Application Note

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

Asymmetric Multiprocessing on ZYNQ with EdkDSP accelerators on Xilinx ZC702 board - ISE 14.5

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

Revision history:

Rev.	Date	Author	Description
1	17.10.2014	Jiří Kadlec	First draft
2	20.10.2014	Jiří Kadlec	Internal UTIA draft
3	21.10.2014	Jiří Kadlec	Release for download from UTIA www pages
4	05.11.2014	Jiří Kadlec	Fixed axiw names in the description

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

1. Summary

1.1 Key features

This application note describes the asymmetric multiprocessing design (AMP) based on the Xilinx application note XAPP1093 [1]. 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 the Xilinx ZC702 designs with AXI bus. This application note explains how to install and use the demonstrator on Windows7, (32 or 64 bit) and the Xilinx ZC702 board [2],[3], [4]. 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 2000 coefficients in single precision floating point arithmetic with sustained performance
    - 627 MFLOP/s on the 100 MHz EdkDSP
    - 146 MFLOP/s on the 666 MHz ARM Cortex A9 (with the vector floating point unit)
    - 5 MFLOP/s on the 100 MHz MicroBlaze processor with the floating point HW unit
- The EdkDSP accelerators can be reprogrammed by the firmware. The programming is possible in C with the use of the UTIA EDKDSP C compiler. Accelerators can be programmed with two firmware programs. Designs can swap in the real time the firmware in only few clock cycles in the runtime.
- 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 firmware for the FIR filter room response to the firmware for adaptive LMS identification of the filter coefficients in the acoustic noise cancellation demo.
- The EdkDSP accelerator is providing single-precision floating point results bit-exact identical to the reference software implementation running on MicroBlaze with the Xilinx HW single precision floating point unit.
- The 100 MHz 8xSIMD EdkDSP accelerator is 4,3x faster than the 666 MHz ARM Cortex A9 (with the vector processing unit) and 125x faster than computation on 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 100 MHz (8xSIMD) EdkDSP accelerator with the floating point performance 994 MFLOP/s. The peak performance (only theoretical) of the single 100 MHz (8xSIMD) EdkDSP accelerator is 1,6 GFLOP/s.
- The peak performance of four 100 MHz (8xSIMD) EdkDSP accelerators implemented in this demo design is 6,4 GFLOP/s (this is only theoretical, peek figure).
- This evaluation package presents two (8xSIMD) EdkDSP accelerator families: one family without pipelined floating point divider data path and one family with a single pipelined floating point divider data path. The members of both families differ by size and by supported vector floating point operations.
- The floating point applications can be scheduled inside of the EdkDSP accelerator by the Xilinx PicoBlaze6 processor [8]. Each firmware program has maximal size of 4096 (18 bit wide words).

1.2 What is included

The asymmetric multiprocessing on ZYNQ (AMP) with the EdkDSP platform evaluation package contains these deliverables for the Windows 7 (32 or 64bit):

- 10 evaluation versions of AMP designs. Each design contains one used ARM Cortex A9 processor core, one MicroBlaze and four instances of the EdkDSP accelerators with 8xSIMD floating point data paths with AXI-lite bus. (ARM 666 MHz, MicroBlaze 100 MHz, Accelerators 100 MHz) Designs are compiled in Xilinx XPS 14.5 and XST [6].
- UTIA is providing source code for the demo applications and SW projects for the Xilinx SDK 14.5 [7]. These source code projects are compiled with the UTIA library libwal.a serving for the EdkDSP communication.
- The included evaluation versions of the UTIA EdkDSP accelerators have HW limitation of maximal number of performed vector operations.
- The UTIA EdkDSPC C compiler is provided as 3 executable applications for Ubuntu in the VMware Player.
- 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 HW design projects with the evaluation versions of the EdkDSP accelerators in form of AXI netlist pcores for free. See chapter 6 for specification of deliverables for the EMC2 project partners and license details.

- Release versions of AMP designs with the EdkDSP package for the Xilinx ZC702 board is offered by UTIA.
  All customers can order and buy from UTIA the release version of this AMP demo. It includes the HW design projects with the EdkDSP accelerators in form of AXI netlist pcores with main limitations removed. See sections 7 of this application note for specification of deliverables and license details.

2. Demonstrator AMP with EdkDSP accelerators on ZC702 board

2.1 Description of EdkDSP accelerators and evaluation designs

This application note describes how to set-up and use the collection of 10 HW designs running in an asymmetric multiprocessing architecture formed by of one ARM processor and one MicroBlaze processor with 4 (8xSIMD) EdkDSP accelerators on Xilinx ZC702 board. See Figure 1 and Figure 2. The demonstrator serves to evaluation of parameters of two floating point accelerator families in the AMP architecture on the Xilinx ZYNQ xc7z020-1 part:

- **bce_fp11_1x8_0_axiw_v1_[10|20|30|40]_a** is a family of four versions of floating point EdkDSP accelerators with 8 SIMD data paths.
- **bce_fp12_1x8_0_axiw_v1_[10|20|30|40]_a** is similar family of four versions of floating point EdkDSP accelerators with 8 SIMD data paths extended by a pipelined floating point division (FPDIV) in a single data path.

The four grades [10|20|30|40] of the EdkDSP accelerator differ in HW-supported vector computing capabilities:

The area optimized accelerators **bce_fp11_1x8_0_axiw_v1_10_a** and **bce_fp12_1x8_0_axiw_v1_10_a** perform vector floating point operations FPADD, FPSUB in 8 SIMD data paths.

The accelerators **bce_fp11_1x8_0_axiw_v1_20_a** and **bce_fp12_1x8_0_axiw_v1_20_a** perform vector floating point operations FPADD, FPSUB in 8 SIMD data paths plus the vector floating point MAC operations in 8 SMD data paths for length of the vector 1 up to 10. These accelerators can be used in applications like floating point matrix multiplication with row and column dimensions <= 10.

The accelerators **bce_fp11_1x8_0_axiw_v1_30_a** and **bce_fp12_1x8_0_axiw_v1_30_a** support identical operations as the bce_fp11_1x8_0_axiw_v1_20_a and bce_fp12_1x8_0_axiw_v1_20_a plus 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 255 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 255).

Finally, the accelerators **bce_fp11_1x8_0_axiw_v1_40_a** and **bce_fp12_1x8_0_axiw_v1_40_a** support identical operations as the bce_fp11_1x8_0_axiw_v1_30_a and bce_fp12_1x8_0_axiw_v1_30_a plus an additional HW support of dot product. It is computed in 8 data paths with the HW supported wind-up into single scalar result.

The **bce_fp11** versions of 8xSIMD accelerators has no support for pipelined vector floating point division and it is suitable for applications like FIR filters or adaptive LMS filters with no need for floating point division.

The **bce_fp12** versions of 8xSIMD accelerators are larger in comparison to the bce_fp11 equivalents and support in a single data path the pipelined vector floating point division. 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.

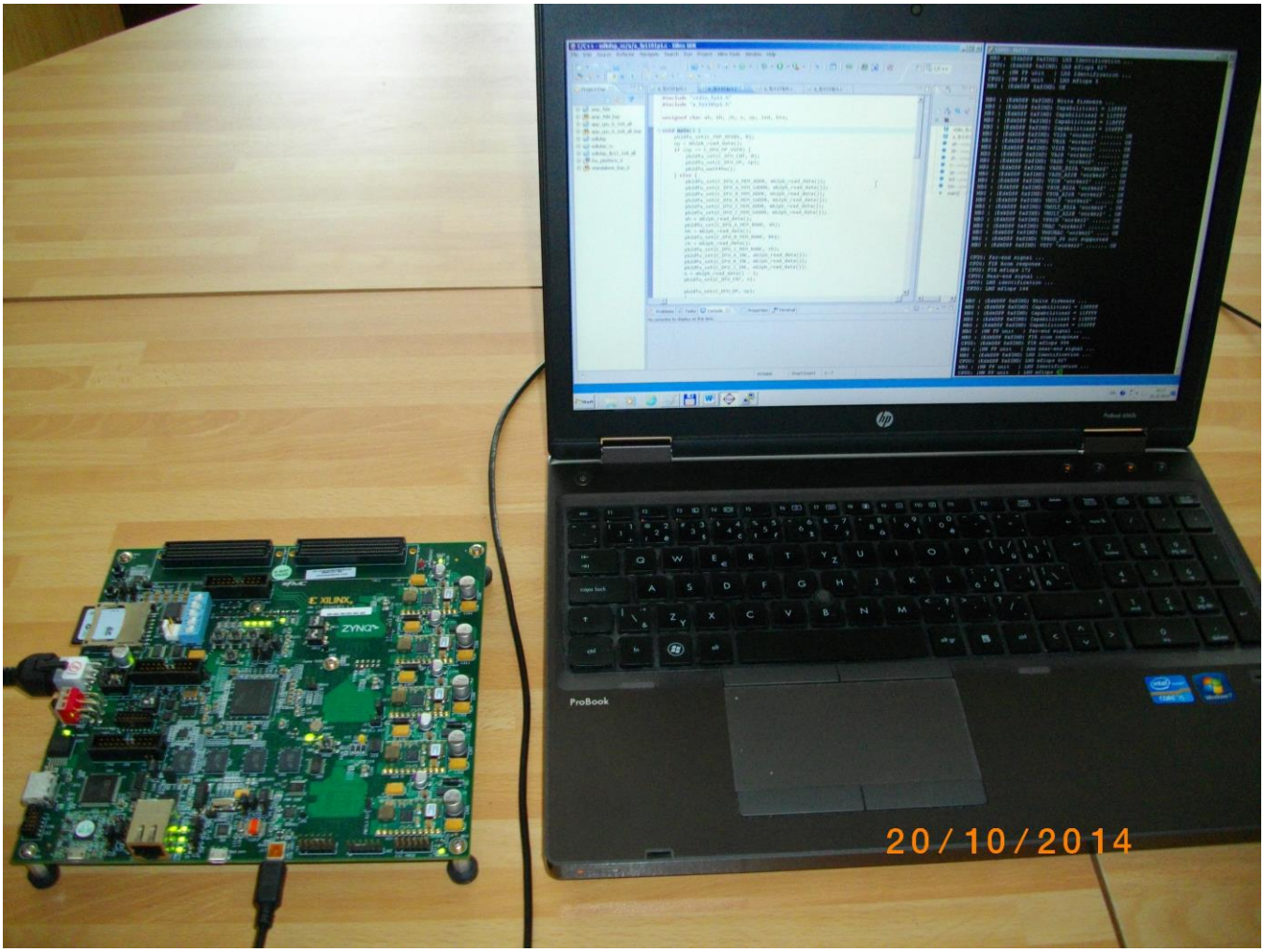

Figure 1: Asymmetric multiprocessing on ZYNQ with ARM, MicroBlaze and 4x (8xSIMD) EdkDSP floating point accelerators on Xilinx ZC702 board;

Ten precompiled HW designs combine ARM Cortex A9 with MicroBlaze and four 8xSIMD EdkDSP accelerators. All designs demonstrate use of single instance of 8xSIMD EdkDSP floating point accelerator on 32bit AXI-lite bus of the Xilinx MicroBlaze soft-core processor on the Xilinx ZYNQ ZC702 FPGA board with system clock of ARM 666 MHz, MicroBlaze 100 MHz and EdkDSP accelerators 100 MHz. See Figure 2.

Common properties of precompiled evaluation designs:

- The EdkDSP floating point accelerators are reconfigurable during runtime by change of firmware.
- Asymmetric multiprocessing of ARM Cortex A9 and MicroBlaze system with shared external DDR3.
- All demos (HW/SW) have been designed in Xilinx EDK/ISE 14.5 tools [6].

Presented HW accelerators can results in better POWER per MFLOPS ratio for certain class of DSP applications in comparison to the computation on standard CPUs with standard HW floating point support.

The demonstrator includes source code of set of SW demos prepared for easy import of projects and compilation in the Xilinx SDK 14.5 [7].

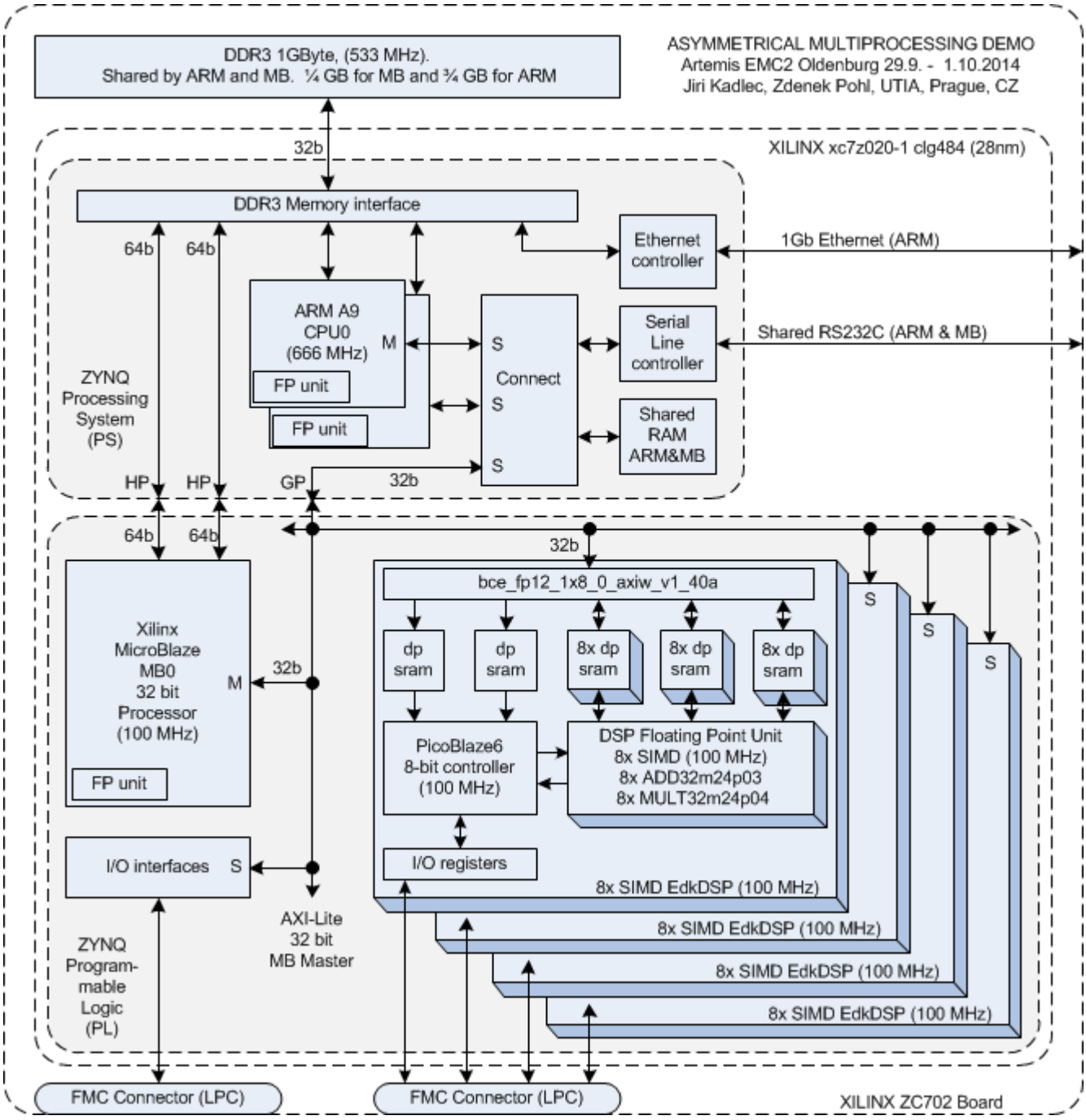

Figure 2: Simplified architecture of AMP with EdkDSP evaluation designs; Xilinx ZC702 board;

2.2 Resources used by the designs

The resources used by the 10 presented designs are summarised in Figure 3 and Figure 4.

