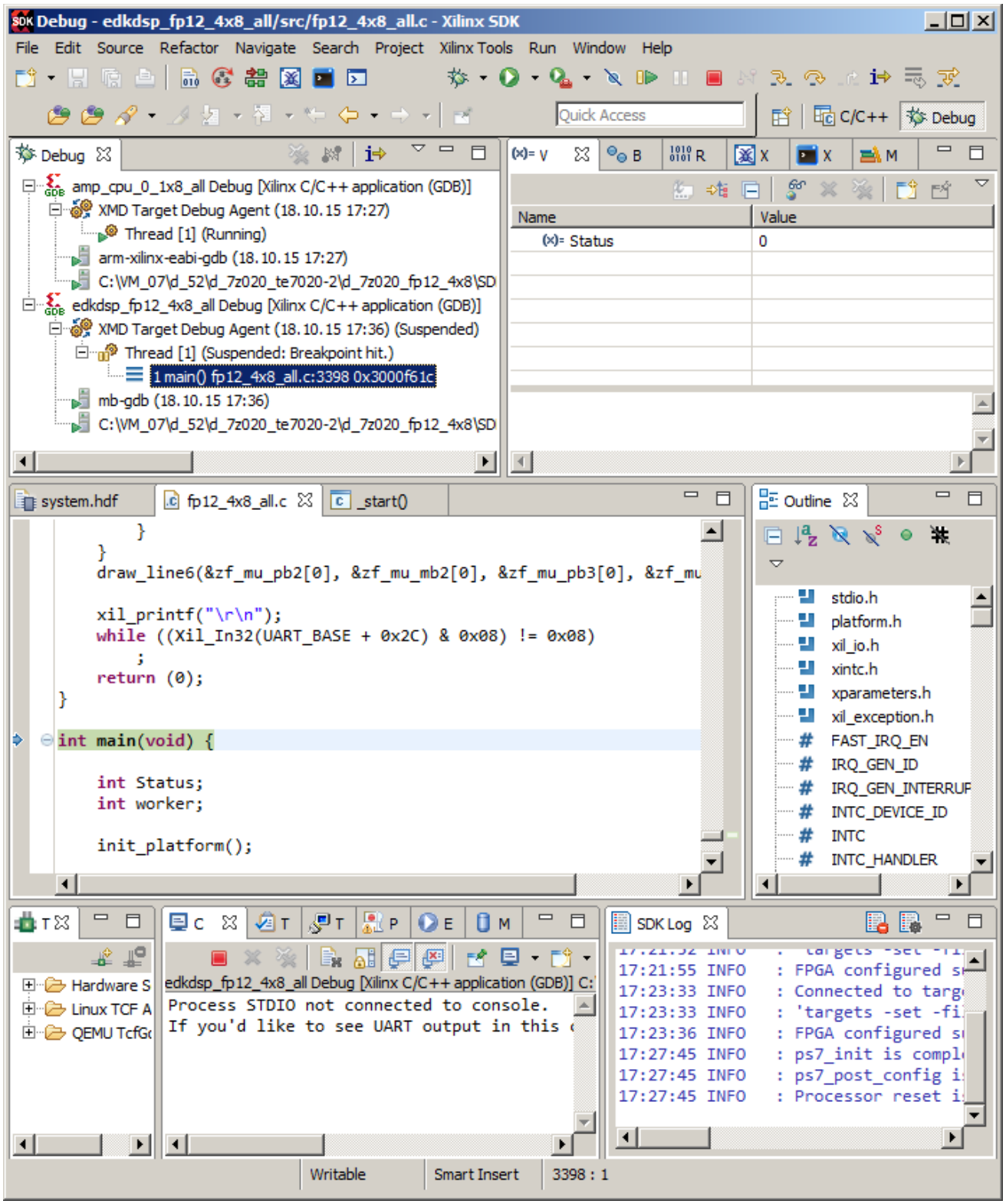

Figure 20: Debug view with ARM and MicroBlaze. ARM is running. Start MicroBlaze

The ARM application is running. The MicroBlaze application is downloaded into DDR3 and the debugger is waiting on the automatically inserted break point in the first MicroBlaze instruction. See Figure 20. You can step through the MicroBlaze program or start free run of the program.

Run the MicroBlaze to get the output on the terminal from both processors (ARM and MicroBlaze) running in parallel in the free run in the asymmetric multiprocessing configuration. See the terminal output on Figure 21.

The demo application (2000 coefficient FIR filter) and (2000 coefficient LMS identification of filter coefficients) is computed in single precision floating point on ARM CPU0 first. The same demo application (2000 coefficient FIR filter) and (2000 coefficient LMS identification of filter coefficients) is computed in single precision floating point in the first 8xSIMD EdkDSP accelerator (with support from MicroBlaze) next. Finally, the same demo application (2000 coefficient FIR filter) and (2000 coefficient LMS identification of filter coefficients) is also computed in single precision floating point on MicroBlaze with the HW floating point unit to verify the EdkDSP result.
See Figure 21.

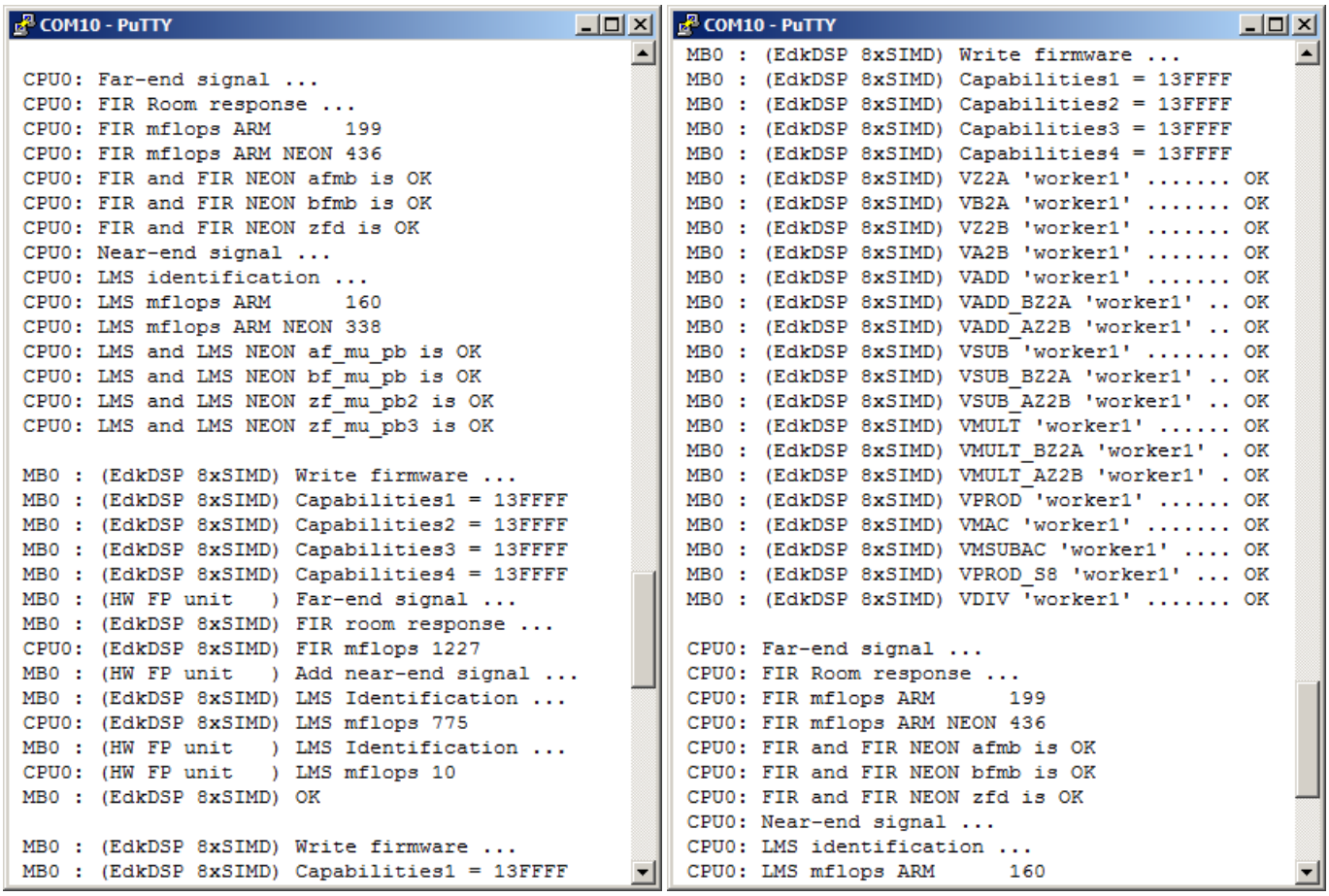

Figure 21: Terminal output from the debugged AMP demo application

The AMP demo continues with testing of all basic vector floating point operations of the 8xSIMD accelerator. These tests vector operations are also computed in single precision floating point on MicroBlaze with the HW floating point unit to verify the EdkDSP result. See Figure 21. The demo application loops infinitely. The accelerators are named worker1 … worker4. All 4 workers have identical capabilities, depending on the HW design. See Figure 21.

Each of the two parallel running processors (ARM and MicroBlaze) can be stopped/resumed/terminated from the debugger.

Terminate the debug session by this sequence of commands:

1. Stop MicroBlaze.
2. Stop Arm.
3. Terminate MicroBlaze.
4. Terminate Arm.
5. Close the debug perspective.

3.3 Performance of ARM NEON and EdkDSP accelerator

The performance of programs running on processors is dependent on the optimisation level of C compiler for ARM and C compiler for MicroBlaze (see Figure 22 and Figure 23).

Set the optimisation level –O0 for debug of the ARM code and for debug of the MicroBlaze code.

Lines FIR ARM and LMS ARM document how the ARM performance improves with the compiler optimisation levels. Additional improvement can be reached by manual transformation of the source code. This is visible in the lines FIR ARM NEON and LMS ARM NEON. The corresponding functions have been reorganised to group together parallel computation of 4 single precision floating point MAC operations in parallel. This resulted in better use of the vector capabilities of the NEON unit. See the corresponding C source code **main.c** in the ARM project **amp_cpu_0_4x8_all**.

FIR MFLOP/s	-O0	-O1	-O2	-O3
FIR ARM (666 MHz)	37	125	116	199
FIR ARM NEON (666 MHz)	69	320	321	436
FIR MB EdkDSP (125 MHz) **TE0720-02-2IF**	**1224**	**1227**	**1227**	**1227**
FIR MB EdkDSP (100 MHz) **TE0720-02-1CF, TE0720-02-1QF**	**987**	**989**	**989**	**989**

Figure 22: Measured performance in MFLOP/s for FIR filter computation

LMS MFLOP/s	-O0	-O1	-O2	-O3
LMS ARM (666 MHz)	43	129	128	160
LMS ARM NEON (666 MHz)	56	270	282	338
LMS MB EdkDSP (125 MHz) **TE0720-02-2IF**	**775**	**775**	**775**	**775**
LMS MB (125 MHz) TE0720-02-2IF	4	9	10	10
LMS MB EdkDSP (100 MHz) **TE0720-02-1CF, TE0720-02-1QF**	**621**	**622**	**622**	**622**
LMS MB (100 MHz) TE0720-02-1CF, TE0720-02-1QF	3	7	8	8

Figure 23: Measured performance in MFLOP/s for LMS filter computation

The EdkDSP (8xSIMD) accelerator works with hand optimized code for the FIR and the LMS filter. The optimisation levels of the MicroBlaze C compiler have impact on the MicroBlaze performance. This is visible in line LMS MB in Figure 23. See the corresponding C source code **fp12_4x8_all.c** in the MicroBlaze project **edkdsp_fp12_4x8_all.**

The optimisation levels of the MicroBlaze C compiler have only minimal influence on performance of the EdkDSP accelerator. This is visible in lines FIR MB EdkDSP and LMS MB EdkDSP in Figure 22 and Figure 23. See the corresponding hand optimized code for the FIR and the LMS filter **a_fp1124p0.c** and **a_fp1124p1.c** in the directory **edkdsp_cc\a**. This code is compiled by the UTIA EdkDSP C compiler. Use of this compiler is described next.

3.4 Evaluation of the EdkDSP C compiler

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 four (8xSIMD) EdkDSP accelerators in the AMP evaluation designs for the ZYNQ TE0720-02-2IF, TE0720-02-1CF or TE0720-02-1QF module on TE0701-05 carrier board.

The evaluation package includes also precompiled files with the firmware ready for download from PC to TE0720-02-2IF, TE0720-02-1CF or TE0720-02-1QF module on TE0701-05 carrier board. These files can be used to test the demo without installation of the EdkDSP C compiler to your PC.

The UTIA EdkDSP C compiler is provided as implemented as several Ubuntu binary applications. The "VMware player" software and the compatible Ubuntu image version is needed to run the UTIA EdkDSP C compiler on Windows 7 (64bit or 32bit) PC.

The Ubuntu image used in UTIA needs two DVD disks (8GB) 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 these two DVD with correct Ubuntu image from UTIA (free of charge).

Install VMware Workstation 12 Player [9] on Win 7 64 bit PC.