7z020-1c	fp	fp	fp	fp	fp	Design size				Performance	
	Add Mul	Mac	Dot Prod	S8 Prod	div	ff %	Lut %	Bram no(of)	Slices %	LMS Mflop/s	FIR Mflop/s
(4x) fp11_1x8_10	8x					20	58	127(140)	74		
(4x) fp11_1x8_20	8x	8x				24	66	127(140)	81		
(4x) fp11_1x8_30	8x	8x	8x			28	78	127(140)	93		
(4x) fp11_1x8_40	8x	8x	8x	1x		28	78	127(140)	93	**(4x) 627**	**(4x) 994**
mix fp11_1x8_10 fp11_1x8_20 fp11_1x8_30 fp11_1x8_40	 8x 8x 8x 8x	 8x 8x 8x	 8x 8x	 1x	 	25	70	127(140)	85	**627**	**994**

Figure 3: AMP designs on ZYNQ, ARM A9, MicroBlaze and 4x (8xSIMD) EdkDSP, no FP division

7z020-1c	fp	fp	fp	fp	fp	Design size				Performance	
	Add Mul	Mac	Dot Prod	S8 Prod	div	ff %	Lut %	Bram no(of)	Slices %	LMS Mflop/s	FIR Mflop/s
(4x) fp12_1x8_10	8x				1x	22	64	127(140)	80		
(4x) fp12_1x8_20	8x	8x			1x	26	72	127(140)	87		
(4x) fp12_1x8_30	8x	8x	8x		1x	30	84	127(140)	95		
(4x) fp12_1x8_40	8x	8x	8x	1x	1x	30	86	127(140)	97	**(4x) 627**	**(4x) 994**
mix fp12_1x8_10 fp12_1x8_20 fp12_1x8_30 fp12_1x8_40	 8x 8x 8x 8x	 8x 8x 8x	 8x 8x	 1x	 1x 1x 1x 1x	27	76	127(140)	91	**627**	**994**

Figure 4: 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 Xilinx ZC702 development board [2], [3], [4]. See Figure 1. It is using the 1GB DDR3 memory with clock signal 533 MHz. The DDR3 is connected to Xilinx ZYNQ xc7z020-1 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 APP demo is following the Xilinx aapplication note XAPP1093 [1]. See Figure 2 for the architecture of the design.

2.4 Re-programmability of EdkDSP accelerators

Each of the four 100 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 994 MFLOP/s in case of 2000 tap FIR filter computation and 627 MFLOP/s in case of the adaptive 2000 tap LMS filter identification demo. Each design in this package has four 8xSIMD EdkDSP units.

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.

Each 8xSIMD EdkDSP accelerator works with 2 program memories, each with the 4096 (18bit wide) words. Both program memories are accessible by 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 2.

ARM and MicroBlaze can be both debugged simultaneously from the SDK 14.5 debuggers integrated in the Xilinx SDK tool. The debugging of both processors is using the Xilinx application note XAPP1093 [1]. The PicoBlaze6 processors [8] 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 side of the dual-ported 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 in the (8xSIMD) EdkDSP units can overlap with the communication to and from the DDR3 performed by MicroBlaze and supported by data and program cache. A Ping-Pong swap of memory banks is used. The 8xSIMD EdkDSP firmware is computing (in parallel) in some banks of all dual ported memories and MicroBlaze is communicating (sequentially) to/from DDR3 in another set of banks of the dual-ported memories. This process can be stopped, inspected and debugged by the MicroBlaze debugger from the SDK.

3. Installation of AMP with EdkDSP platform on the ZC702 board

3.1 Import of precompiled projects and SW into Xilinx SDK 14.5

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_145_7z

```
19.10.2014  10:24    <DIR>          .
19.10.2014  10:24    <DIR>          ..
19.10.2014  10:23    <DIR>          d_7z020_fp11_4x8
19.10.2014  10:23    <DIR>          d_7z020_fp11_4x8_IMPORT
19.10.2014  10:23    <DIR>          d_7z020_fp11_4x8_v1_10a
19.10.2014  10:23    <DIR>          d_7z020_fp11_4x8_v1_20a
19.10.2014  10:23    <DIR>          d_7z020_fp11_4x8_v1_30a
19.10.2014  10:23    <DIR>          d_7z020_fp11_4x8_v1_40a
19.10.2014  10:23    <DIR>          d_7z020_fp11_4x8_v1_mx1
19.10.2014  10:23    <DIR>          d_7z020_fp12_4x8
19.10.2014  10:24    <DIR>          d_7z020_fp12_4x8_IMPORT
19.10.2014  10:24    <DIR>          d_7z020_fp12_4x8_v1_10a
19.10.2014  10:24    <DIR>          d_7z020_fp12_4x8_v1_20a
19.10.2014  10:24    <DIR>          d_7z020_fp12_4x8_v1_30a
19.10.2014  10:24    <DIR>          d_7z020_fp12_4x8_v1_40a
19.10.2014  10:24    <DIR>          d_7z020_fp12_4x8_v1_mx1
```

Select SDK 14.5 workspace in c:\VM_07\d_145_7z \d_7z020_fp12_4x8\SDK_Workspace. See Figure 5.

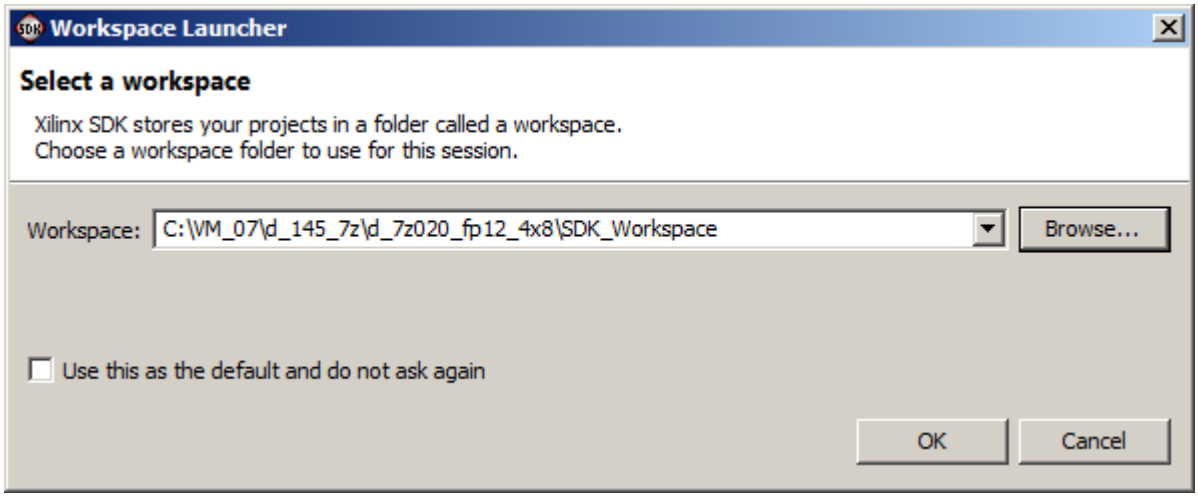

Figure 5: Select the SDK Workspace

Add **c:\VM_07\d_145_7z\d_7z020_fp12_4x8\repo_edkdsp** path to the UTIA EdkDSP repository.
See Figure 6.

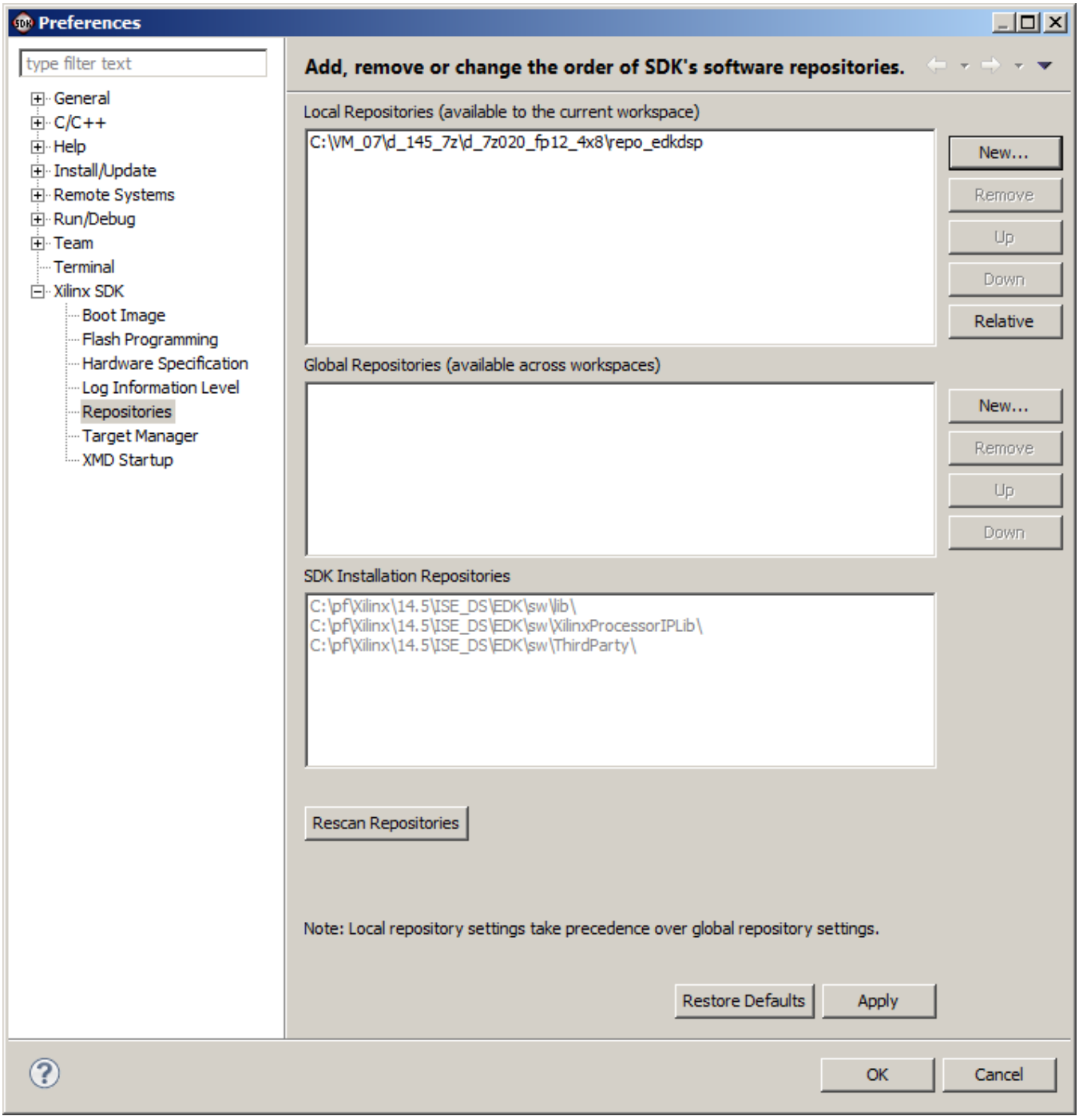

Figure 6: Include the UTIA EdkDSP Repository

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.

In SDK, select File -> New -> Project … -> Xilinx -> Hardware Platform Specification. See Figure 7. Click on the Next button.

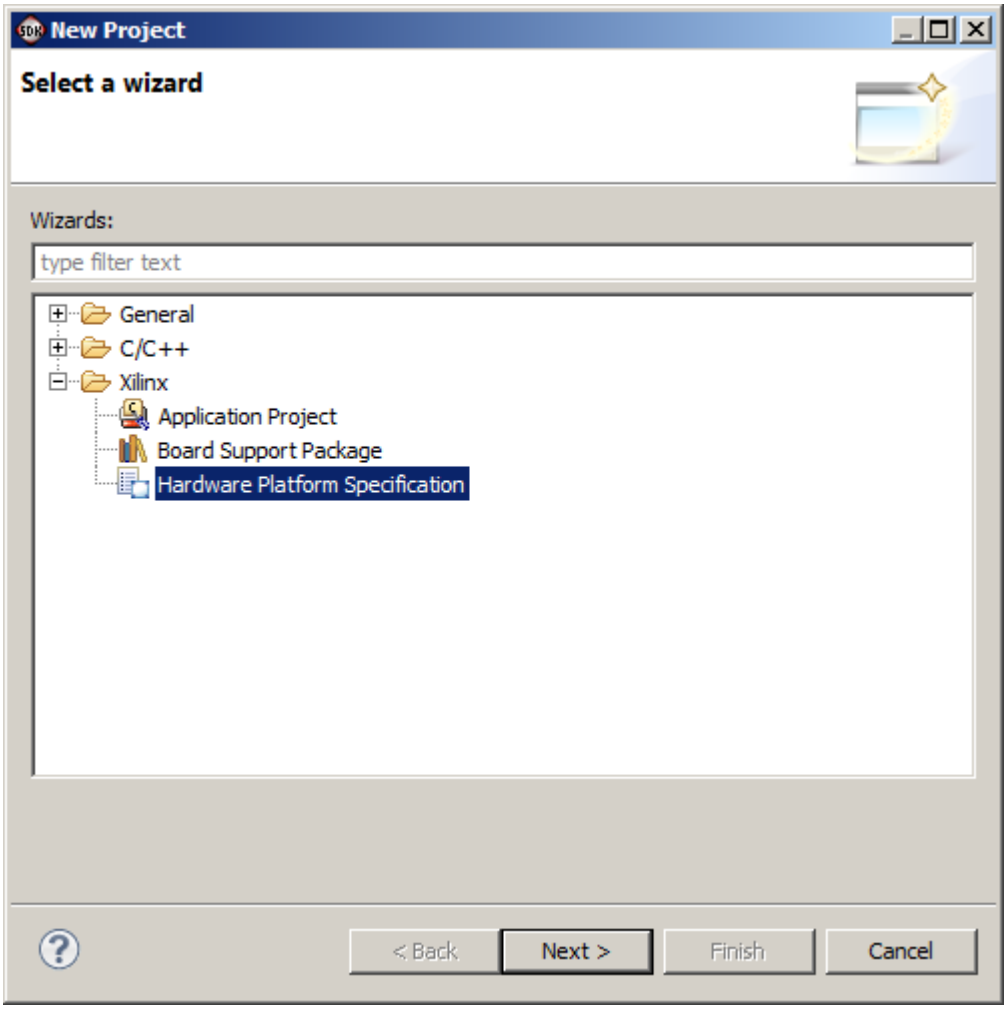

Figure 7: Specify the Hardware Platform

In the "New Hardware Project" screen, fill into the Project name:
hw_platform_0

In the New Hardware Project screen, fill into the Target Hardware Specification:

c:\VM_07\d_145_7z\d_7z020_fp12_4x8_v1_mx1\SDK\SDK_Export\hw\system.xml

This will specify one of the 10 precompiled HW designs present in the evaluation package. See Figure 8.

We have selected the "mix"design, demonstrating the use of four the UTIA EdkDSP accelerators, all with 8xSIMD data path, with floating point single data path division. The (8xSIMD) accelerators compiled in this design have four different capabilities defined by the pcores: bce_fp12_1x8_0_axiw_v1_40_a ; bce_fp12_1x8_0_axiw_v1_30_a ; bce_fp12_1x8_0_axiw_v1_20_a ; and bce_fp12_1x8_0_axiw_v1_10_a . This helps to demonstrate all four versions of the EdkDSP family in a single AMP demo.

Click on "Finish" button to finalize the selection of the precompiled HW design. See Figure 8.

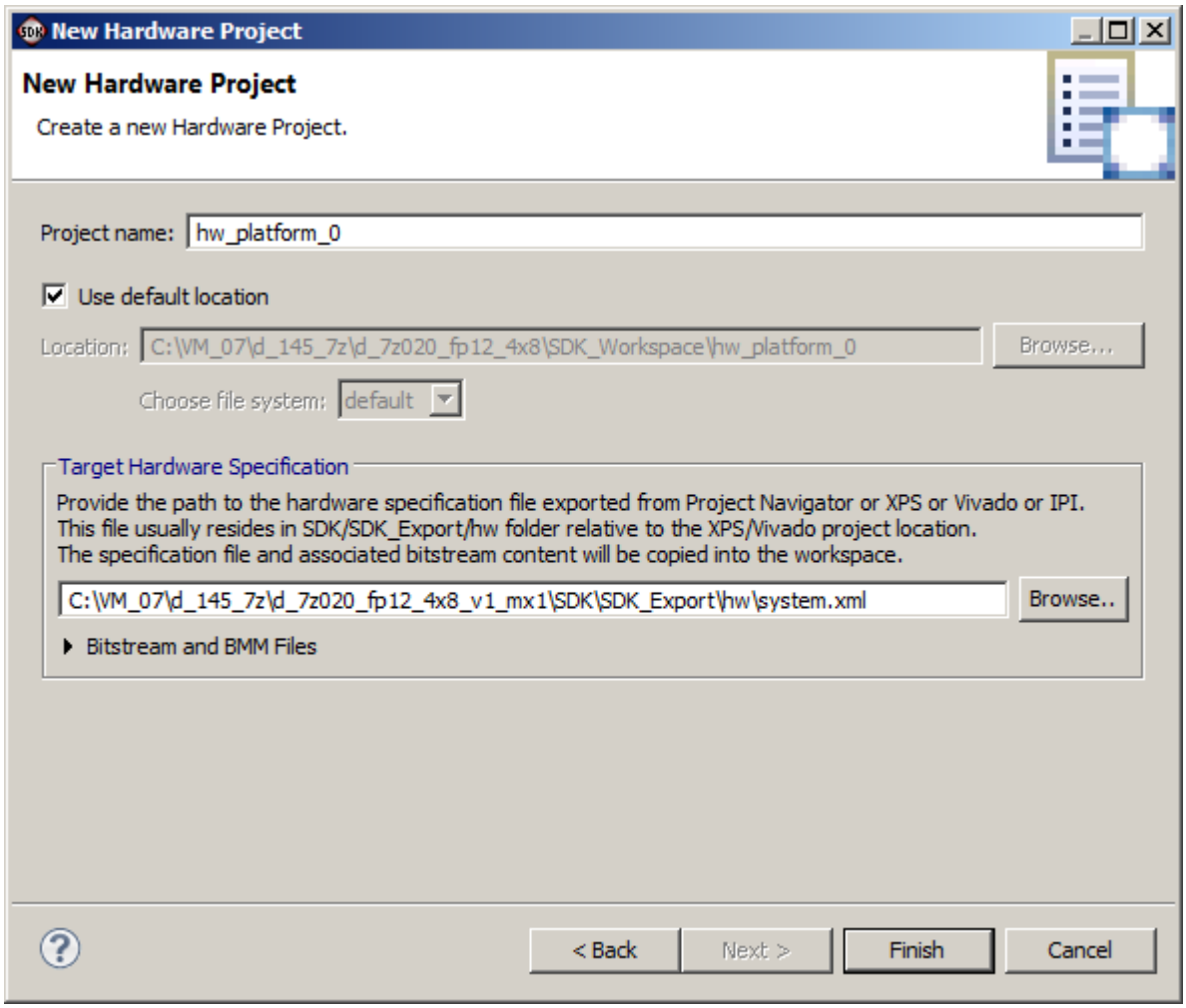

Figure 8: Use the name "hw_platform_0" and select one of the provided xml design descriptions

SDK is interpreting the system.xml and presents HW cores of in the design. See Figure 9.

The hardware platform "hw_platform_0" has been created.

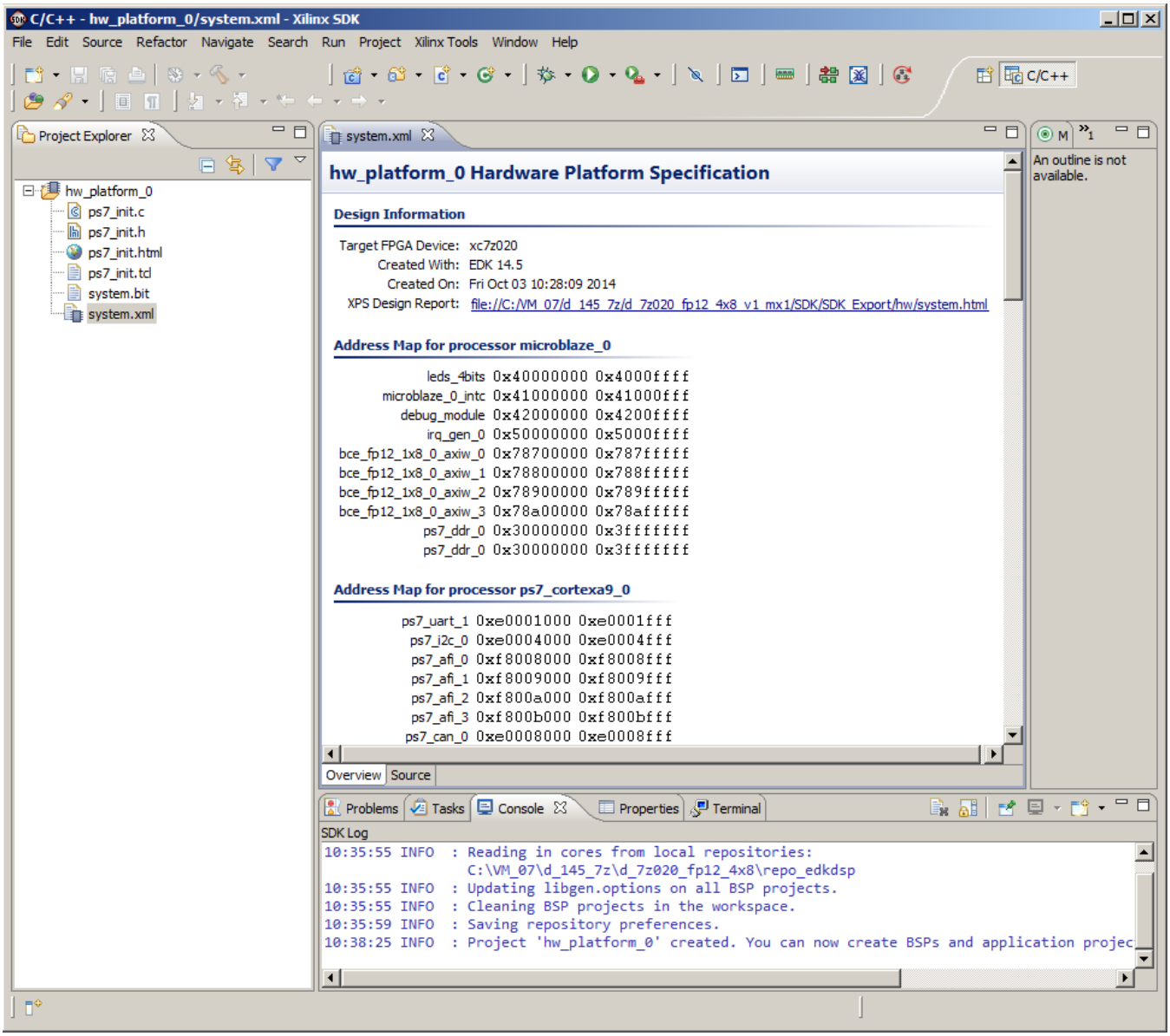

Figure 9: AMP Hardware Platform with the MicroBlaze and ARM Address Map

SW projects can be imported into SDK now. Select:

File -> Import -> General -> Existing Projects into Workspace
Click on Next button. See Figure 10.

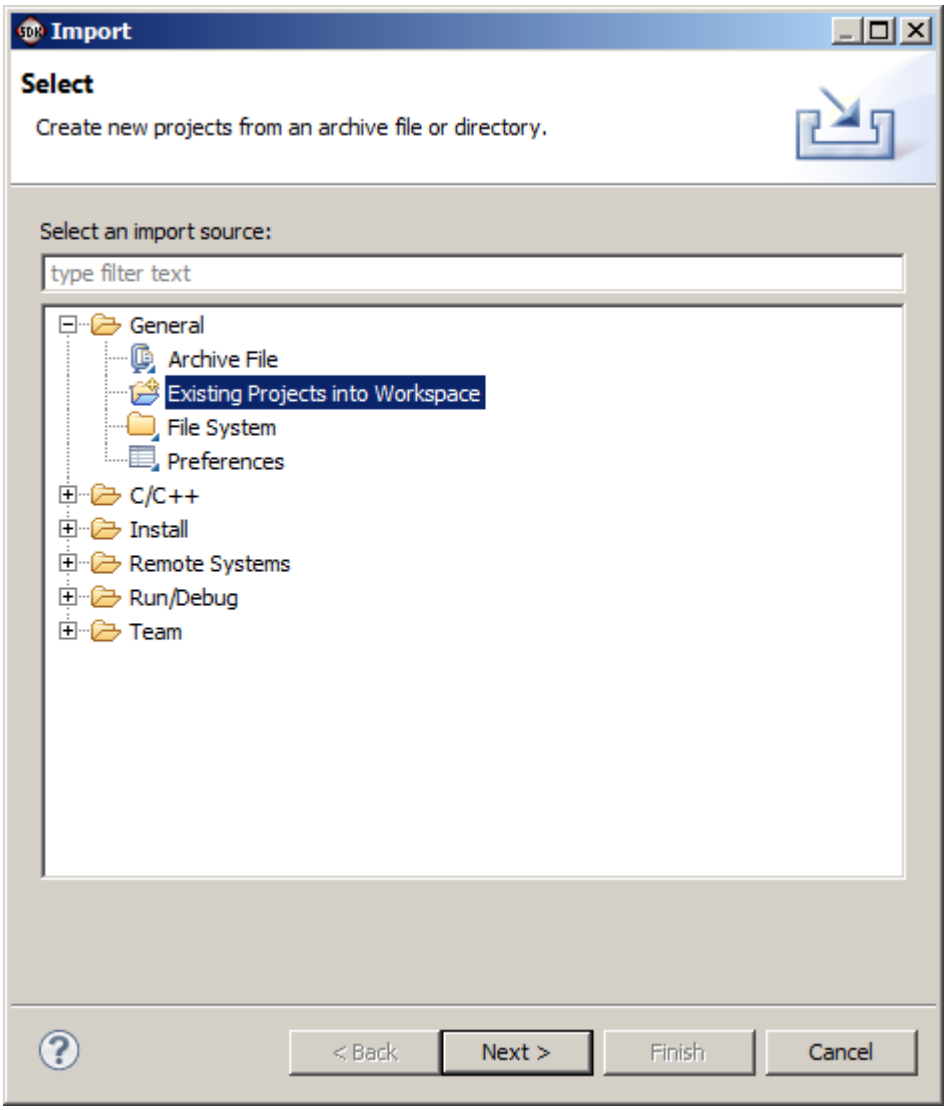

Figure 10: Import Existing Projects into Workspace

Type directory with projects to be imported. See Figure 11.

c:\VM_07\d_145_7z\d_7z020_fp12_4x8_IMPORT

Set the “Copy projects into workspace” check box.
Click on Finish button. See Figure 11.

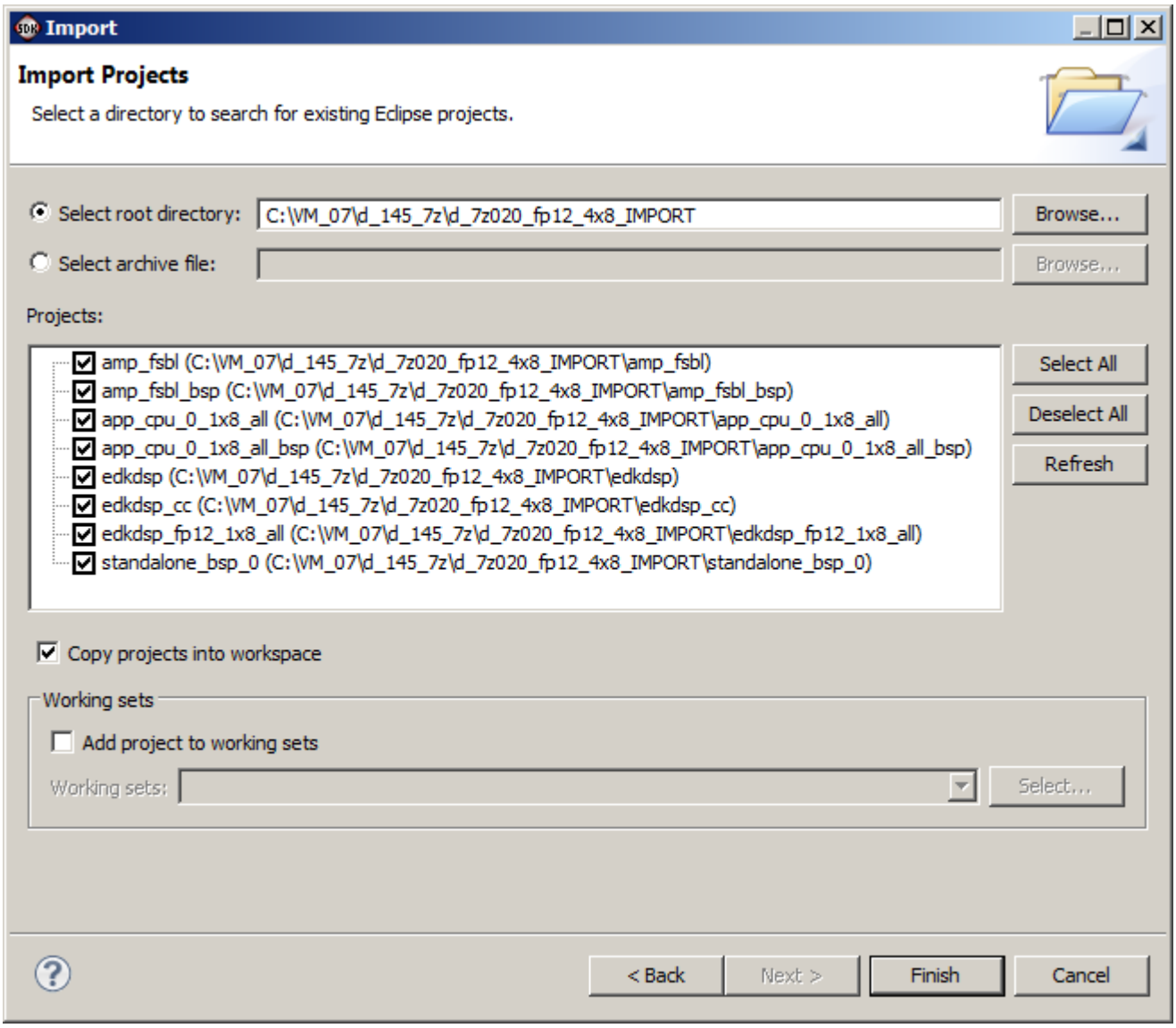

Figure 11: Select Copy Projects into Workspace and Finish the Import of all Projects.

All the UTIA EdkDSP SW projects are imported into SDK workspace from the directory **c:\VM_07\d_145_7z\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 12.

3.2 Asymmetric Multiprocessing Demo in Debug Mode

The “app_cpu_1x8_all” project in the “Project Explorer” window of the SDK 15.5 [7] will be used to program ARM core part of the AMP demo. The “edkdsp_fp12_1x8_all” project will be used to program the MicroBlaze core and the four EdkDSP accelerators of the AMP demo. The “app_fsbl” project is the first stage boot loader program for ARM. It will control the boot from the SD card. The “edkdsp” project contains the same SW content as the “edkdsp_fp12_1x8_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 EDKDSPCC compiler can be executed under the VMware player as a 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 12.