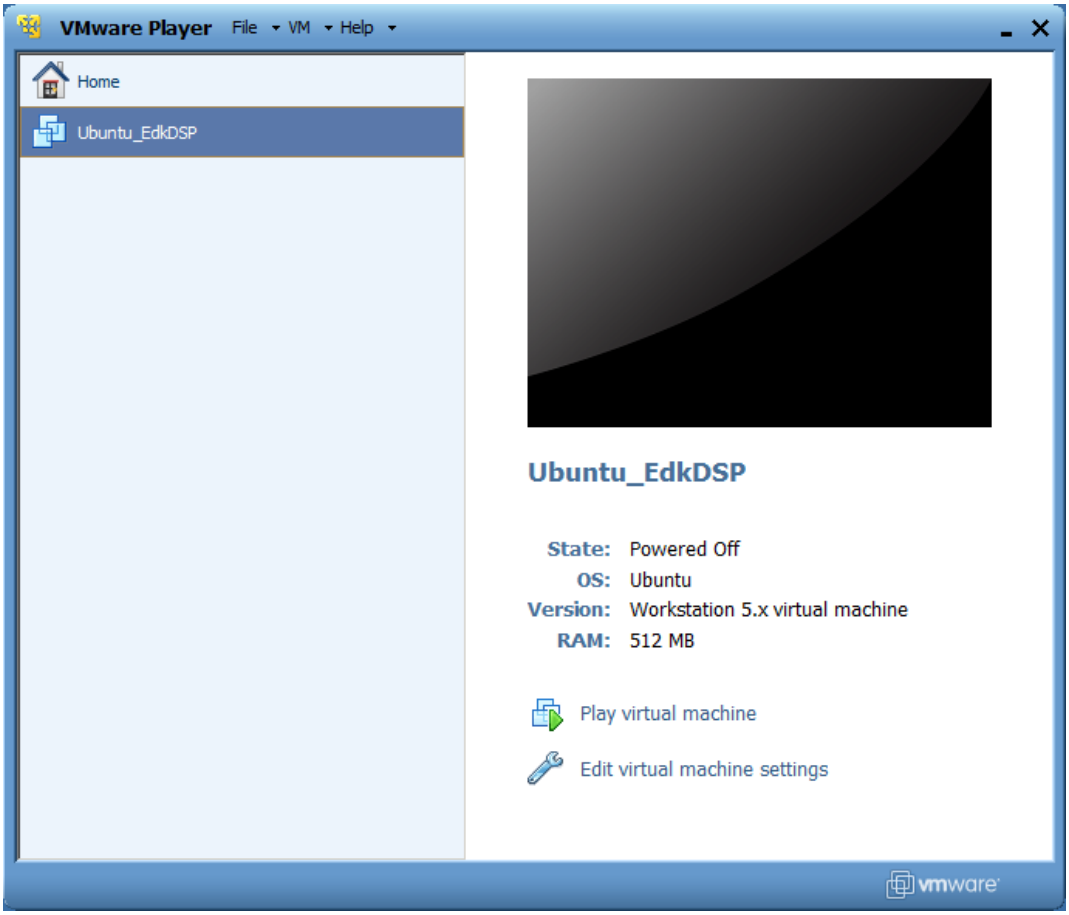

Figure 24: Select the Ubuntu_EdkDSP image in the VMware Player and click "Play"

Open the VMware Workstation 12 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 with the **__vmware_user__** for Read and Write.

In Ubuntu, open terminal and mount the PC directory **c:\VM_07** to Ubuntu by typing:
cd bin
samba_07.sh
The Windows 7 **c:/VM_07** directory is mounted to the Ubuntu OS as: **/mnt/cdrive**
In Ubuntu terminal, change the directory to:
/mnt/cdrive/d_52/d_7z020_te7020-2/d_7z020_fp12_4x8/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:
source settings.sh

In Ubuntu terminal, change the directory to the example directory: **cd a**
/mnt/cdrive/d_52/d_7z020_te7020-2/d_7z020_fp12_4x8/SDK_Workspace/edkdsp_cc/a$
See these steps in Figure 25.

C source code examples can be compiled by the script **ca_fp11.sh** with parameter **a**.
Type in the Ubuntu terminal: **ca_fp11.sh a**

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_FA1124P0_program_store.h**

The EdkDSP firmware before compilation is presented in Figure 26. See also the C code listing of the firmware for computation of the LMS in the (8xSIMD) EdkDSP platform. The EdkDSP firmware after the compilation is presented in Figure 27. See also the C code listing of the firmware for the basic test of vector operations in the (8xSIMD) EdkDSP platform.

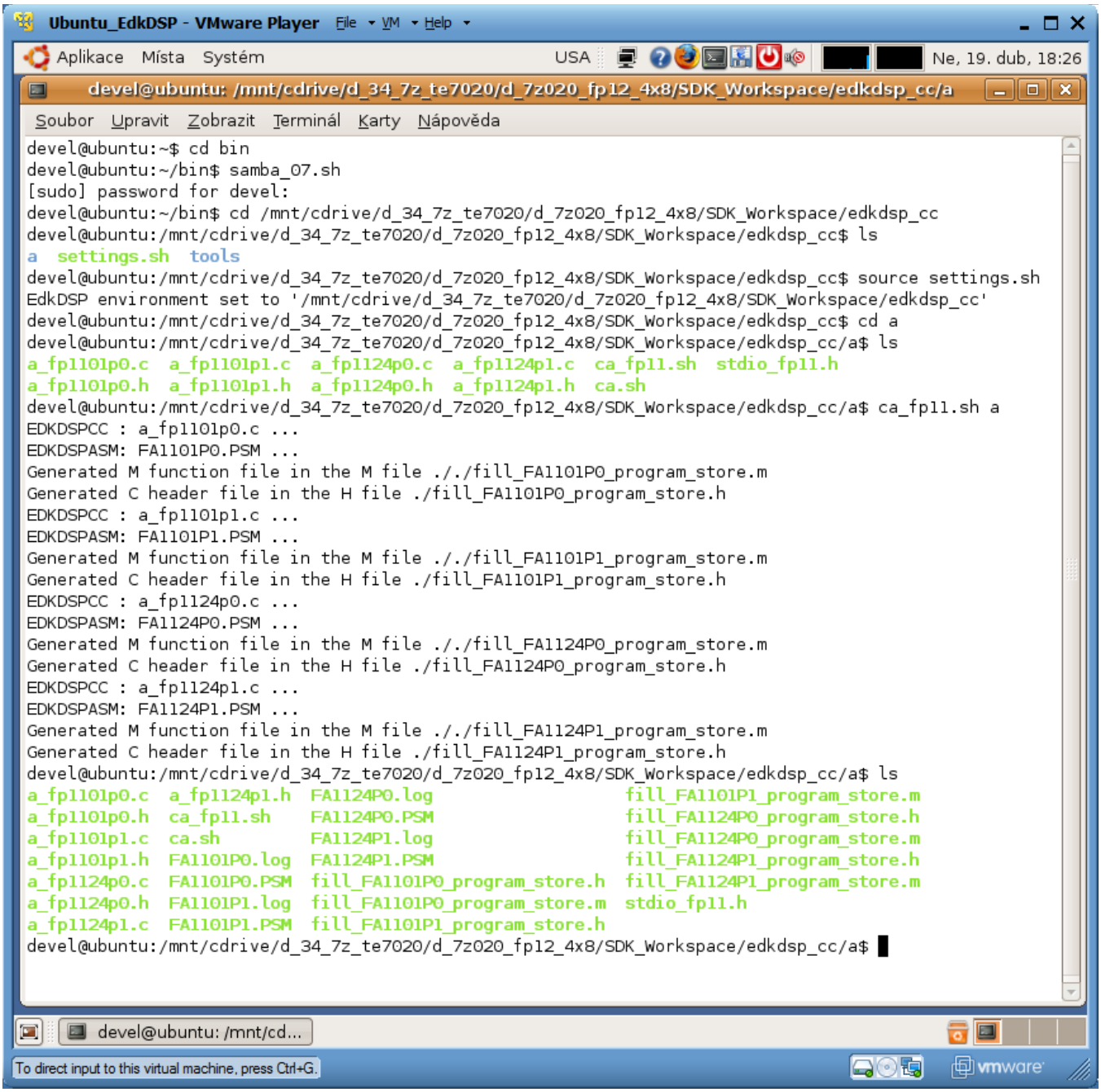

Figure 25: Compilation of EdkDSP firmware in Ubuntu

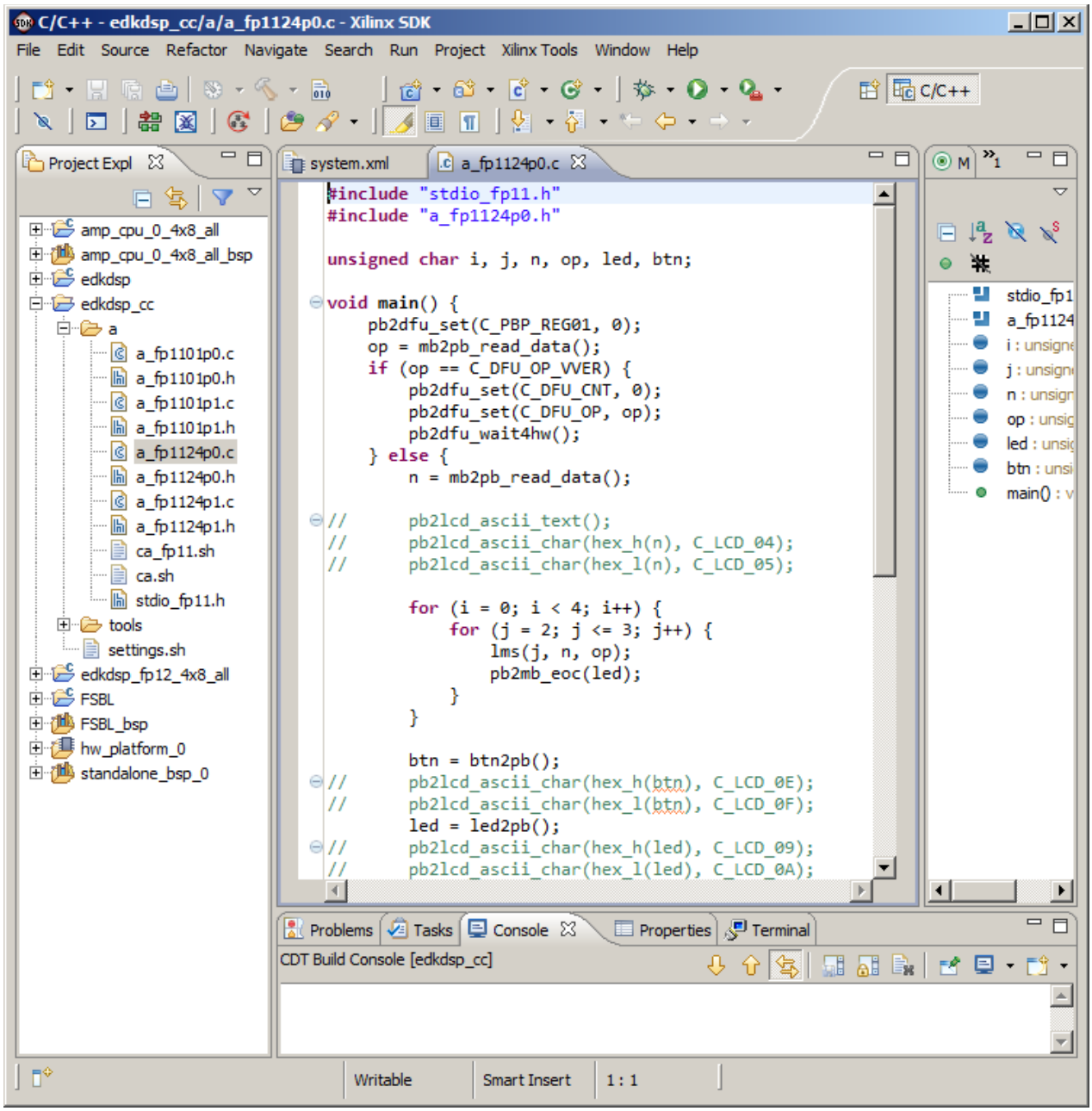

Figure 26: Initial firmware files and C listening of the LMS filter firmware for the EdkDSP

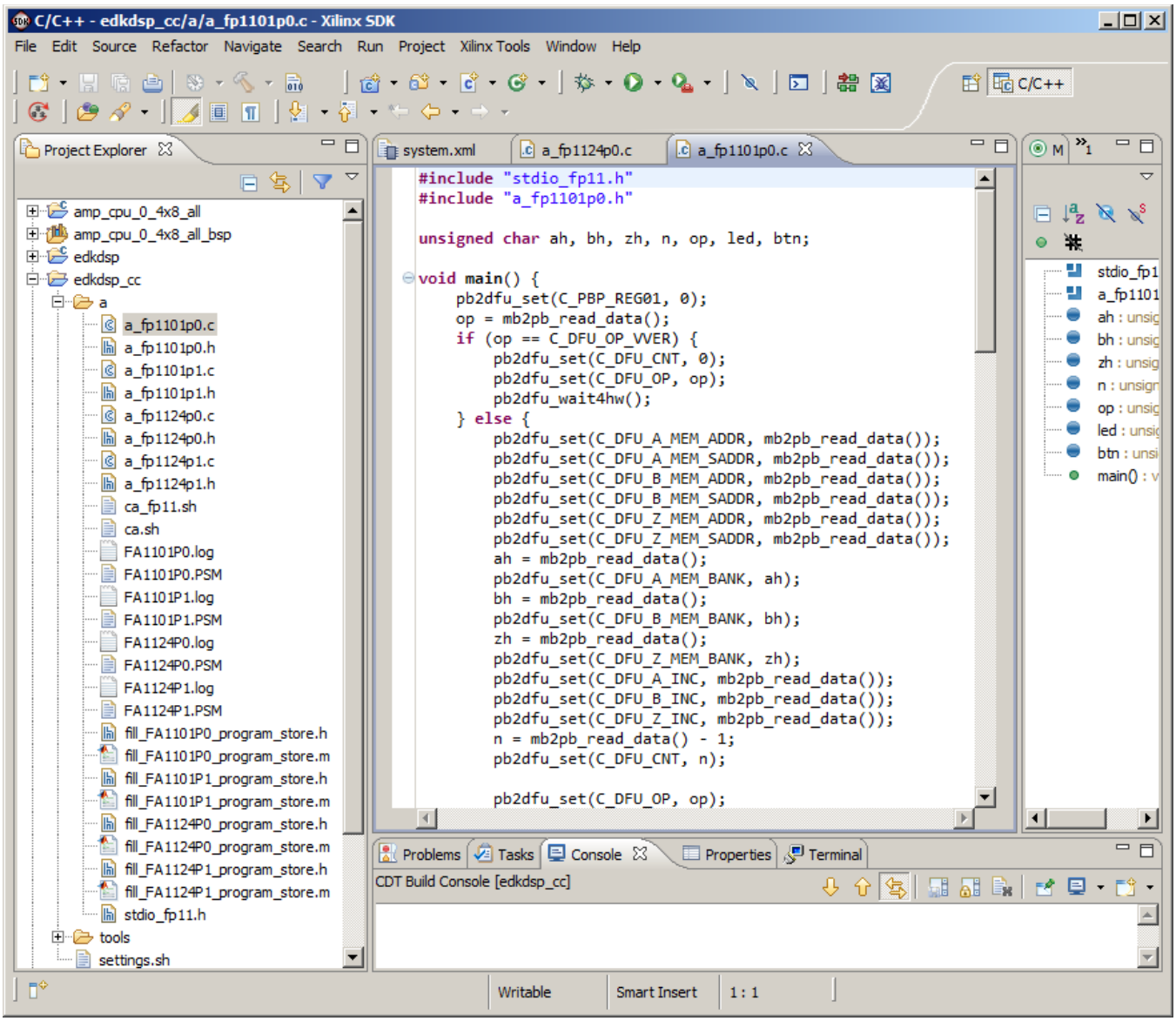

Figure 27: Directory shared by the Ubuntu and by the SDK 2015.2

To use the compiled headers in the SDK project, copy and paste the binary header files

edkdsp_cc/a/ fill_FA1101P0_program_store.h
edkdsp_cc/a/ fill_FA1101P1_program_store.h
edkdsp_cc/a/ fill_FA1124P0_program_store.h
edkdsp_cc/a/ fill_FA1124P0_program_store.h

to the SDK MicroBlaze AMP project directory: **edkdsp_fp12_4x8_all/src/** and recompile the MicroBlaze project "**edkdsp_fp12_4x8_all**". The compiled firmware for the (8xSIMD) EdkDSP will be used by the MicroBlaze part of the AMP demo for programming of the (8xSIMD) EdkDSP accelerators.

3.5 Asymmetric Multiprocessing Demo with Boot from SD Card

The AMP demo can be booted from the SD card without the need of jtag booting. This section describes steps needed to create the image. The ARM application code needs to be slightly modified.

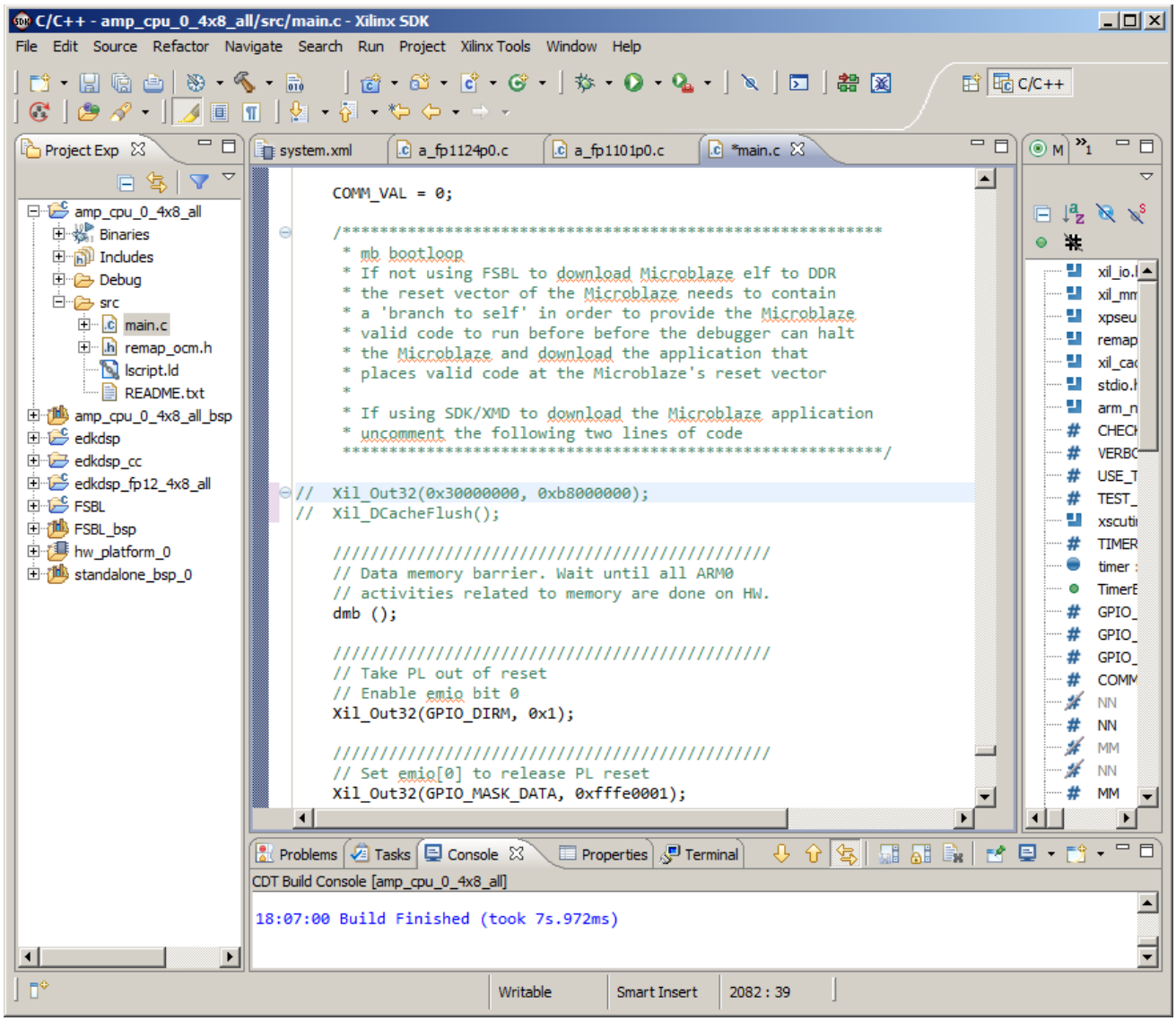

Figure 28: Modification of ARM code for boot of the AMP application from SD card

Open the **main.c** source code in **amp_cpu_0_all** project in the SDK Project Explorer. Comment the two lines as indicated in Figure 28. This is the section of program, where ARM processor writes in the MicroBlaze assembly code in hex format a loop to itself program for the MicroBlaze, starting at the DDR3 address 0x30000000. These 2 lines are needed only in the case of the jtag boot of MicroBlaze from the second remote debugger. The MicroBlase must be running in the initial infinite loop. Secondary debugger can stop the MicroBlaze and download the real debug program, starting from the DDR3 address 0x30000000.

In case of the boot of the AMP demo from the SD card the first stage boot loader ARM program **FSBL.elf** will write the final MicroBlaze program from the address 0x30000000 before giving control to the ARM application program **amp_cpu_0_4x8_all.elf.**

To generate the **BOOT.BIN** file for the SD card take these steps. In SDK, select:
Xilinx Tools -> Create Zynq Boot Image
Select Create new BIF file and browse to the directory for the file with the structure of **BOOT.BIN** file.
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\boot\boot-O3\output.bif

The file **output.bif** file will be used by the **bootgen.exe** program for creation of the **BOOT.BIN** file.
Fill all the paths as indicated in Figure 29. The sequence is important:
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\SDK_Workspace\FSBL\Debug\FSBL.elf
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\SDK_Workspace\hw_platform_40\top.bit
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\SDK_Workspace\amp_cpu_0_1x8_all\Debug\amp_cpu_0_1x8_all.elf
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\SDK_Workspace\edkdsp_fp12_4x8_all\Debug\edkdsp_fp12_4x8_all.elf

The first is the ARM first stage boot loader; the second is the .bit file with the HW content of the PL part of the chip; the third is the application code for ARM processor and the fourth is the application code for the MicroBlaze processor.

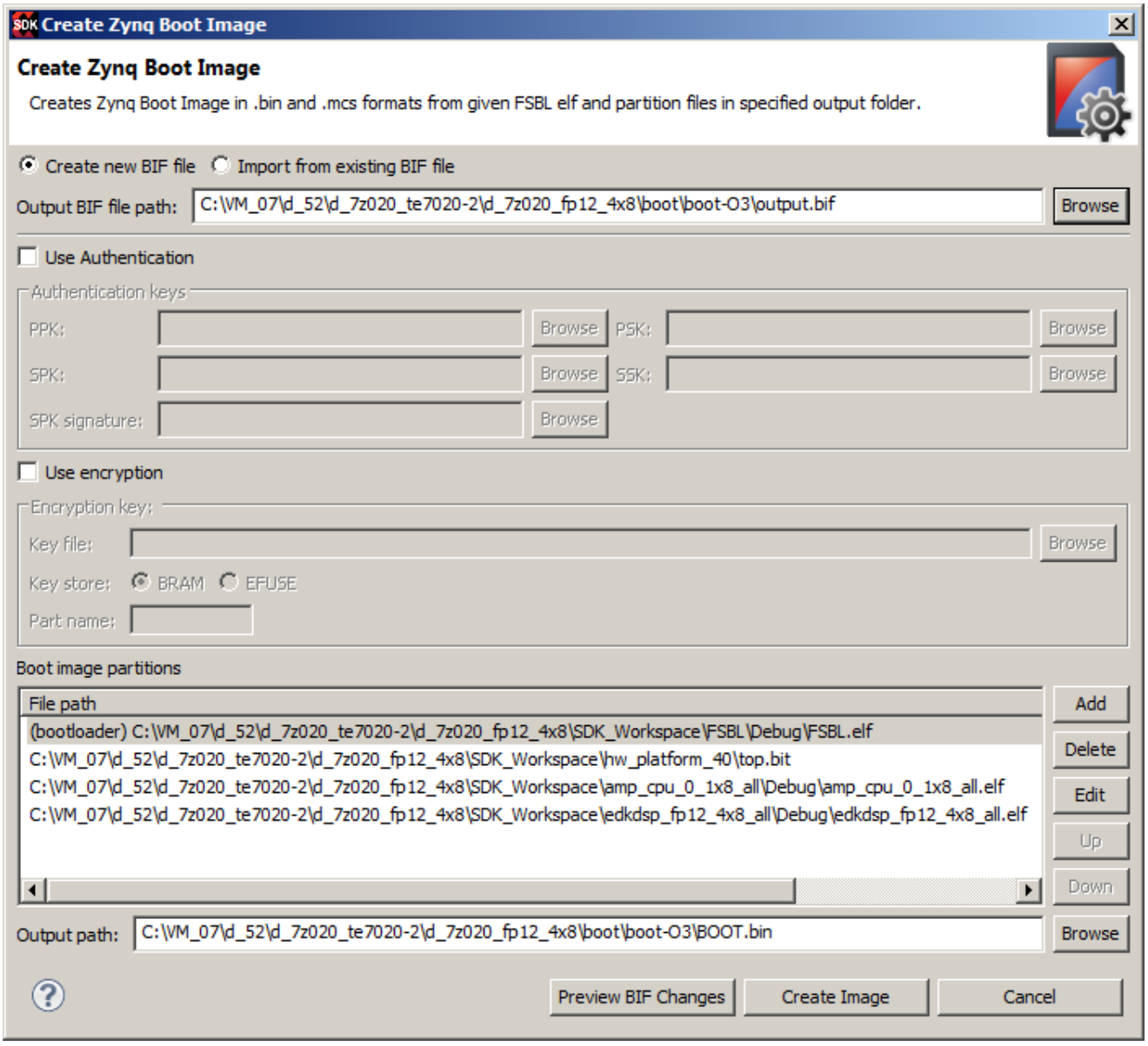

Figure 29: Select files for generation of BOOT.BIN image file for the SD card

Click Create Image and the tool will generate file
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\boot_ila\boot-O3\BOOT.bin
Copy **BOOT.bin** file to the top level directory SD card.

Insert the SD card to the TE0701 carrier board and reset the board. ZYNQ will boot from the SD card with output to the terminal. The terminal output will be identical to Figure 21.

The evaluation package includes the precompiled files for different optimisations of ARM and MicroBlaze compilers:

C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\boot\boot-O0\BOOT.BIN
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\boot\boot-O1\BOOT.BIN
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\boot\boot-O2\BOOT.BIN
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_4x8\boot\boot-O3\BOOT.BIN

You can use one of these files, copy it to the SD card and boot from the SD card, as the first evaluation step.