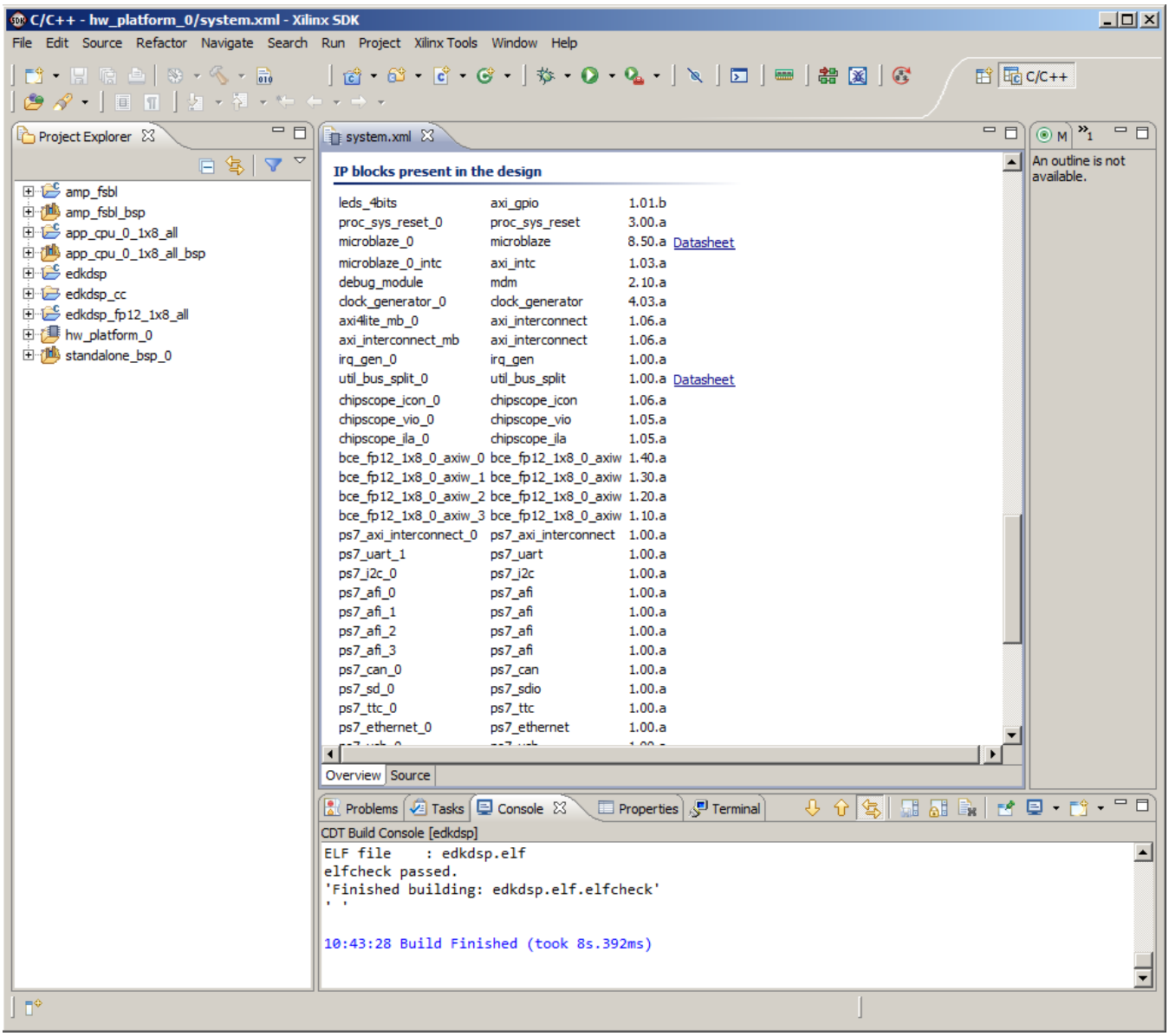

Figure 12: All projects are compiled. See IP Blocks present in the design.

Connect the jtag and serial line USB cables to your ZC702 board. Remove the SD card. Switch ON the board.

In SDK 14.5 [6], set up the jtag communication for the Digilent USB Cable. See Figure 13.

In SDK, select:
Xilinx Tools -> Configure JTAG Settings.

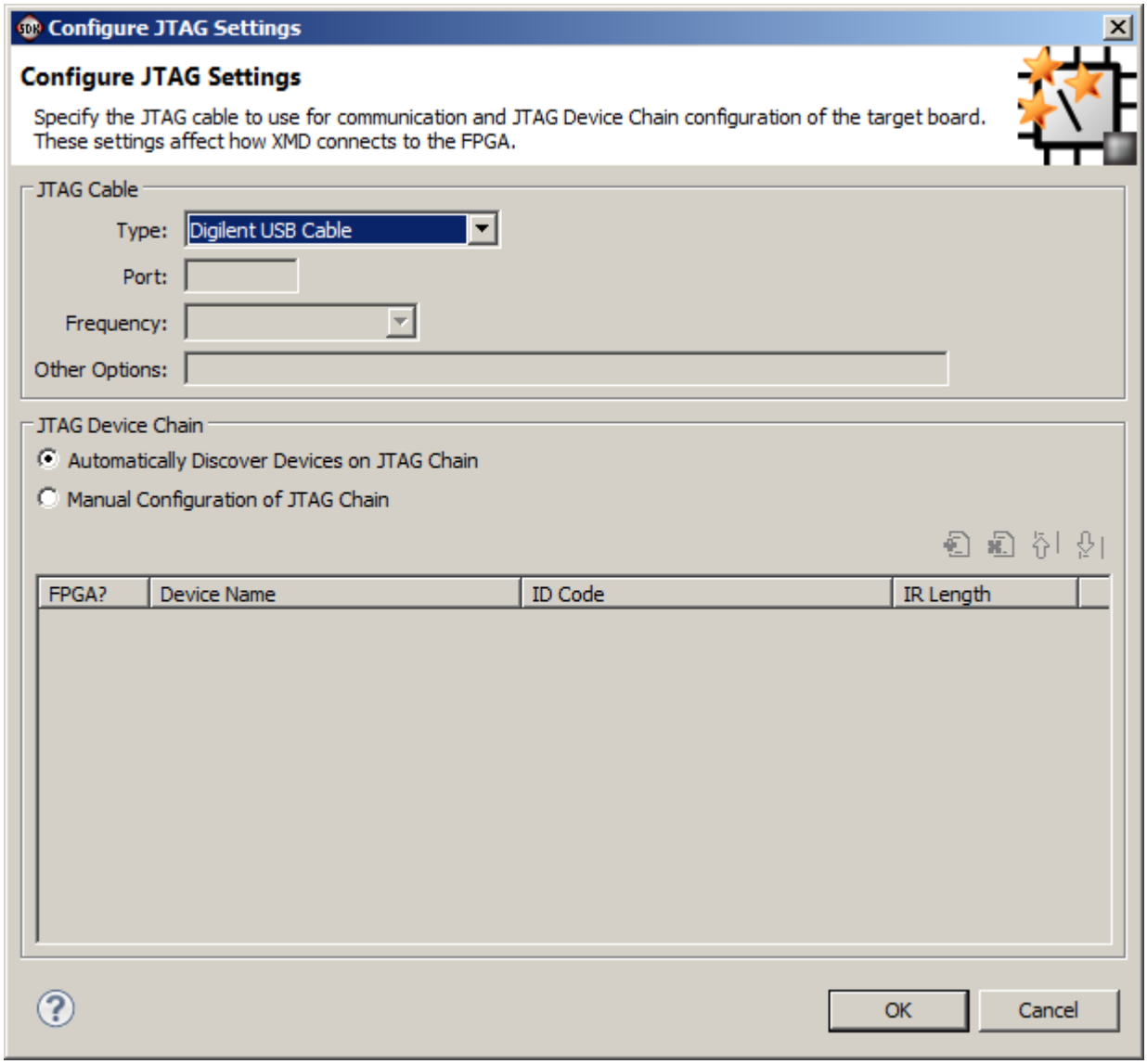

Figure 13: Specify JTAG cable to the Digilelnt USB Cable

In SDK, program the ZC702 board. See Figure 14.

In SDK, select:
Xilinx Tools -> Program FPGA

Click on the “Program” button.

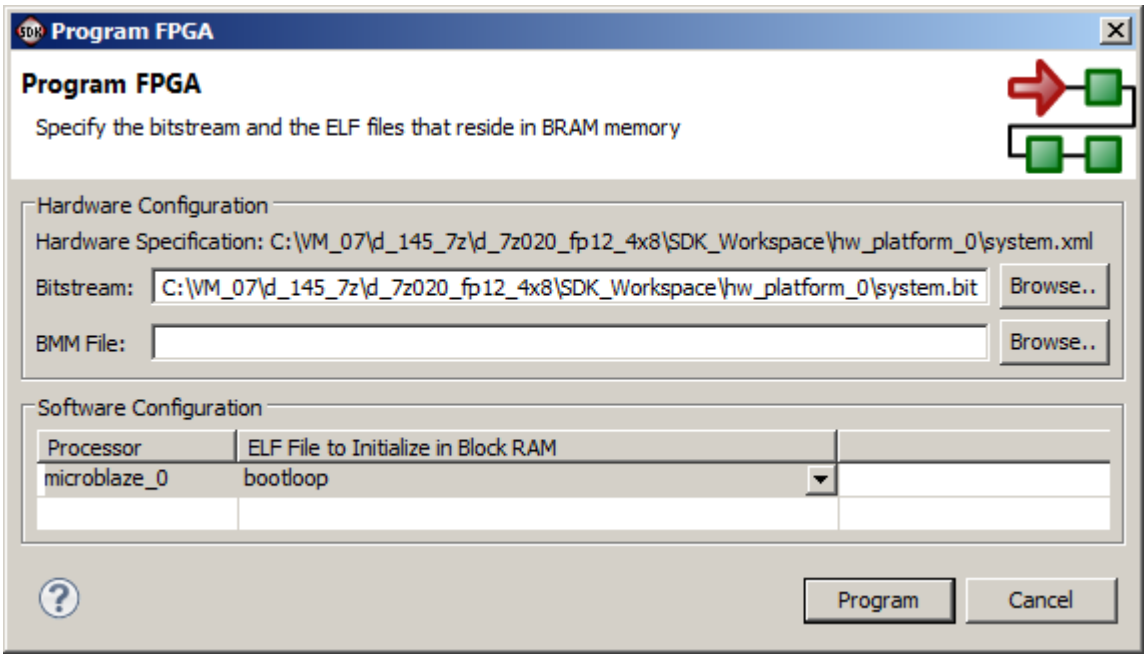

Figure 14: The bitstream system.bit is selected by the tool.

The ZC702 board is programmed with the .bit file now.

On PC, start the Putty terminal. Set 115200 baud and "Flow control" to None. See Figure 15 and Figure 16.

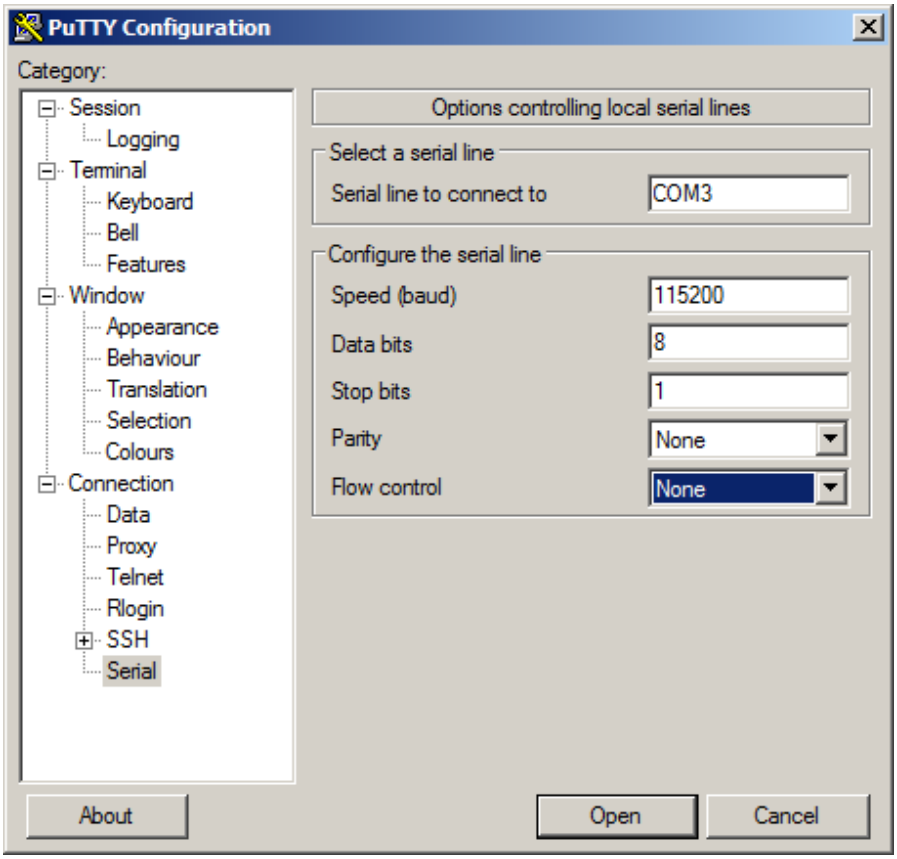

Figure 15: Setup your COM port. Select speed to 115200 baud, and Flow control "None"

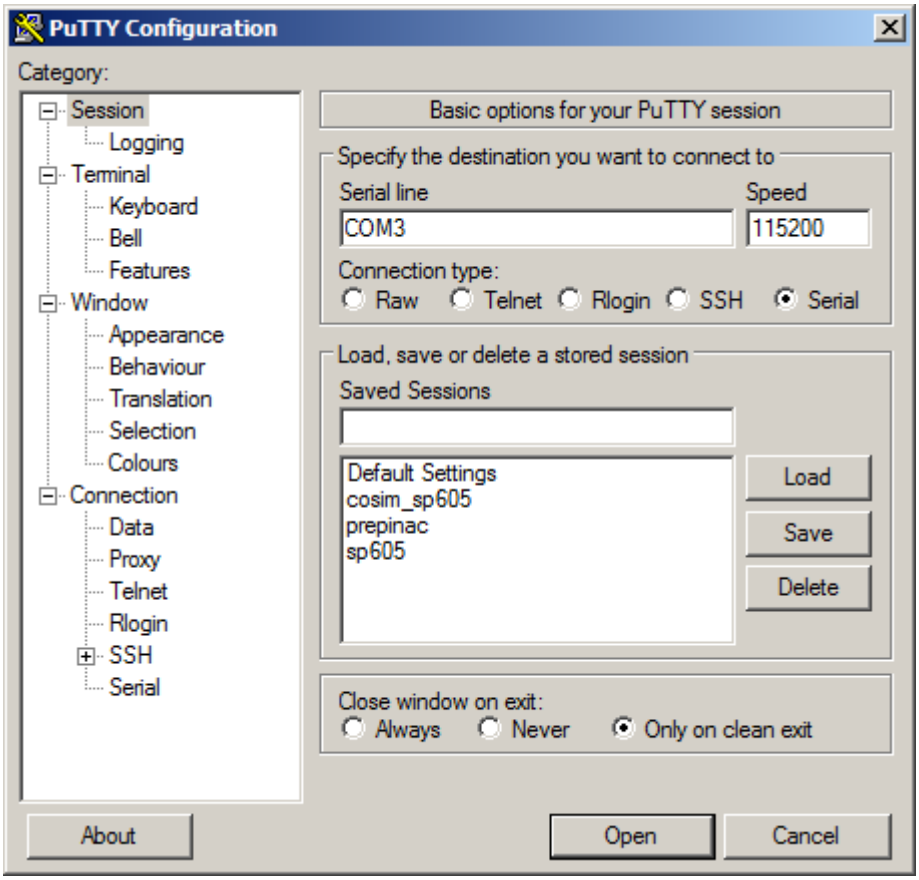

Figure 16: Select "Serial" in the category Session and click Open.

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

Select the "app_cpu_0_1x8_all" project by clicking on it in the SDK Project Explorer Window.

In SDK, select:
Run -> Debug Configuration ->Xilinc C/C++ ELF

Click on the "New launch configuration" in the Debug configuration screen. The "app_cpu_0_1x8_all" project is open and the ARM executable Debug\app_cpu_0_1x8_all.elf is ready for download to DDR3 on the ZC702 board via the jtag cable. See Figure 17.

Click on "Debug" button to download the executable. See Figure 17.

Click Yes in the perspective switch dialog window. See Figure 18.