3.6 Asymmetric Multiprocessing Demo with single EdkDSP accelerator

The evaluation design with ARM Cortex A9 processor and MicroBlaze processor with single (8xSIMD) EdkDSP accelerator contains two HW platforms. Platforms have single accelerator and are configured without or with the Xilinx In-circuit Logic Analyser (ILA) for debug of EdkDSP accelerator.

C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\d_7z020_fp12_1x8
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\d_7z020_fp12_1x8_IMPORT

You can repeat all evaluation and compilation steps as described in this application note for the system without ILA: In SDK, select: **Xilinx Tools -> Program FPGA** select the default **“hw_platform_40”**
Click on the “**Program**” button.

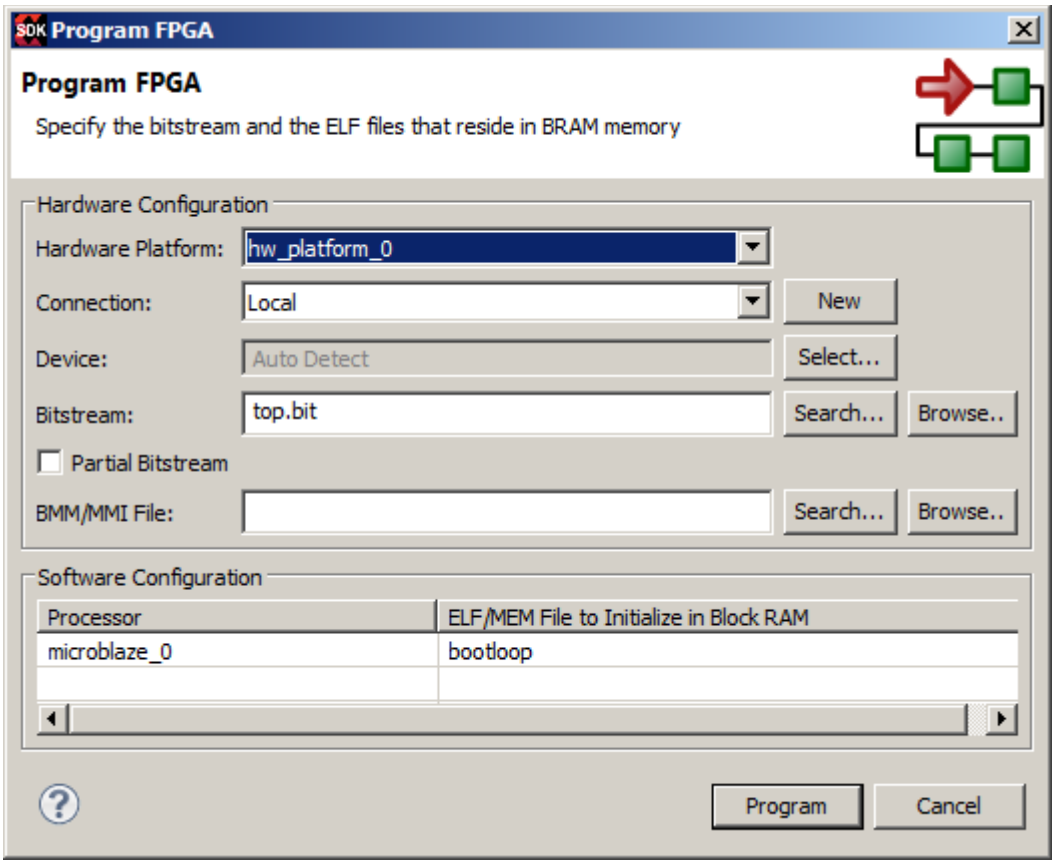

Figure 30: The default bitstream top.bit is selected automatically

This version of evaluation HW can be also quickly tested by booting from SD card with one of these files:

C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\boot\boot-O0\BOOT.BIN
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\boot\boot-O1\BOOT.BIN
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\boot\boot-O2\BOOT.BIN
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\boot\boot-O3\BOOT.BIN

SW projects (for the platform with single EdkDSP accelerator) can be also re-compiled, debugged and tested on the enclosed demo demonstrating the AMP for ARM, MicroBlaze and EdkDSP PicoBlaze6. The compilation and use of the demo has been already described in this application note for platform with 4 EdkDSP accelerators.

3.7 AMP demo with single EdkDSP accelerator with In-circuit Logic Analyser (ILA)

The evaluation design with ARM Cortex A9 processor and MicroBlaze processor with single (8xSIMD) EdkDSP accelerator contains second HW platforms configured with the Xilinx In-circuit Logic Analyser (ILA) for debug of EdkDSP accelerator from the Vivado 2015.2 Lab Edition tool. Free downloadable from Xilinx support portal [12]: The AMP platform with single EdkDSP accelerator and ILA support is present in the directory:

C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\d_7z020_fp12_1x8
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\d_7z020_fp12_1x8_IMPORT

You can repeat all evaluation and compilation steps as described in this application note for the system without ILA: In SDK, select: **Xilinx Tools -> Program FPGA** select the default **"hw_platform_40_ila"**
Click on the "**Program**" button.

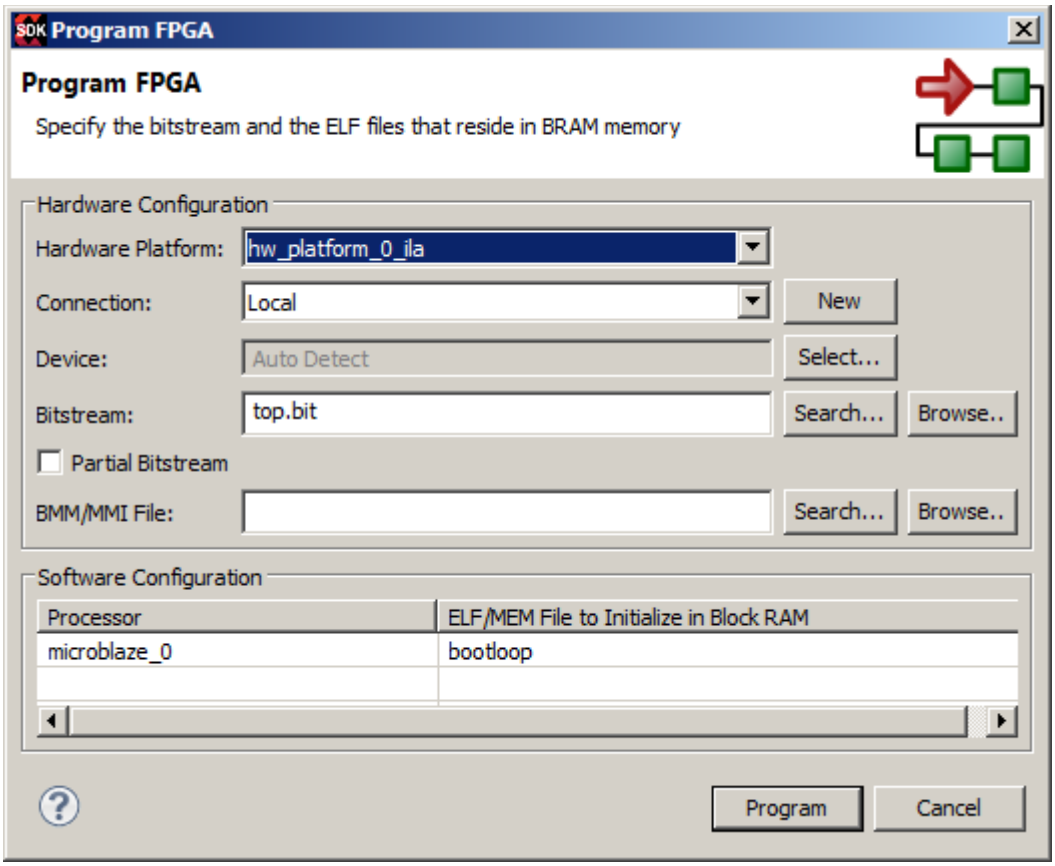

Figure 31: Change the default hw_platform_0 to the hw_platform_0_ila

This version of evaluation HW with ILA support can be also quickly tested by booting from SD card with one of these files:

C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\boot_ila\boot-O0\BOOT.BIN
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\boot_ila\boot-O1\BOOT.BIN
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\boot_ila\boot-O2\BOOT.BIN
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\boot_ila\boot-O3\BOOT.BIN

The implemented precompiled In-Circuit Logic Analyser (ILA) stores 32k samples of all output signals of the (8xSIMD) EdkDSP Accelerator. See Figure 33 and detailed zoom of EdkDSP accelerator in Figure 3. Resources used by the design with ILA are summarised in Figure 32. Compare with designs without ILA presented in Figure 6 and Figure 7.

TE0720-02-2IF	fp32	fp32	fp32	fp32	fp32	Resources (complete PL)			EdkDSP performance	
EdkDSP vector op Clk: 125 MHz	Add Mul	Mac	Dot Prod	S8 Prod	div	FF %	Lut %	Bram no(of)	LMS Mflop/s	FIR Mflop/s
fp12_1x8_10	8x				1x	14,0	30,9	114(140)		
fp12_1x8_20	8x	8x			1x	15,0	32,6	114(140)		
fp12_1x8_30	8x	8x	8x		1x	16,0	34,6	114(140)		
fp12_1x8_40	8x	8x	8x	1x	1x	16,0	35,1	114(140)	**776**	**1228**
TE0720-02-1CF **TE0720-02-1QF**	fp32	fp32	fp32	fp32	fp32	Resources (complete PL)			EdkDSP performance	
EdkDSP vector op Clk: 100 MHz	Add Mul	Mac	Dot Prod	S8 Prod	div	FF %	Lut %	Bram no(of)	LMS Mflop/s	FIR Mflop/s
fp12_1x8_10	8x				1x	14,0	31,0	114(140)		
fp12_1x8_20	8x	8x			1x	15,0	32,61	114(140)		
fp12_1x8_30	8x	8x	8x		1x	16,0	34,48	114(140)		
fp12_1x8_40	8x	8x	8x	1x	1x	16,0	35,03	114(140)	**621**	**984**

Figure 32: AMP design with Vivado In-Circuit Logic Analyser (ILA) on ZYNQ, ARM A9, MicroBlaze and 1x (8xSIMD) EdkDSP, with FP division

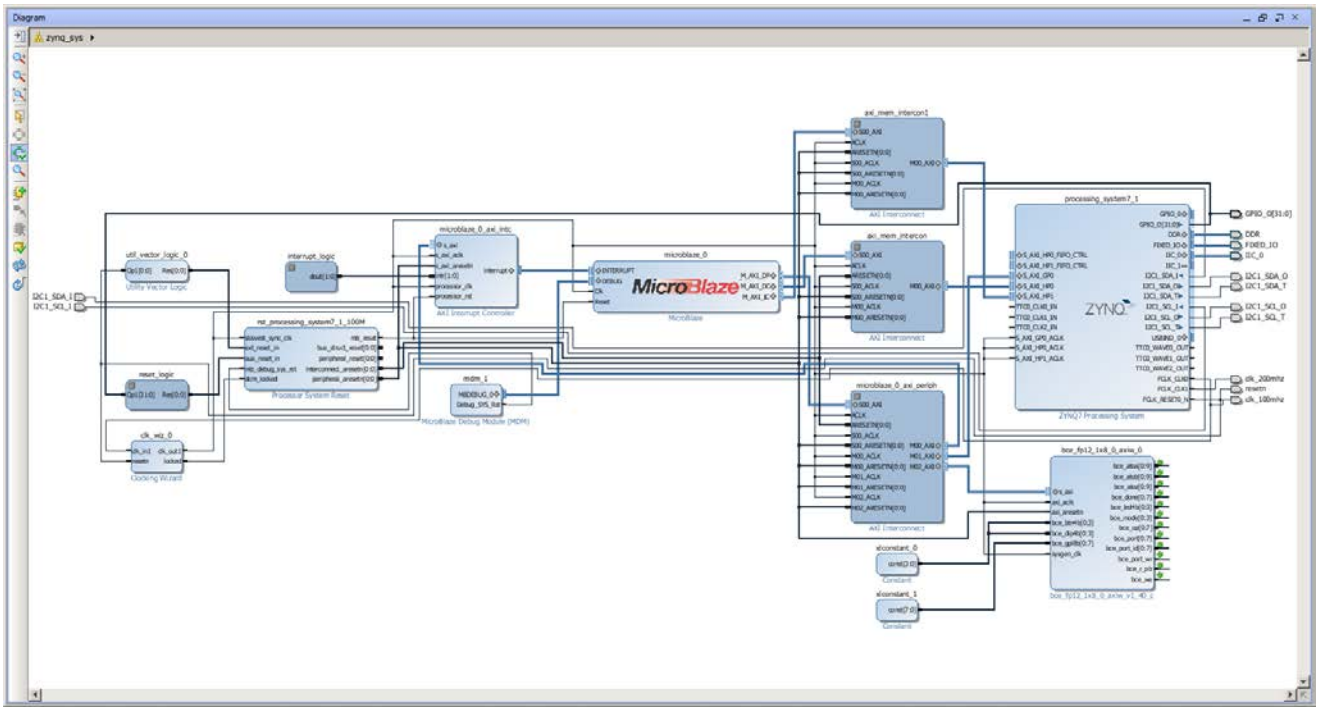

Figure 33: Evaluation design with single (1x8 SIMD) EdkDSP and ILA

The debug ports provide the basic visibility of the vector (8xSIMD) EdkDSP accelerator. Focus is on addresses and vector operation schedule. Concrete processed floating point data are not displayed. These data can be better analysed in the MicroBlaze debugger. MicroBlaze and its debugger can access all dual-ported memories of the (8xSIMD) EdkDSP accelerator.