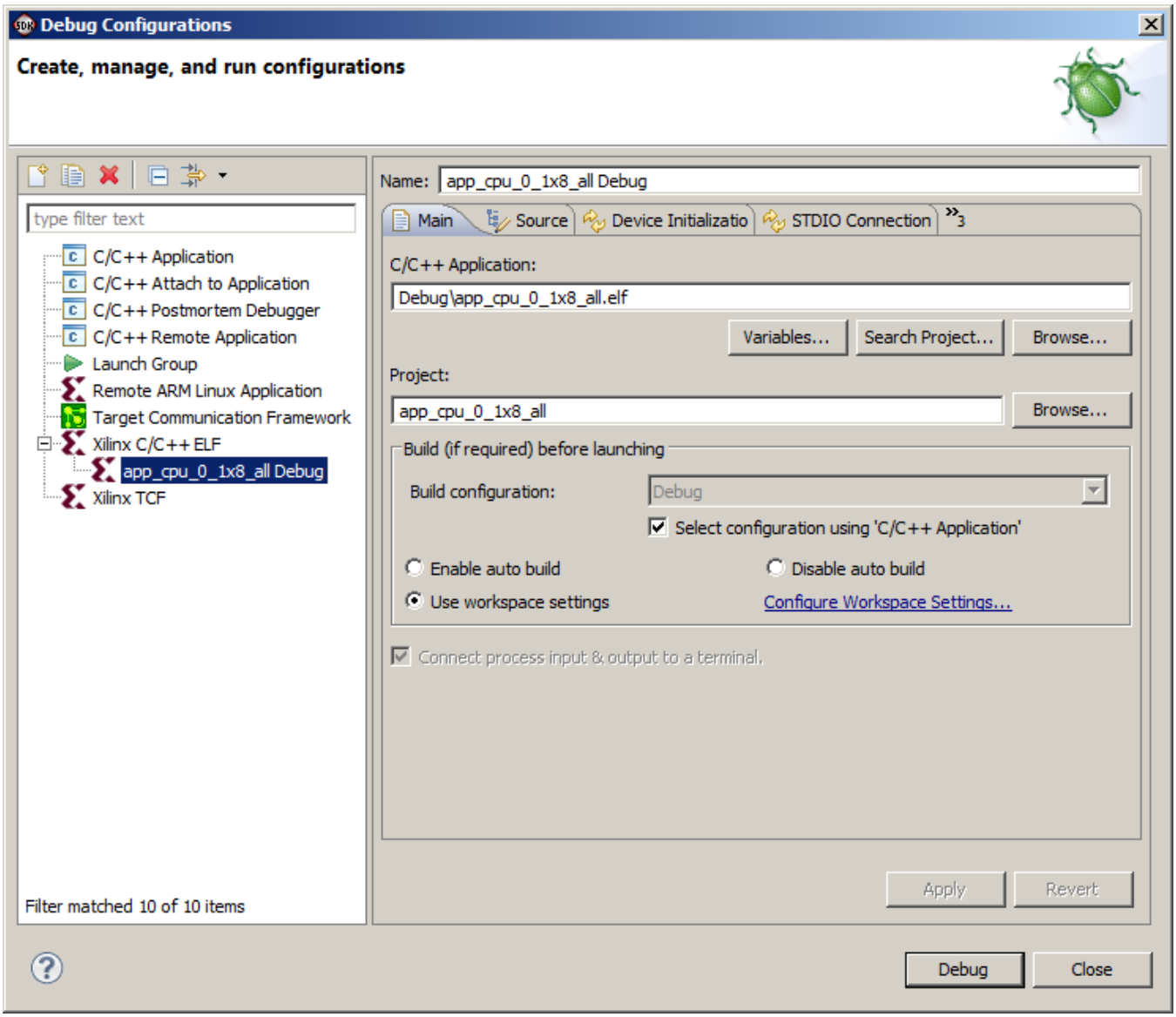

Figure 17: Select "app_cpu_0_1x8_all.elf" code for Debug on ARM

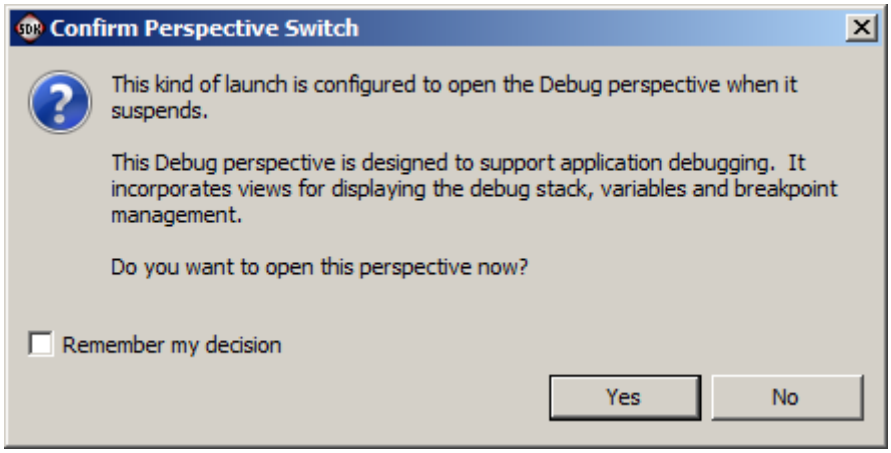

Figure 18: Click Yes to switch to the debug perspective.

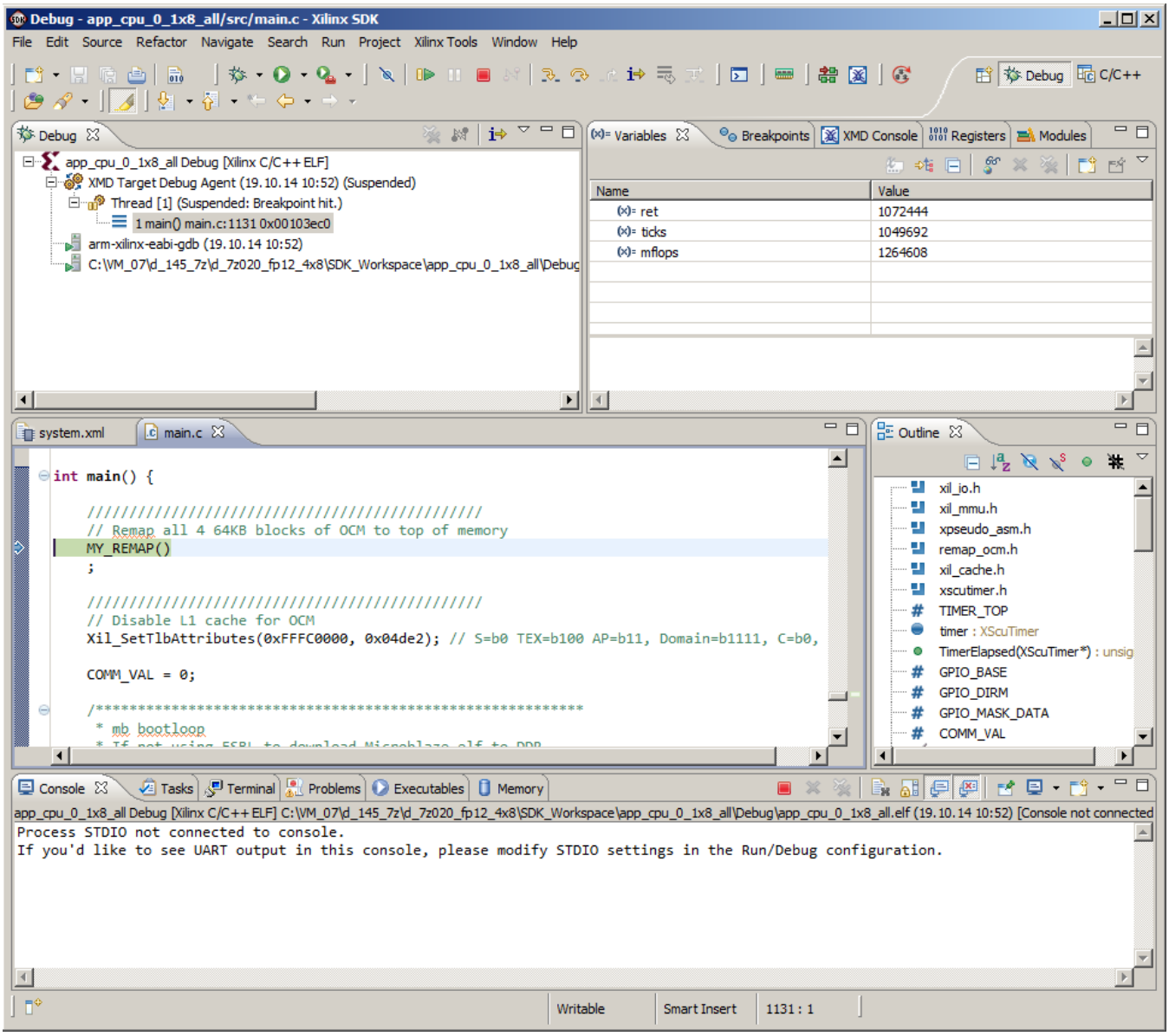

Figure 19: Run app_cpu_0_1x8_all.elf from the debugger as free running

The debug perspective is opened and Debug\app_cpu_0_1x8_all can be debugged or started on the ARM core. See Figure 19. Start the free running 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 20.

The ARM application app_cpu_0_1x8_all.elf has prepared the initial waiting loop code for the the MicroBlaze processor at the address 0x3000000 in the DDR3. The MicroBlaze has been released from reset by the ARM application. MicroBlaze is running the initial loop code at the address 0x3000000 now.

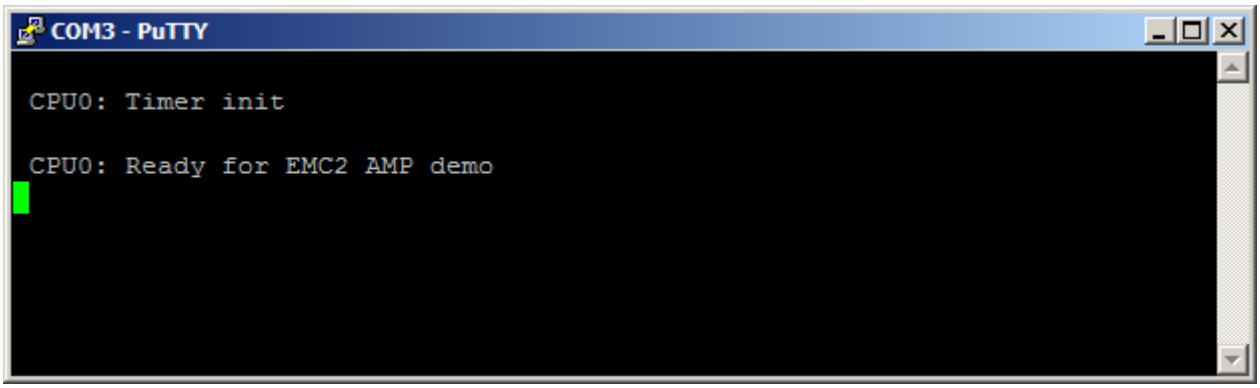

Figure 20: app_cpu_0_1x8_all.elf will write to the terminal the initial text and it will wait for the MicroBlaze part of the AMPapplication

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.

To open the second debugger instance select in SDK:
Xilinx Tools -> Launch Shell

In this new shell terminal, start the Xilinx MicroBlaze debugger by typing:

xmd

Connect to the MicroBlaze debugging core by typing:

connect mb mdm

The new GDB server will be started for the MicroBlaze at TCP port 1235.

Note: The ARM debugger is using the default TCP port 1234.

See the console on Figure 21.

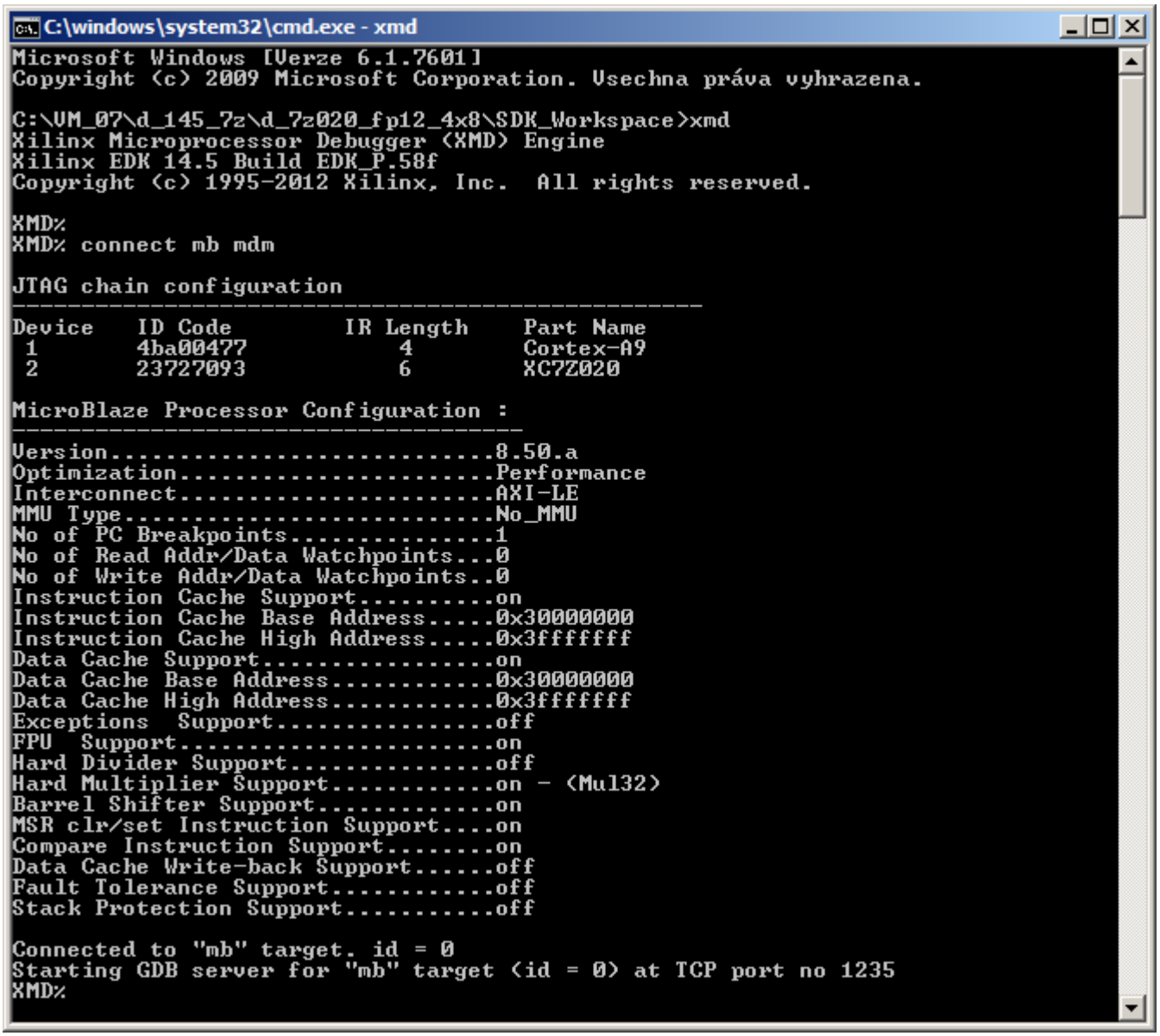

Figure 21: Start second GDB server at TCP port 1235

The MicroBlaze debug can be started now.

Select the “edkdsp_fp12_1x8_all” project by clicking on it in the SDK Project Explorer Window.

In SDK, select:
Run -> Debug Configuration ->Xilinc C/C++ ELF

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 ZC702 board via the jtag cable, to the address 0x30000000. See Figure 22.