Description of ILA debug ports of the (8xSIMD) EdkDSP accelerator IP (See Figure 3). All stored as 32k samples:

- 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 signals are used for the real-time analysis of the computation inside of the 8xSIMD vector processing unit of the EdkDSP accelerator IP ((See Figure 3). This helps with the debug of the coordination of the PicoBlaze6 firmware code, the vector processing unit together with MicroBlaze code.

3.8 Debug of Asymmetric Multiprocessing Demo with single EdkDSP accelerator with In-circuit Logic Analyser (ILA)

Start demo design with ILA from the Vivado 2015.2 SDK or use the precompiled SD card image as described above. Start the terminal. The AMP demo design is running.

It executes AMP demo SW on ARM with NEON unit, MicroBlaze with the single EdkDSP accelerator (with PicoBlaze6 reprogrammable firmware) and the ILA HW interface configured for debugging of the EdkDSP accelerator.

Start Vivado Lab Edition 2015.2 and select “Open Hardware Manager”. See Figure 34.

Figure 34: Vivado Lab Edition 2015.2

Select: **Open Target** See Figure 35. Take all defaults by clicking **Next** button in coming screens.
See Figure 36, Figure 37, Figure 38 and **Finish** in Figure 39.

The Vivado Lab Edition 2015.2 is at this stage connected to the debugged board jtag. See Figure 40.

Names and parameters of probes (see Figure 3) which can be captured by the ILA configuration on HW and visualised by Vivado Lab Edition 2015.2 are stored in file **debug_nets.ltx.**

In Vivado Lab Edition 2015.2 (see Figure 40) click on the "specify the probes file and refresh the device" link in the Trigger setup hw_ila_1 window.

Specify file
C:\VM_07\d_52\d_7z020_te7020-2\d_7z020_fp12_1x8\SDK_Workspace\hw_platform_0_ila\ debug_nets.ltx
See Figure 41.

This will add the names and parameters of probes (see Figure 3) to the ILA Waveform window. See Figure 43.
Use **+** to select probes used for triggering, and select the condition for trigger for each probe and their combination (use AND as default).