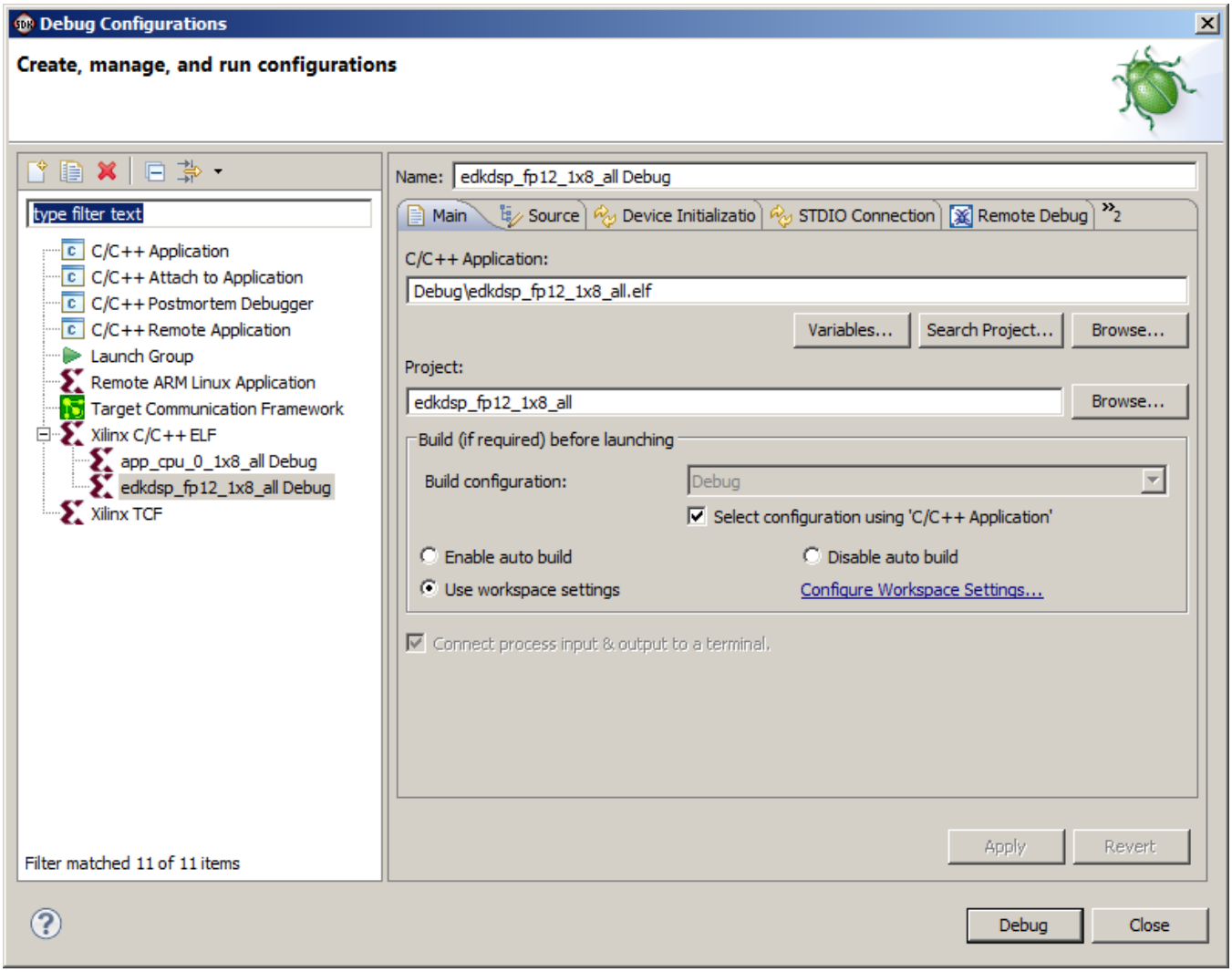

Figure 22: Select "edkdsp_fp12_1x8_all.elf code for remote debug on MicroBlaze

This debug session will need 2 adjustments to co-exist together with the running ARM debug session. See Figure 23 and Figure 24.

Select the Device Initialisation dialog screen and delete the default initial .tcl script. It is not needed in this case. The Programmable System has been already initiated by the first ARM debug session. See Figure 23.

Select the Remote debug screen and select: Connect to GDB on another machine (we will be connecting to the second TCP server on the same machine indicated as localhost). See Figure 24.

Change the port from the default 1234 to the port 1235 used by the second server. See Figure 24.

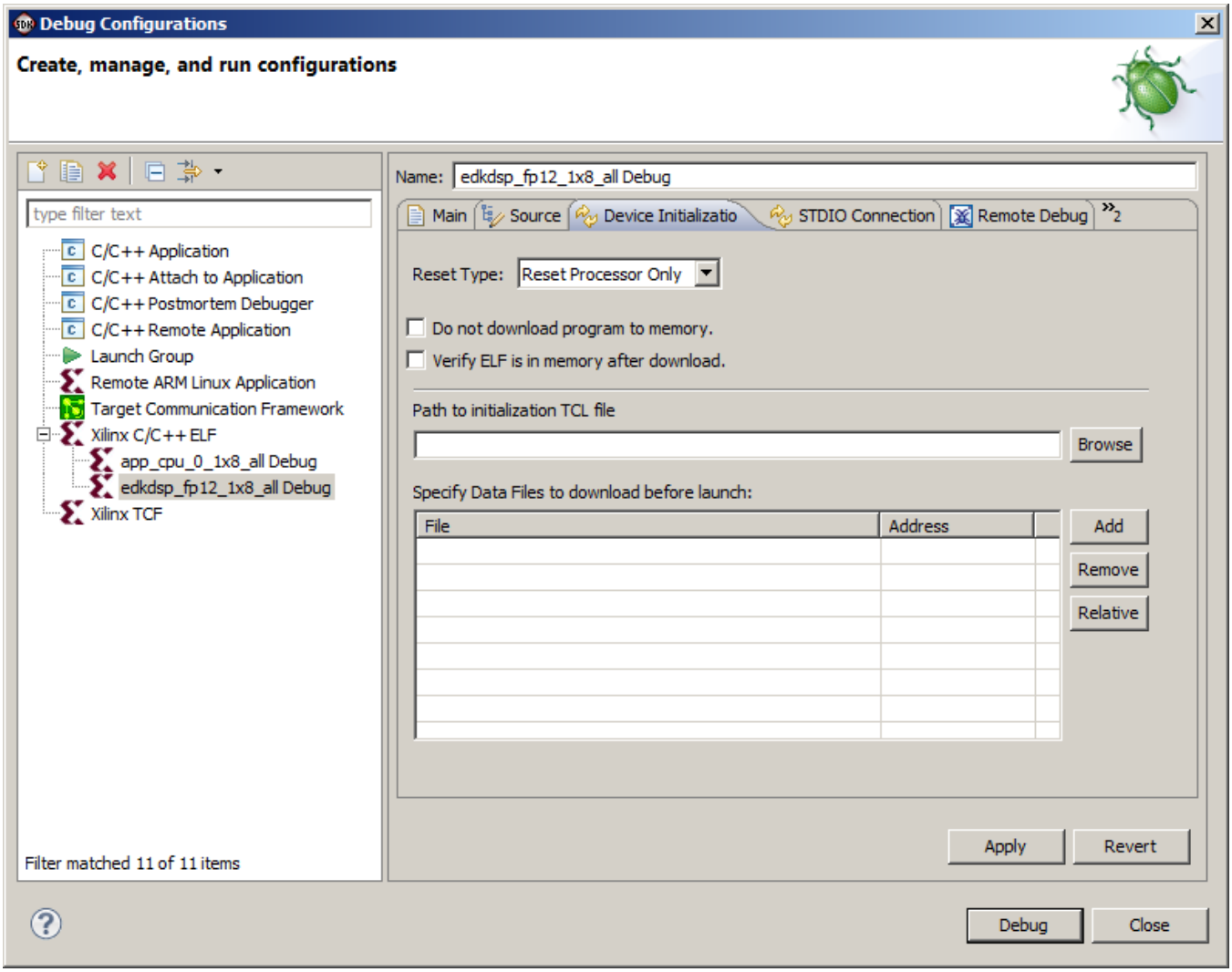

Figure 23: Clear the pre-defined .tcl file in the Device Initialization sub-screen

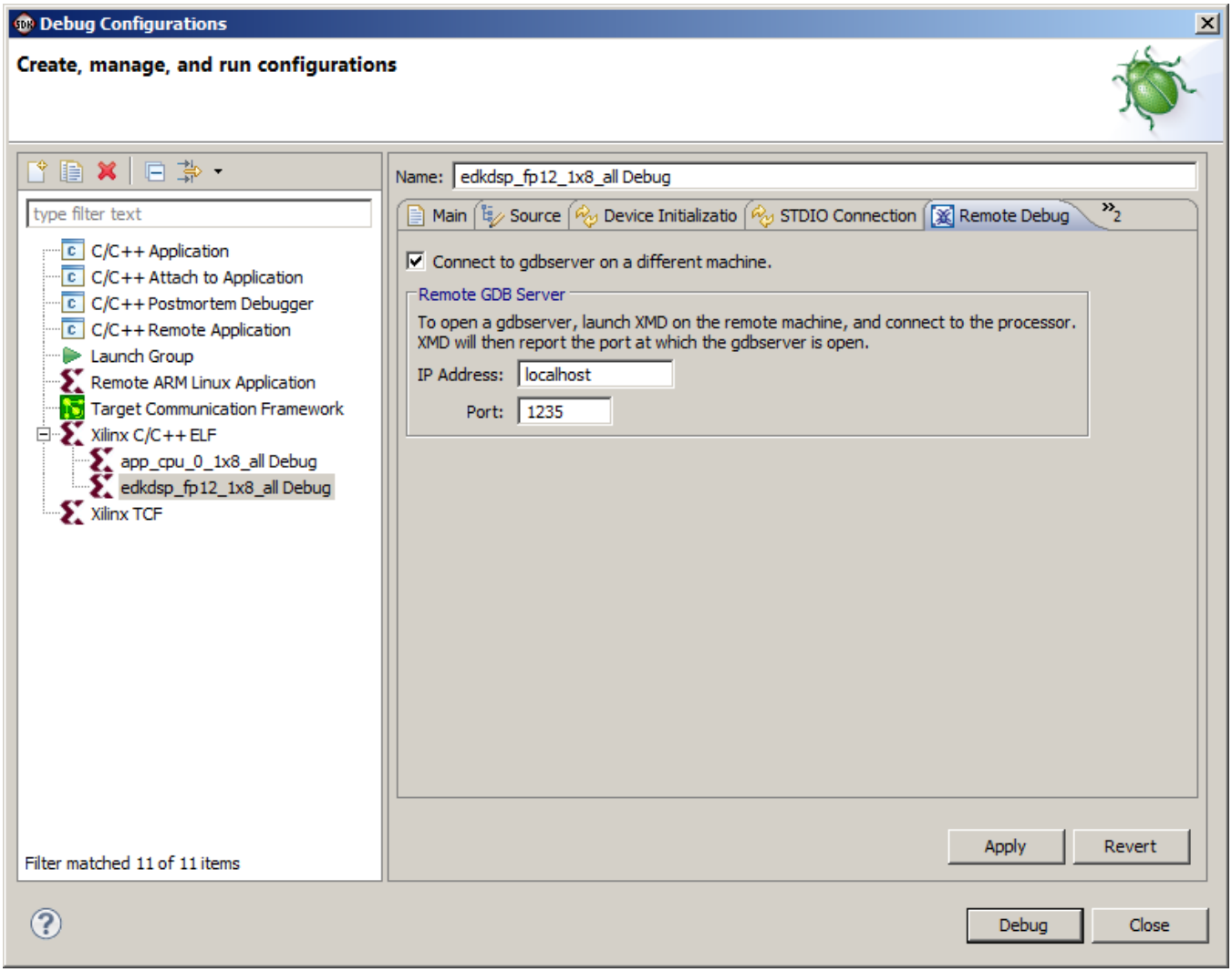

Figure 24: Select Connect to gdbserver on different machine and modify port to 1235

Click on Apply to apply both changes. See Figure 24.

Click on Debug to start the debuger.

The program edkdsp_fp12_1x8_all.elf will be downloaded to DDR3 from address 0x3000000 and this second remote MicroBlaze debug section will be opened in the SDK, together with the already running ARM debug section. See Figure 25.

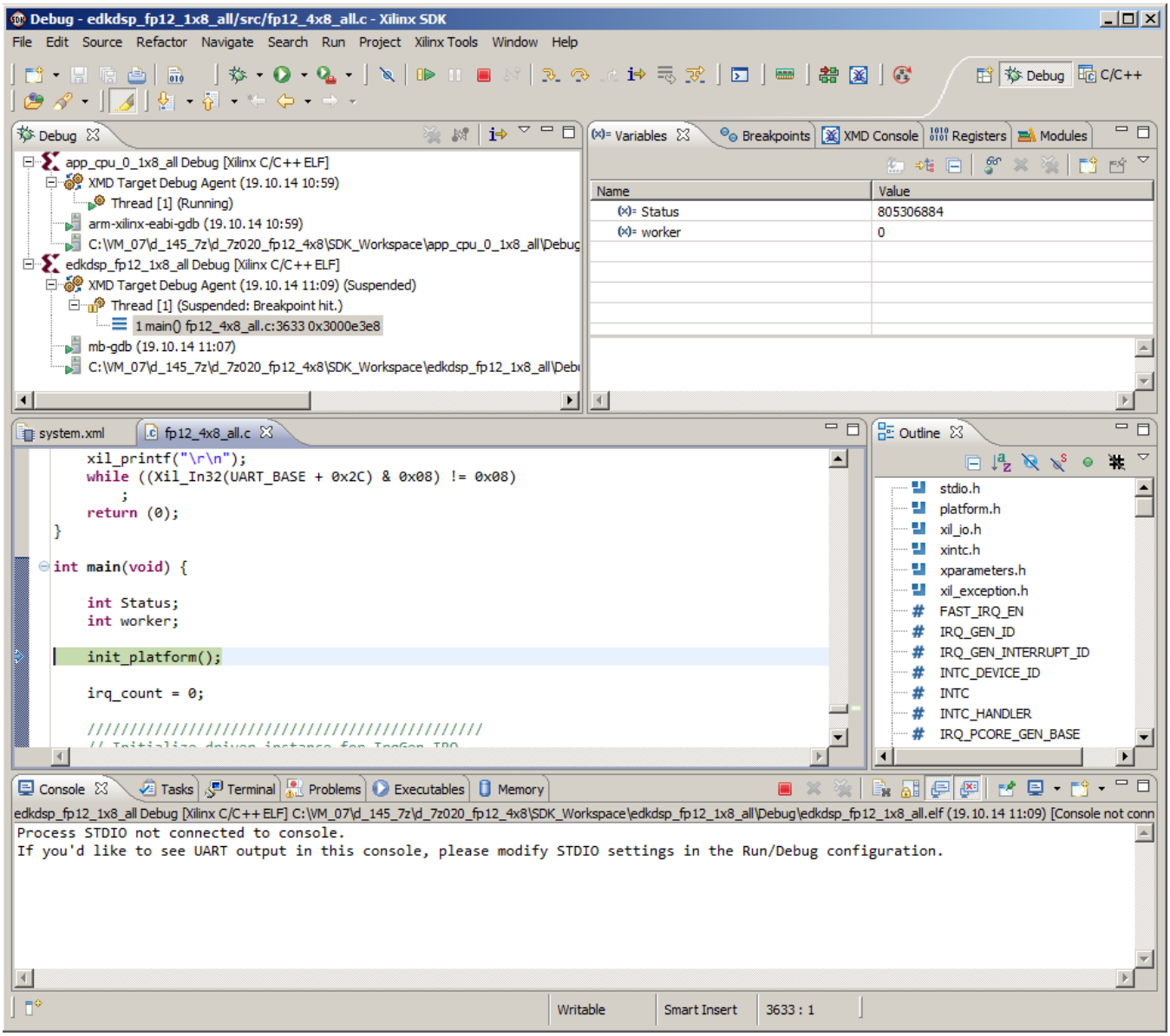

Figure 25: Debug view with ARM and MicroBlaze. ARM is running. Start MicroBlaze.

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 25. 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 26.

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. 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 26.