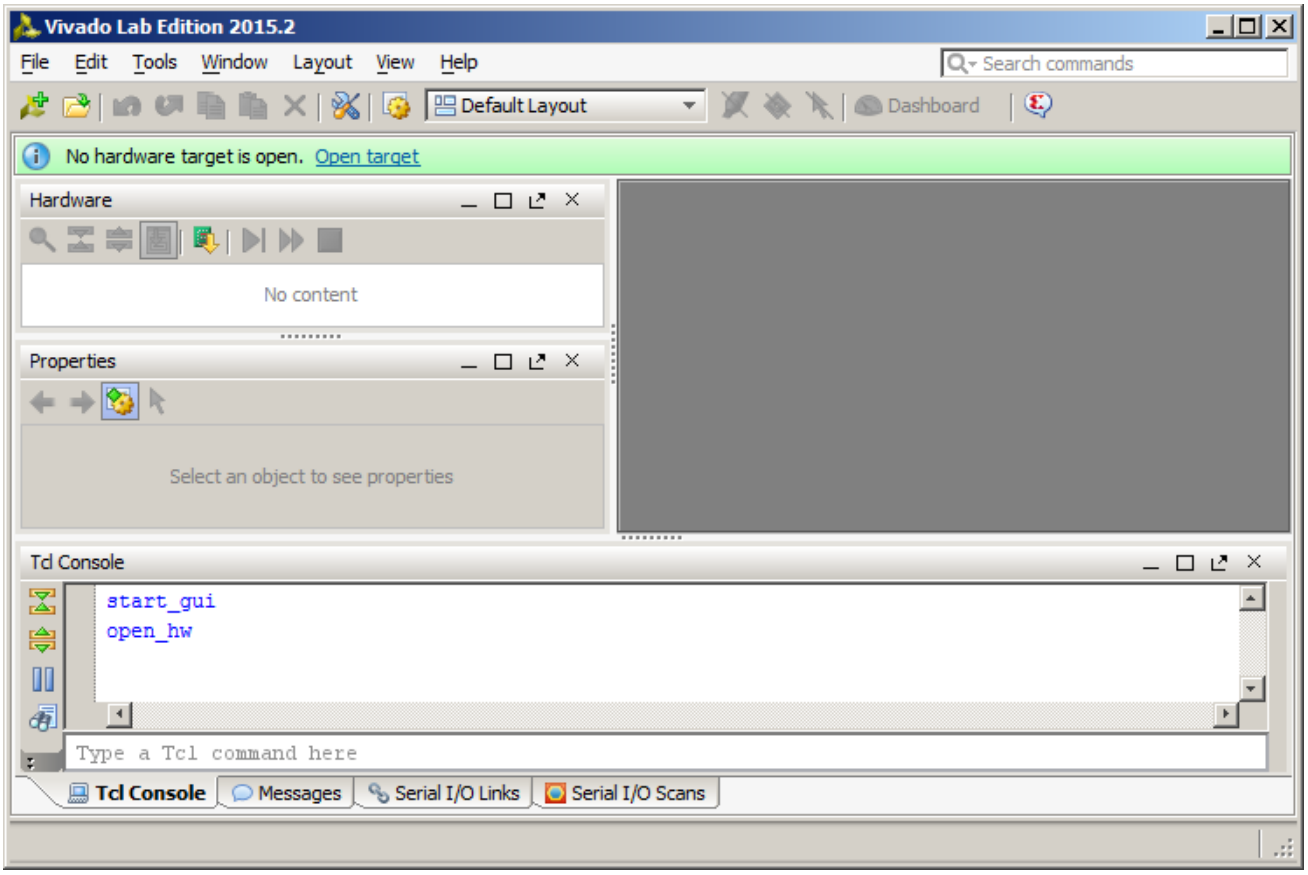

Figure 35: Select Open Target

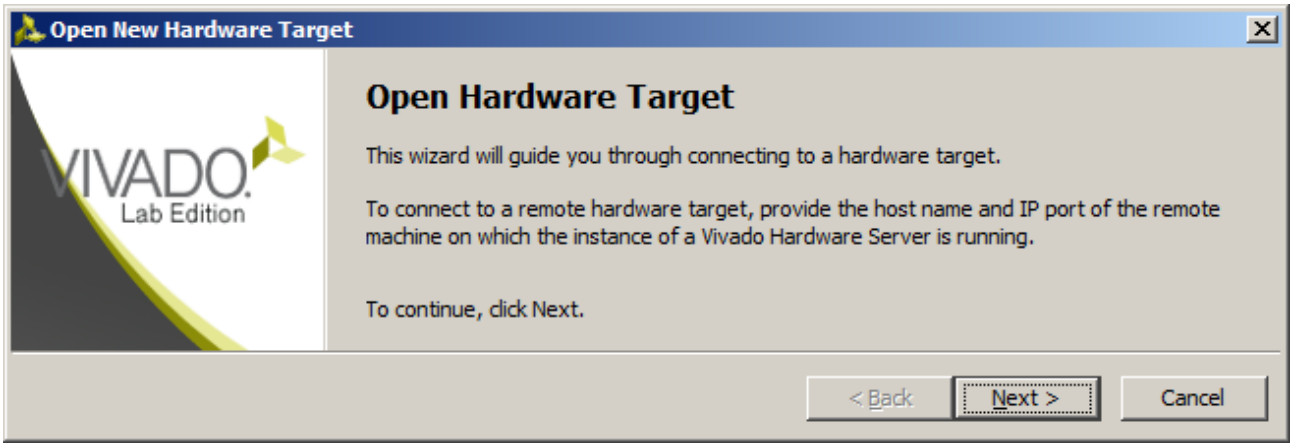

Figure 36: Open Hardware Target, next

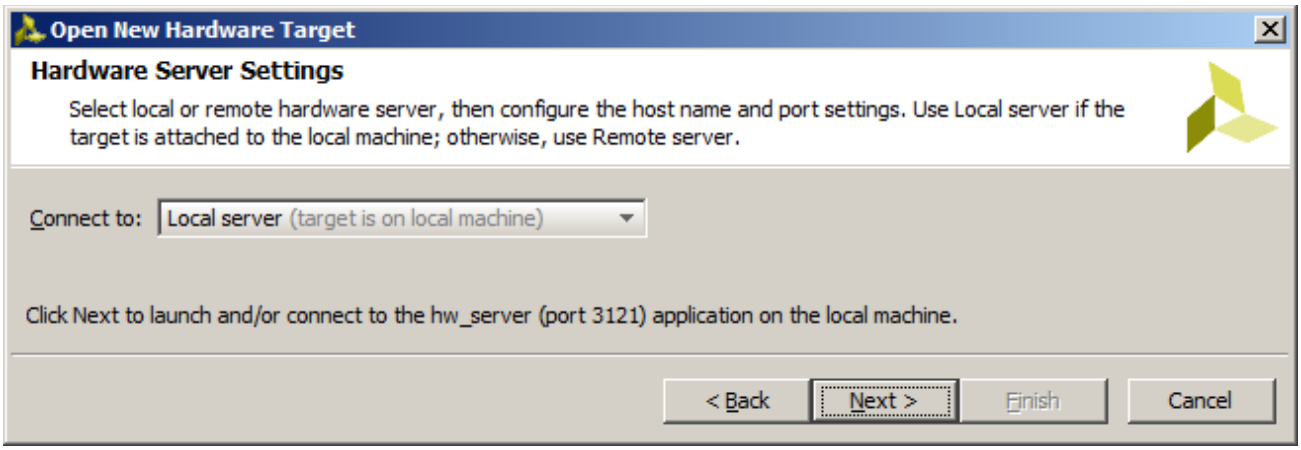

Figure 37: Local server default, next

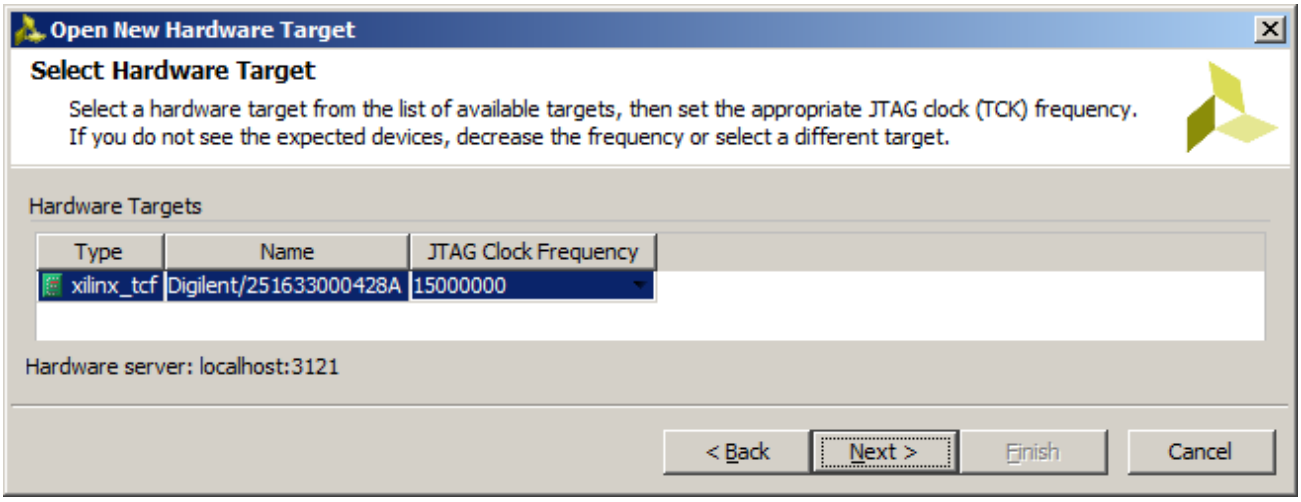

Figure 38: List of hardware targets, default, next

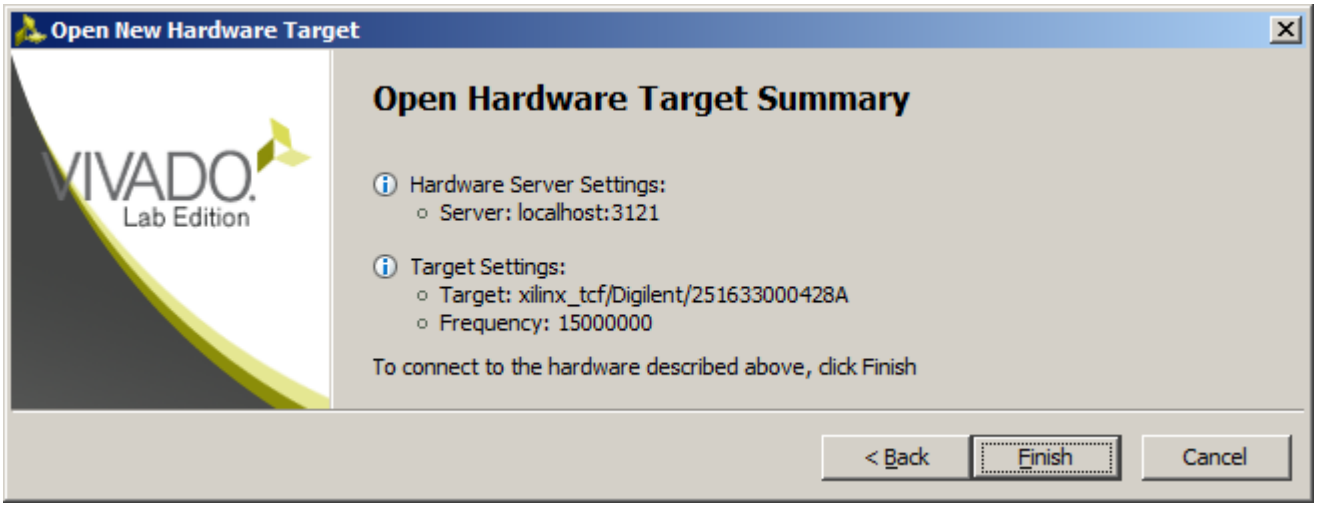

Figure 39: Summary, click Finish

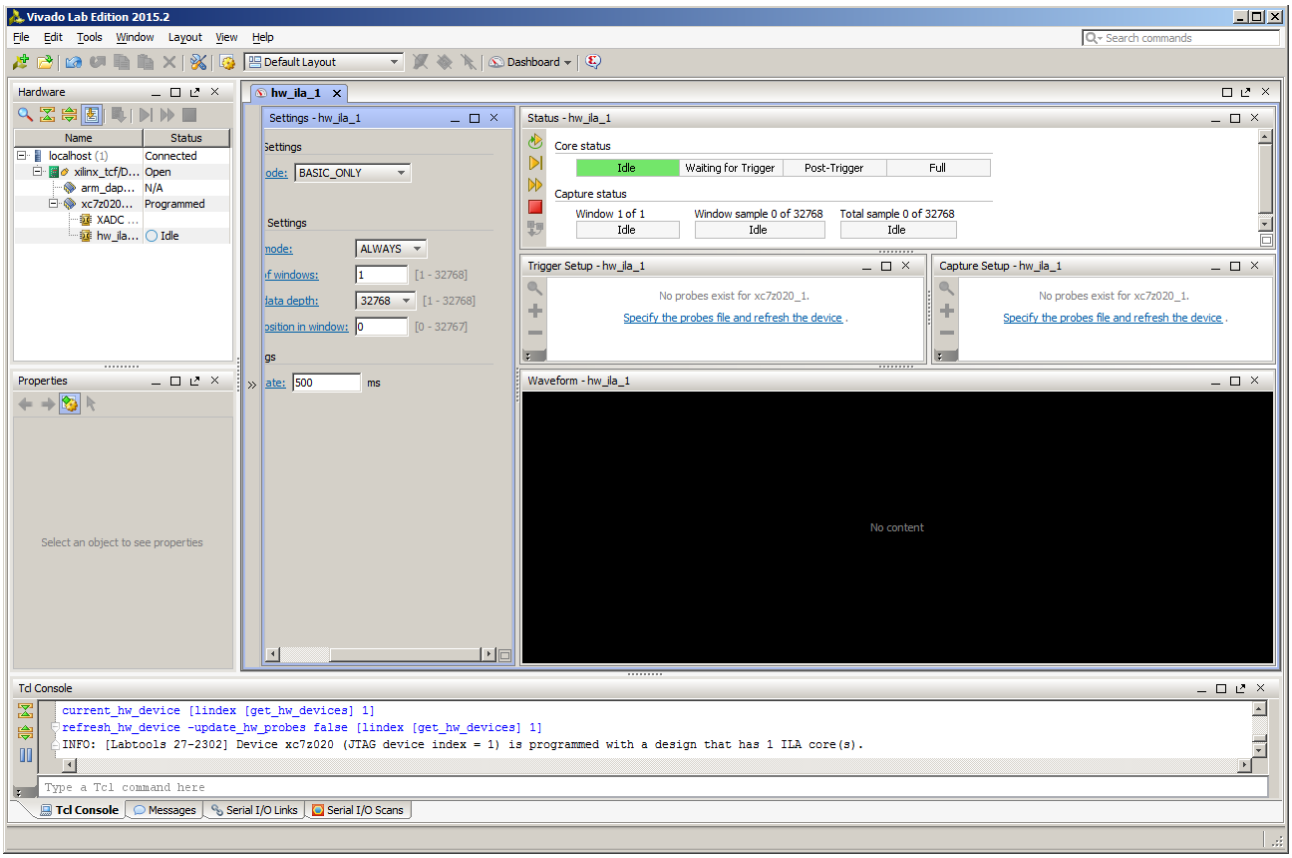

Figure 40: Vivado Lab edition is connected to the board via jtag (single shared USB cable)

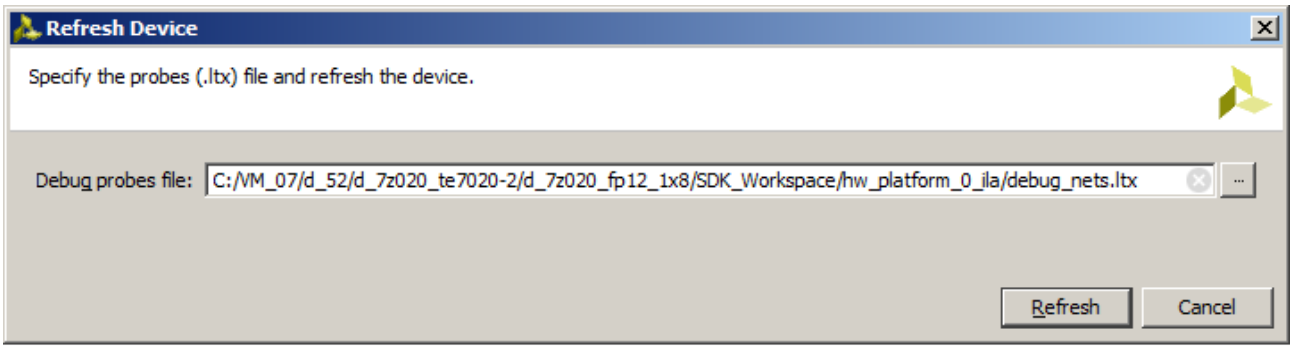

Figure 41: Select file with definition of probes present in HW

Some of probes used to trigger the capturing of data by ILA can be initiated and modified from the EdkDSP firmware running on the PicoBlaze6 running inside of the (8xSIMD) EdkDSP unit. This firmware can be modified and recompiled in the SDK 2015.2 as already described in this application note.

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. We will use it for selective triggering of the ILA in this specified point of computation of the EdkDSP accelerator.

File **fill_FA1124P1_program_store.h** contains firmware resulting from compilation of C source code **a_fp1124p1.c** . See Figure 42.

```
…
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);
        }
}
…
```

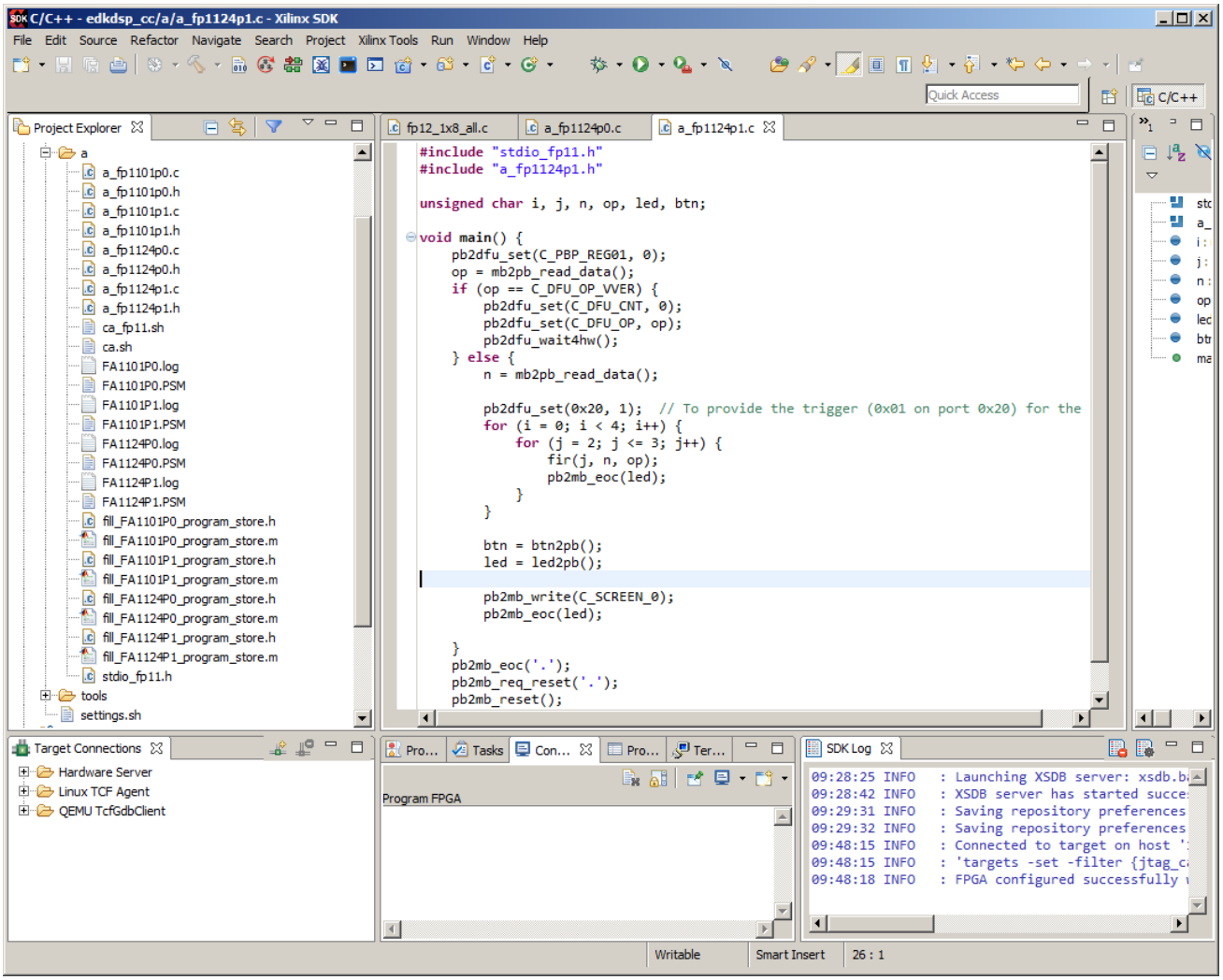

Figure 42: Compilation results of EdkDSP CC compiler. FIR filter code with probe trigger call

In Vivado Lab Edition, 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**)
See Figure 43.

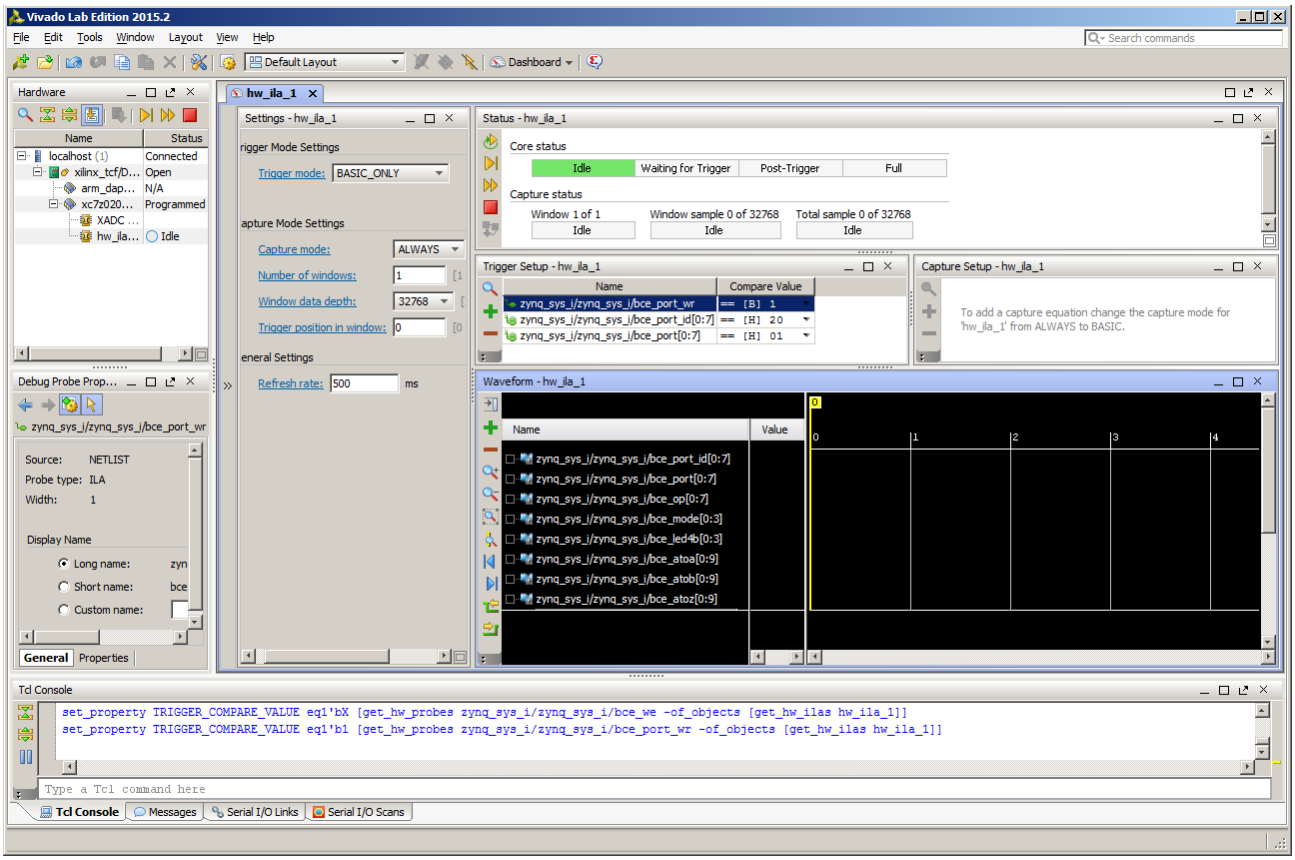

Figure 43: Trigger conditions. Selection in Vivado Lab Edition 2015.2

In Vivado Lab Edition 2015.2, arm the hw_ila_1 core by pressing **Run Trigger** button in **Hardware** window.