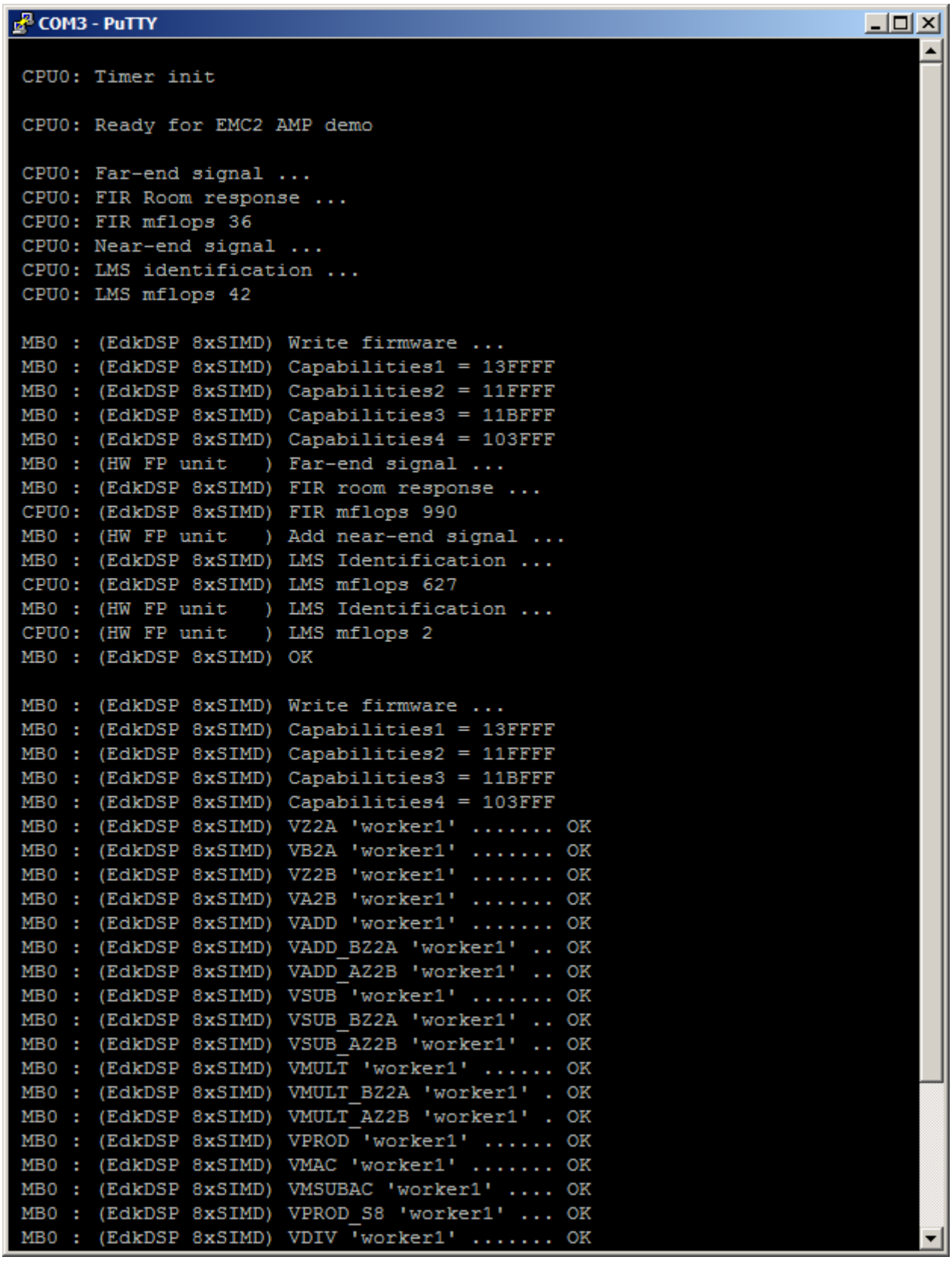

Figure 26: Terminal output from the debugged AMP demo application.

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 26. The demo application loops infinitely, and the test is performed in a sequence on all 4 (8xSIMD) EdkDSP accelerators. The accelerators are named worker1 … worker4. Workers have different capabilities (in case of the mix HW design). This is reflected in the output from tests. Worker 1 supports all vector operations, worker 2 less, worker 3 even less and worker 4 supports only minimal set of vector operations. This corresponds to the four (8xSIMD) EdkDSP cores used in the design:

Worker 1 = bce_fp12_1x8_0_axiw_v1_40_a ;
Worker 2 = bce_fp12_1x8_0_axiw_v1_30_a ;
Worker 3 = bce_fp12_1x8_0_axiw_v1_20_a ;
Worker 4 = bce_fp12_1x8_0_axiw_v1_10_a .

The capabilities of each worker are reported by each worker to the MicroBlaze. See Figure 26.

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

The debug session has to be stopped from the debugger and the second GDB server can be stopped from the shell window by typing:

stop	to stop the MicroBlaze processor
rst	to perform the system reset
exit	to exit from the second GDB server program
exit	to close the console

See Figure 27.

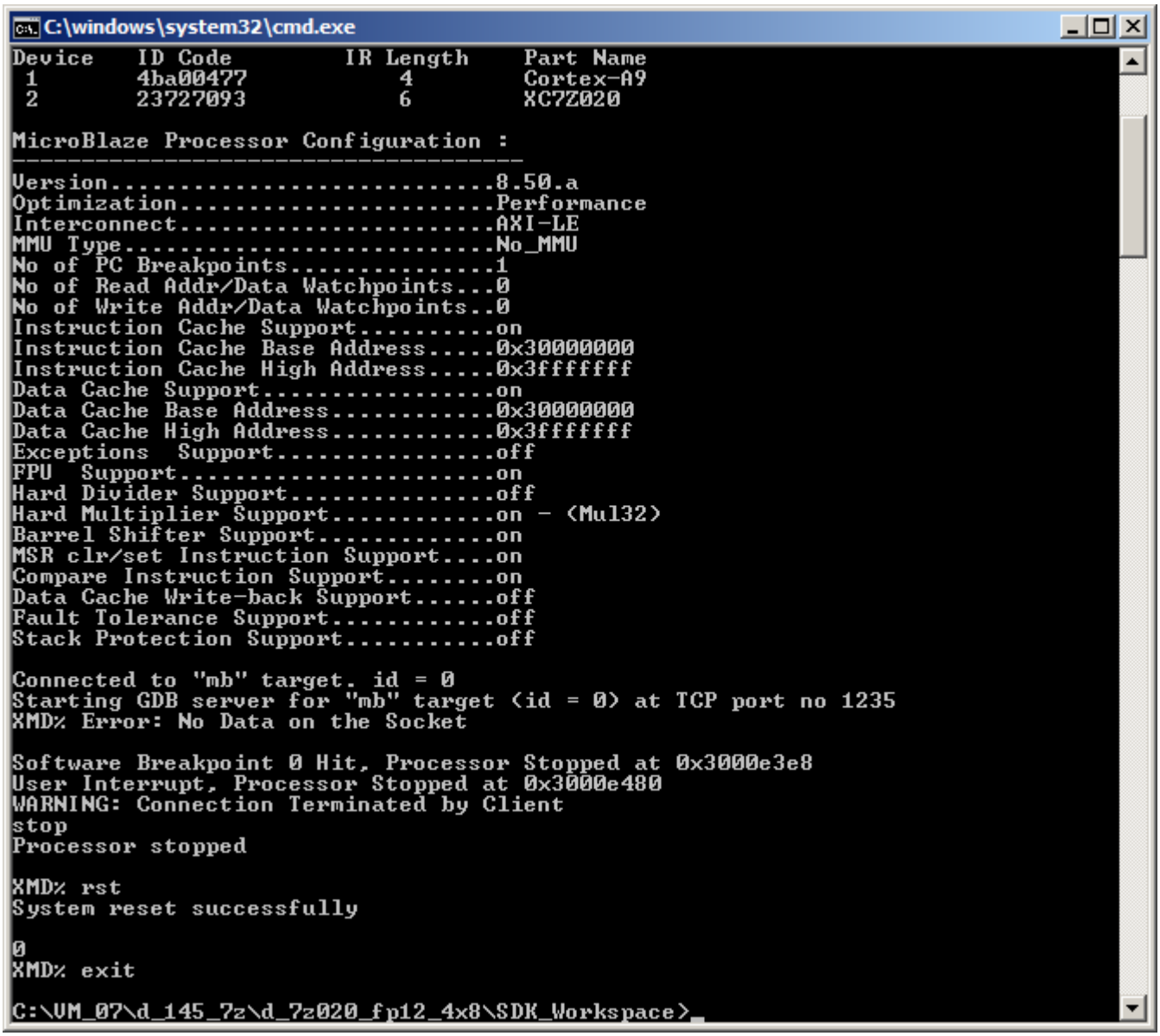

Figure 27: Stop, Reset and Exit from the second GDB server

3.3 Asymmetric Multiprocessing Demo with Optimised Code

All demos can be recompiled into the faster release mode with the -O3 optimization to get increased performance of ARM and MicroBlaze processors. In SDK 14.5, recompile all projects to the release subdirectories and delete the debug subdirectories.

Switch-OFF and ON the ZC702 board.
Repeat all steps described in section 3.2, to run the optimised version of the AMP demo.
Start with application app_cpu_0_1x8_all.elf for the ARM processor. See Figure 28.

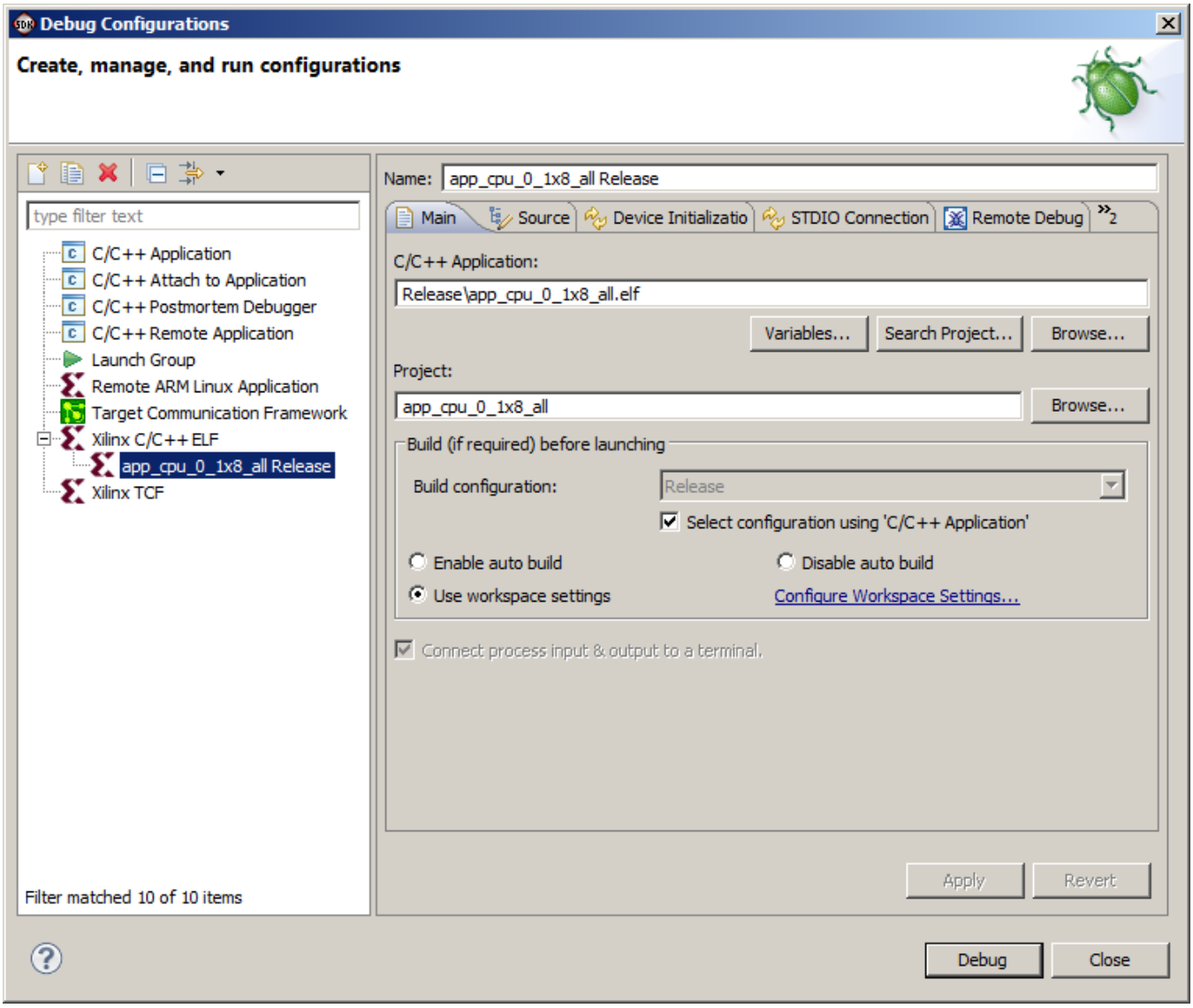

Figure 28: Open the app_cpu_0_1x8_all.elf application (compiled for release) in the ARM debugger.

Start the free run on the ARM CPU0. See Figure 29. There is no source code debug information, now.

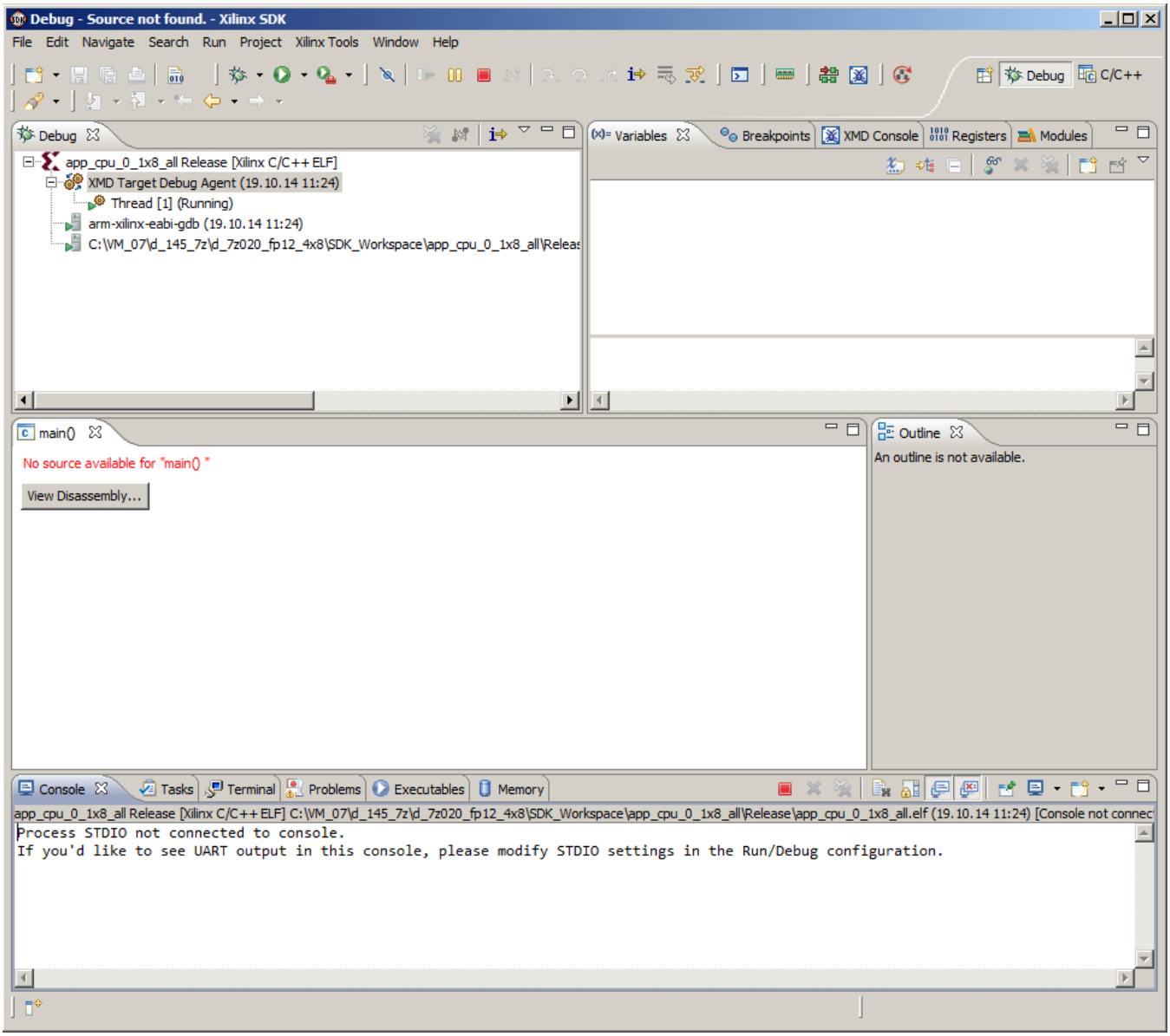

Figure 29: Run the app_cpu_0_1x8_all.elf application on ARM

In SDK, open new Shell. Type: xmd
In XMD type: connect mb mdm
Select "edkdsp_fp12_1x8_all" project by clicking on it in the SDK Project Explorer Window.
In SDK, select:
Run -> Debug Configuration ->Xilinc C/C++ ELF
Click on the "New launch configuration" in the Run configuration screen. The "edkdsp_fp12_1x8_all" project is open and the MicroBlaze executable Release\edkdsp_fp12_1x8_all.elf will be ready for download to the DDR3 address 0x30000000. This debug session will again need the adjustments. Select the Device Initialisation dialog screen and delete the initial .tcl script. Select the Remote debug screen and select: Connect to GDB on another machine and port 1235. Start debug and free run the MicroBlaze code.

The terminal output will demonstrate the increased MFLOP/s performance of the ARM processor and the increased performance of the MicroBlaze processor. See Figure 30.