Armed hw_ila_1 core will wait until the running AMP design comes to the point, where PicoBlaze6 calls this dedicated function call **pb2dfu_set(0x20, 1);** as defined in FIR C code and the corresponding the PicoBlaze6 firmware. See Figure 42. ILA core will start to trigger 32K samples of all debug signals with the sampling rate 125 MHz. Data are captured and sent via jtag to Vivado Lab Edition 2015.2 for visualisation and analysis in the waveform window. Data capture the detailed trace of the initial 32k clock cycles of the FIR filter computation as defined by the SW fragment in Figure 42. See Figure 44. 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 306 clock cycles (125 MHz = 8ns clock period) to compute the vector product of two floating point vectors (coefficients and data), both with length 248*8=1984 elements and to update the data vector (circular buffer).

This demonstrates how the Vivado Lab Edition 2015.2 [12] provides (in combination with the HW ILA support instantiated in the design) sufficient level of visibility and debug capabilities for the developer of the (8xSIMD) EdkDSP firmware.

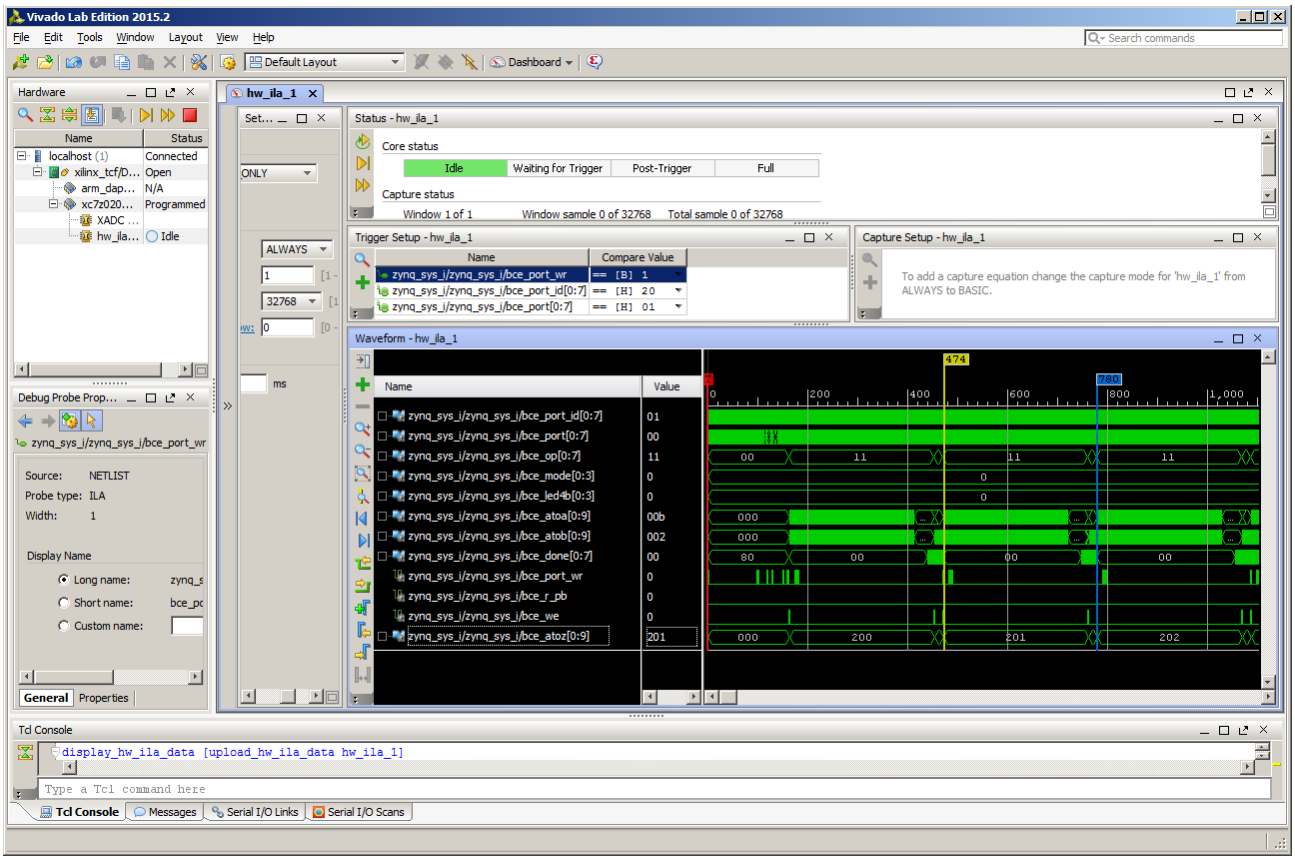

Figure 44: FIR filter waveforms after the trigger in Vivado Lab Edition 2015.2

While the AMP is running, we can modify the trigger condition and capture the initial phase of the LMS filter Running on the same EdkDSP accelerator in the next phase of the demo.

In SDK, see **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);
        }
}
…
```

The dedicated additional function is sending (0x00 on port 0x20) this time. In Vivado Lab Edition 2015.2 modify in the Trigger window the trigger condition for the same hw_ila_1 core to:

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

In Vivado Lab Edition 2015.2, arm the hw_ila_1 core again by pressing **Run Trigger** button in **Hardware** window. Armed hw_ila_1 core will wait until the running AMP design comes to the point, where PicoBlaze6 calls this dedicated function call pb2dfu_set(0x20, 0); as defined in LMS C code and executed by the corresponding the PicoBlaze6 controller inside of the EdkDSP accelerator. This will trigger new 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 45 and Figure 46).

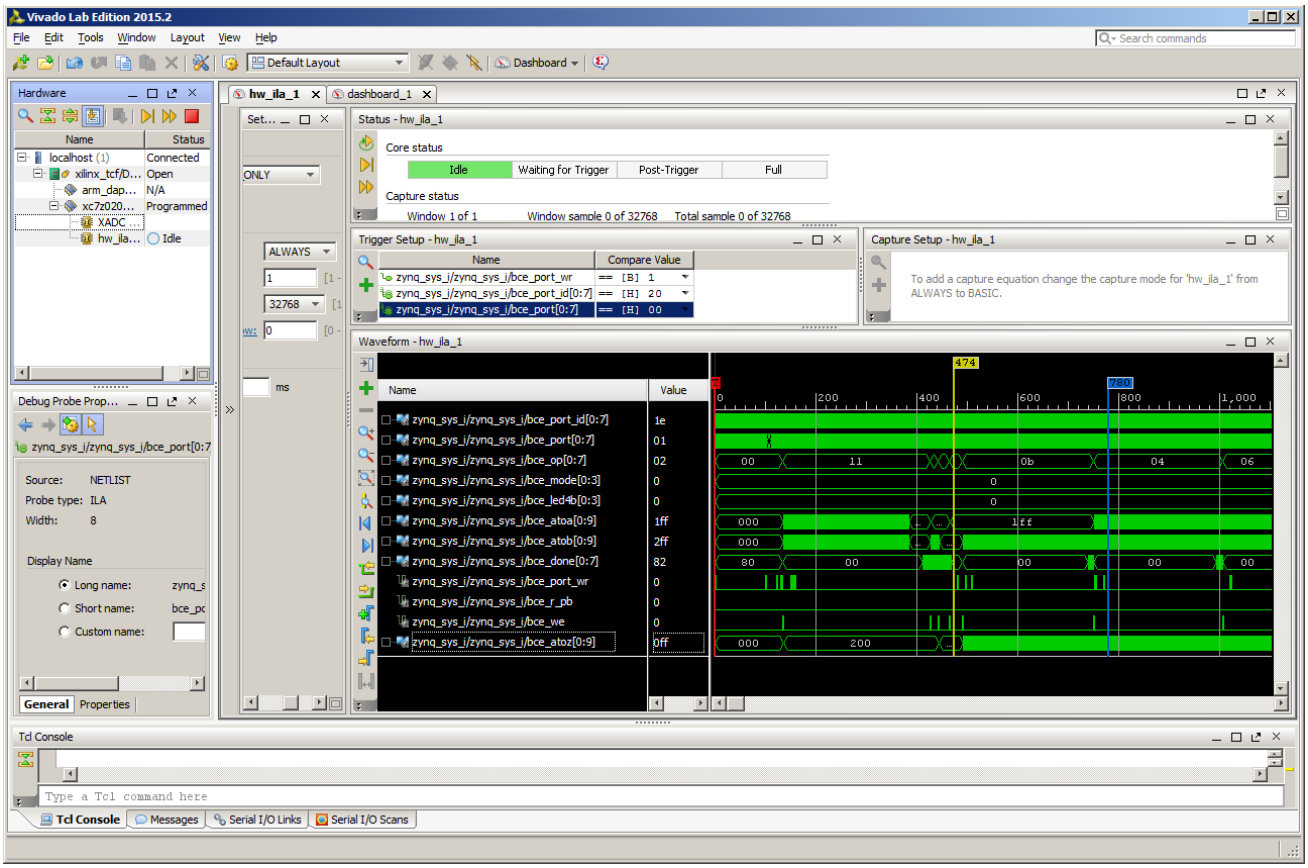

Figure 45: LMS filter waveforms after the trigger in Vivado Lab Edition 2015.2

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 1155 clock cycles (125 MHz = 8ns clock period) to compute the vector product of two floating point vectors (coefficients and data), both with length 248*8=1984 elements, update the data vector (circular buffer), compute the prediction error and adapt the coefficients of the LMS filter (see Figure 46).

In Figure 46, 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.

The ARM code and the MicroBlaze code can be compiled with –O0, … , -O3 optimisations and executed under both debuggers in combination with the ILA HW debug and visualisation in Vivado Lab Edition 2015.2. The –O0 option provides lower performance on ARM and MicroBlaze, but the corresponding binary code includes no transformations. This makes the co-debugging of the ARM and MicroBlaze C code easier.

The MicroBlaze debugger helps also in debugging of the interactions of the MicroBlaze with the EdkDSP accelerator. Blocks of exchanged floating point data can be inspected and verified with support of 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 can deliver bit-exact identical floating point results to the (8xSIMD) EdkDSP vector unit. This is used for verification of algorithms executed by the EdkDSP accelerator.

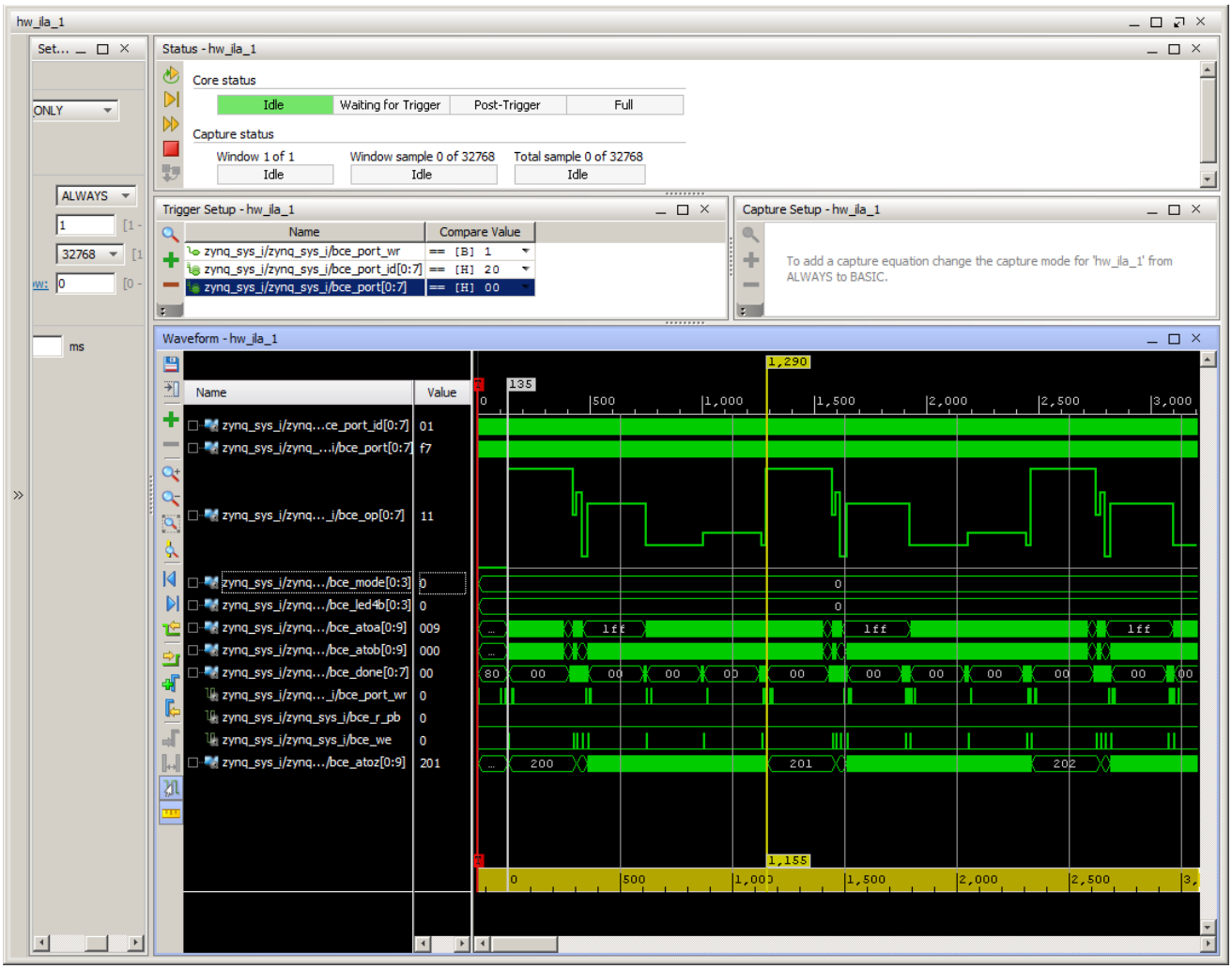

Figure 46: LMS filter waveforms in separate window and analog/hold format for bce_op

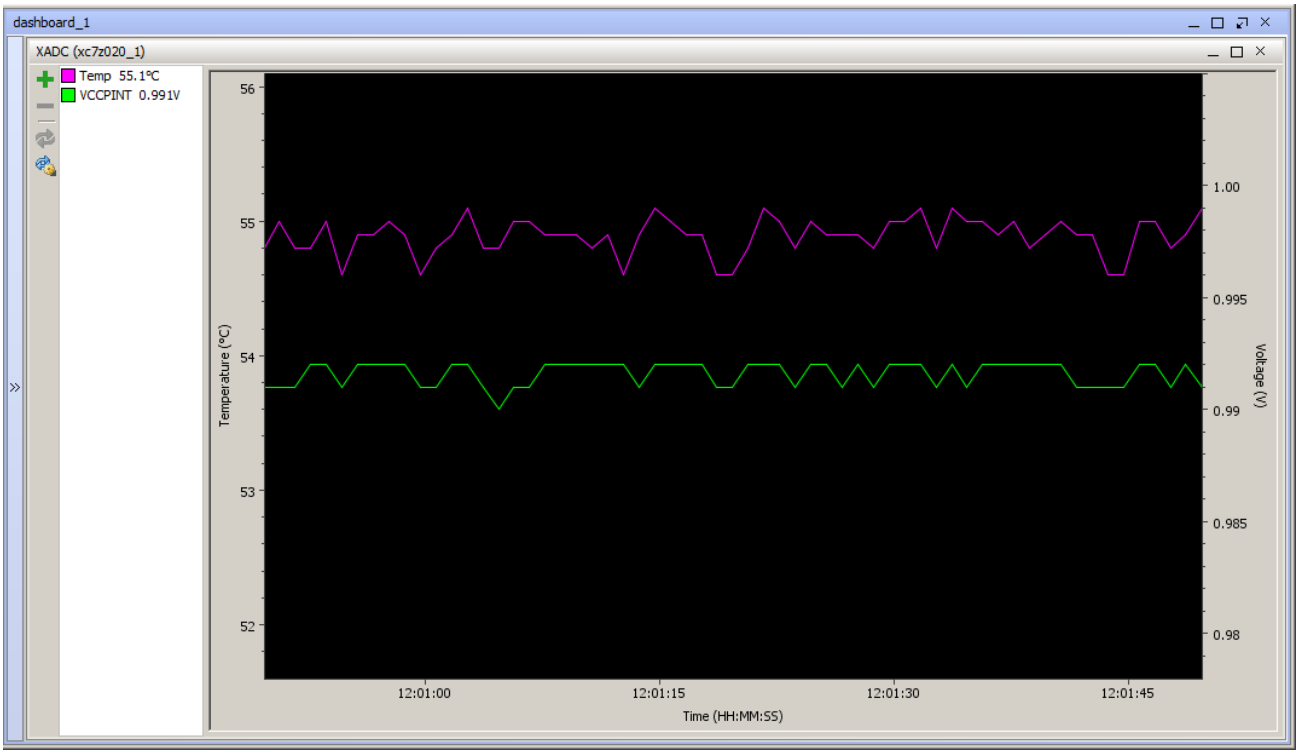

Figure 47: Separate dashboard with display of temperature and voltage in Vivado Lab Edition 2015.2

The Vivado Lab Edition 2015.2 jtag based server supports continuous download of additional signals measure in the ZYNQ fabric. Data can be opened in separate dashboard. See Figure 47.

The dashboard displays the temperature inside of the ZYNQ (round 55 degrees Celsius) and measures the voltage of the VCCPINT rail (round 0,99V) with sampling rate 0,5 sec.

These measurements run in the background and does not influence the HW and SW running on the monitored device.

4. References

[1] John McDougall: Simple AMP: Zynq SoC Cortex-A9 Bare-Metal System with MicroBlaze Processor, XAPP1093 (v1.0.1) January 24, 2014
http://www.xilinx.com/support/documentation/application_notes/xapp1093-amp-bare-metal-microblaze.pdf

[2] Trenz Electronic TE0720-02-2I Series (Z-7020)
http://www.trenz-electronic.de/products/fpga-boards/trenz-electronic/TE0720-02-2I-zynq.html

[3] TE0701 Carrier Board User Manual
https://wiki.trenz-electronic.de/display/4X5B/TE0701+Carrier+Board+User+Manual

[4] Zynq-7000 All Programmable SoC Technical Reference Manual UG585 (v1.8.1) September 19, 2014
http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf

[5] XST User Guide for Virtex-6, Spartan-6, and 7 Series Devices, UG687 (v 14.5) March 20, 2013.
http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/xst_v6s6.pdf

[6] Xilinx Software Development Kit Help Contents
http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_5/SDK_Doc/index.html

[7] 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, 201.
http://www.xilinx.com/support/documentation/ip_documentation/ug129.pdf

[8] EMC² – 'Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments' is an ARTEMIS Joint Undertaking project in the Innovation Pilot Programme 'Computing platforms for embedded systems' (AIPP5).
http://www.artemis-emc2.eu/

[9] VMware Workstation Player Documentation
https://www.vmware.com/support/pubs/player_pubs.html

[10] Vivado 2015.2 WebInstall for Windows 64
http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2015-2.html

[11] SDK 2015.2 Webinstall for Windows 64
http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/embedded-design-tools/2015-2.html

[12] Vivado Lab Edition - 2015.2 Full Product Installation
http://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2015-2.html

5. Evaluation version of the AMP demo on ZYNQ with (8xSIMD) EdkDSP accelerator IP. Designed in Vivado 2015.2.

The enclosed **Evaluation version of the AMP demo on ZYNQ with UTIA (8xSIMD) EdkDSP accelerator IP designed in Vivado 2015.2** can be downloaded from UTIA www pages free of charge and used for evaluation asymmetric multiprocessing on the TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3] together with four UTIA (8xSIMD) EdkDSP accelerators.

The evaluation package includes one DVD or the www download package with these deliverables:
Precompiled designs with UTIA (8xSIMD) EdkDSP accelerators for the TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board, compiled in Xilinx Vivado 2015.2 [10]. 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 2015.2 [11] SW projects related to the asymmetric multiprocessing on ZYNQ with source code for MicroBlaze processor and ARM processor. SW projects support the family of UTIA (8xSIMD) EdkDSP accelerators for the TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on TE701-05 carrier board [3].

The evaluation package includes this compiled library:

libwal.a	EdkDSP api (SDK 2015.2, MicroBlaze) for EdkDSP accelerators.

This library 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. 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 in VMware Workstation 12 Player.
edkdspcc	EdkDSP C compiler binary for Ubuntu in VMware Workstation 12 Player.
edkdspasm	EdkDSP ASM compiler binary for Ubuntu in VMware Workstation 12 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 precompiled designs. 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 TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3].

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 TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3].
without the need to install the UTIA compiler binaries and the Ubuntu image under the VMware Workstation 12 Player [9]. On email request to kadlec@utia.cas.cz , UTIA will send DVD with the Ubuntu image for the VMware Workstation 12 Player [10] free of charge.

6. AMP projects with evaluation version of (8xSIMD) EdkDSP accelerator IP for the Artemis EMC2 project partners.

The **release version of the AMP HW/SW Vivado 2015.2 projects for** the **TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3] with evaluation version of the (8xSIMD) EdkDSP accelerator IP for the partners in the Artemis EMC2 project [8]** 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 EMC2 project partners) 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 EMC2 project partner (by email) and by the standard mail the invoice for:

Release version of the AMP HW/SW Vivado 2015.2 projects on ZYNQ with the evaluation version of the UTIA (8xSIMD) EdkDSP accelerator cores for partners in the Artemis EMC2 project (without VAT) 0,00 Eur

The package includes this application note and the EdkDSP DVD with these deliverables:
Precompiled designs with UTIA (8xSIMD) EdkDSP accelerators for the TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3], compiled in Xilinx Vivado 2015.2 [10]. 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 **release version of the AMP HW/SW projects on ZYNQ with the evaluation version of the UTIA (8xSIMD) EdkDSP accelerator cores for the Artemis EMC2 project partners** includes source code of Vivado 2015.2 design projects demonstrating the asymmetric processing on ZYNQ and the evaluation versions of the UTIA (8xSIMD) EdkDSP accelerators provided in form of netlisted pcores compiled by Xilinx VIVADO 2015.2 [10]:

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. **Partners in the Artemis EMC2 project [8]** have nonexclusive, non-transferable license from UTIA to integrate these evaluation netlists into their own Vivado 2015.2 [10] designs and to compile them to unlimited number of bit-streams for the asymmetric multiprocessing designs on Xilinx ZYNQ 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 Artemis EMC2 project partners.

The package for the Artemis EMC2 project partners includes the SDK 2015.2 [11] 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 TE0720-02-2IF, TE0720-02-1CF and automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3].

The package for the Artemis EMC2 project partners includes the library:

libwal.a	EdkDSP api (SDK 2015.2, MicroBlaze) for EdkDSP accelerators.

This library 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. Source code of this library is owned by UTIA and it is not provided in this evaluation package.

The package for the Artemis EMC2 project partners includes these binary applications for Ubuntu:

edkdsppp	EdkDSP C pre-processor binary for Ubuntu in VMware Workstation 12 Player.
edkdspcc	EdkDSP C compiler binary for Ubuntu in VMware Workstation 12 Player.
edkdspasm	EdkDSP ASM compiler binary for Ubuntu in VMware Workstation 12 Player.

These binary applications have no time restriction. The Artemis EMC2 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 UTIA EdkDSP accelerators in precompiled designs. Source code of these binaries 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 TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3].

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 TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on TE701-05 carrier board [3] without the need to install the UTIA compiler binaries and the Ubuntu (x86 PC) OS image under the VMware Workstation 12 Player [9].

The release package deliverables also includes two DVDs with the Ubuntu (x86 PC) image for the VMware Workstation 12 Player (free of charge). This image is provided to ease the installation of the UTIA EdkDSP C compiler on Windows 7 64bit in VMware Workstation 12 Player [9].

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. AMP release version on ZYNQ with (8xSIMD) EdkDSP accelerator IP. Designs in Vivado 2015.2.

The **release version of the AMP demo on ZYNQ with UTIA (8xSIMD) EdkDSP package for designs in Vivado 2015.2** 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 DVD 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 the AMP demo on ZYNQ with UTIA (8xSIMD) EdkDSP package for designs in Vivado 2015.2 (without VAT) 400,00 Eur

The release package includes this application note and the EdkDSP DVD with these deliverables:
Precompiled designs with UTIA (8xSIMD) EdkDSP accelerators on the TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3], compiled in Xilinx Vivado 2015.2 [10]. 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 without limitations of the evaluation package.

The release package includes source code of Vivado 2015.2 [10] design projects demonstrating the asymmetric processing on ZYNQ. The UTIA (8xSIMD) EdkDSP accelerator IP cores are provided in the form of netlist pcores generated for Xilinx VIVADO 2015.2 [10]:

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 (8xSIMD) 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 2015.2 [10] designs and to compile them to unlimited number of bit-streams for designs on Xilinx ZYNQ FPGAs. This nonexclusive, non-transferable license has no time restriction. The source code of the (8xSIMD) 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 2015.2 [10] SW projects in source code for MicroBlaze as described in this application note. Projects support the family of UTIA (8xSIMD) EdkDSP accelerators on the TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3].

The release package includes the library:

libwal.a	EdkDSP api (SDK 2015.2, MicroBlaze) for EdkDSP accelerators.

This library 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. 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 in VMware Workstation 12 Player.
edkdspcc	EdkDSP C compiler binary for Ubuntu in VMware Workstation 12 Player.
edkdspasm	EdkDSP ASM compiler binary for Ubuntu in VMware Workstation 12 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. 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 TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3].

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 on the TE0720-02-2IF, TE0720-02-1CF and the automotive TE0720-02-1QF system-on-module [2] on the TE701-05 carrier board [3] without the need to install UTIA compiler binaries and the Ubuntu under the VMware Workstation 12 Player [9].

The release package deliverables also includes DVD with the Ubuntu (x86 PC) image for the VMware Workstation 12 Player [9]. This Ubuntu image is provided by UTIA (free of charge) to ease the installation of the UTIA EdkDSP C compiler on Windows 7 64 bit in the VMware Workstation 12 Player [9].

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.