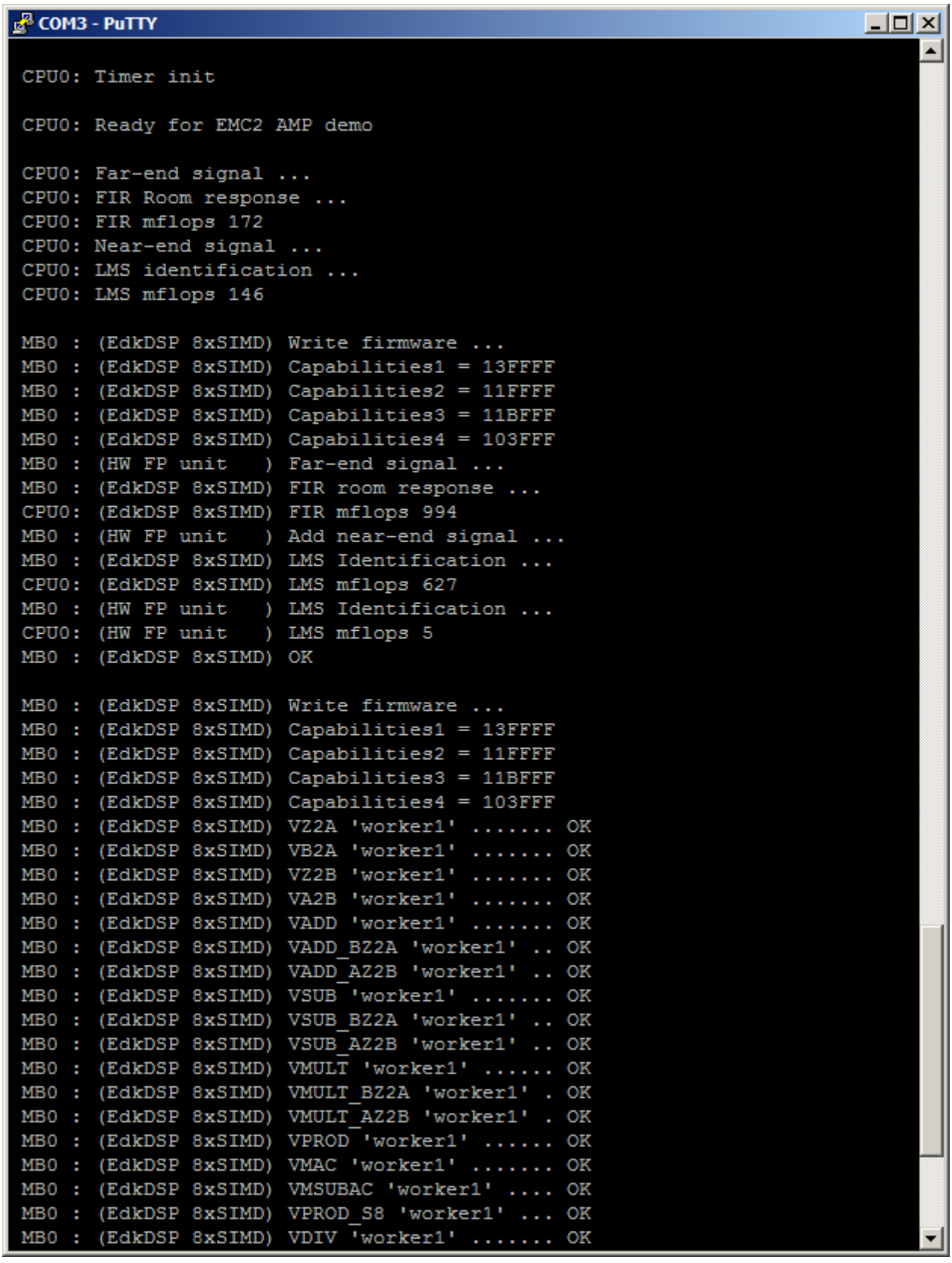

Figure 30: Terminal output from the recompiled optimized AMP demo application.

3.4 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.

Open the main.c source code in app_cpu_0_all project in the SDK Project Explorer. Comment the two lines as indicated in Figure 31. 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 0x3000000. This is needed only in the case of the jtag boot of MicroBlaze from the second remote debugger.

In case of the boot of the AMP demo from the SD card the first stage boot loader program amp_fsbl.elf will be instructed to write the final MicroBlaze program from the address 0x3000000.

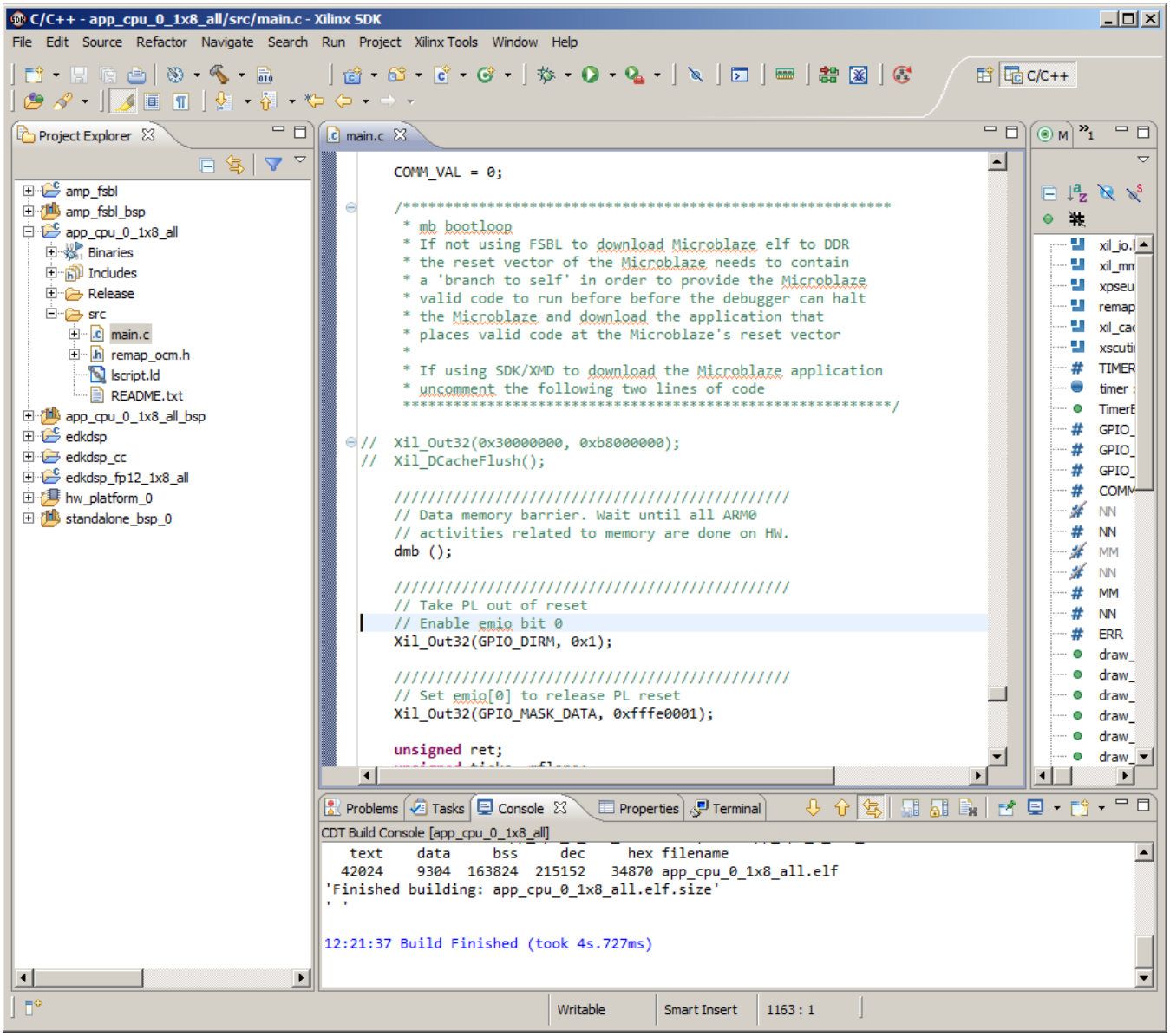

Figure 31: Modification of ARM code for boot of the AMP application from SD card.

In SDK, select:
Xilinx Tools -> Create Zynq Boot Image

Fill all the paths as indicated in Figure 32. The sequence is important. First add the amp_fsbl.elf. Second add the system.bit file. Third is the application code for ARM processor. Fourth is the application code for the MicroBlaze processor.

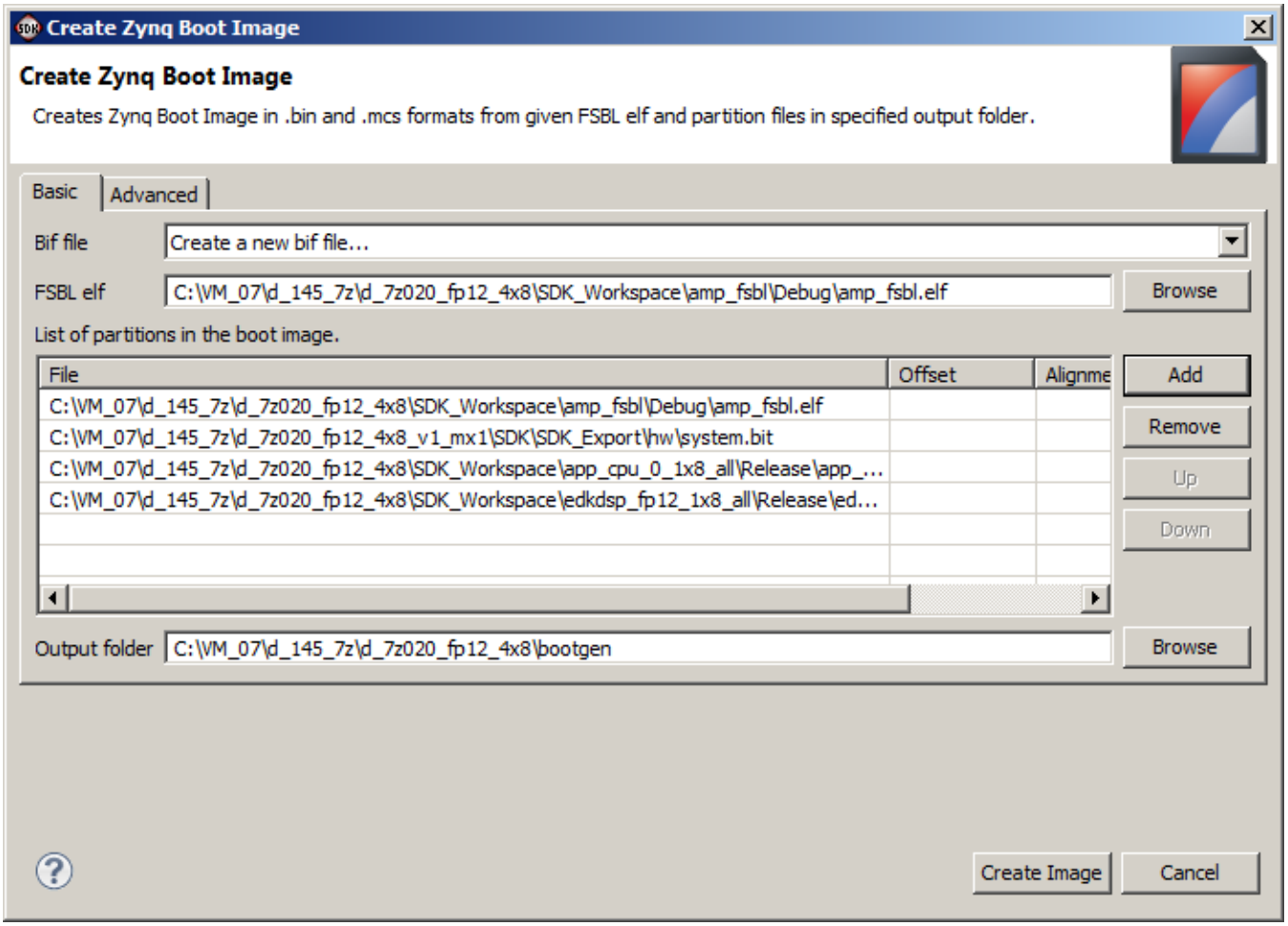

Figure 32: Select files for generation of BOOT.BIN image file for the SD card

Click Create Image and the tool will generate edkdsp_fp12_1x8_all.bin in the bootgen directory.
Copy edkdsp_fp12_1x8_all.bin to the SD card to the top level directory and rename it to BOOT.BIN on the SD card.

Insert the SD card and power up the ZC702 board. The board will boot from the SD card with output to the terminal. The output will be identical to Figure 30.

3.5 Evaluation of 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 ZC702 board.

The evaluation package includes also precompiled files with the firmware ready for download from PC to the ZC702 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 from the Internet the VMware Player software (64bit or 32bit) on your PC.

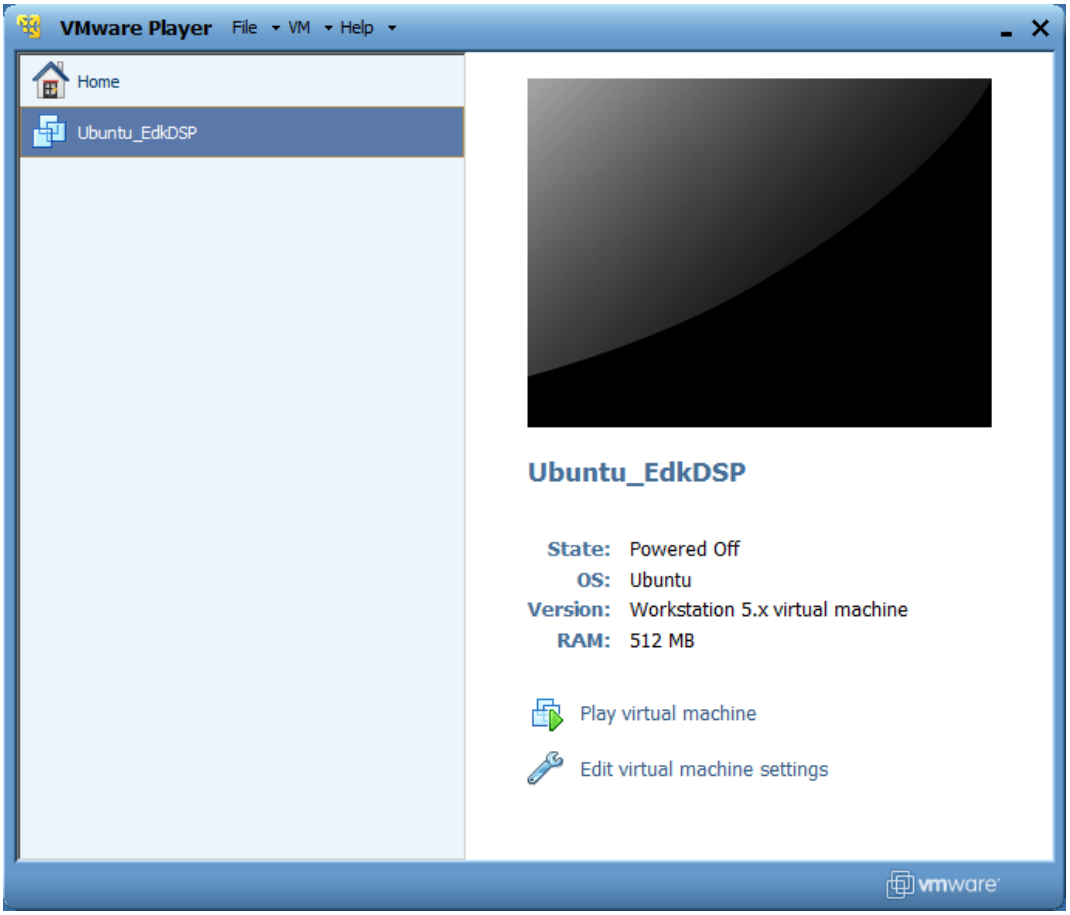

Figure 33: Select the Ubuntu_EdkDSP image in the VMware Player and click "Play".

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 with the __vmware_user__ 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

In Ubuntu terminal, change the directory to:
/mnt/cdrive/d_145_7z/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

devel@ubuntu:/mnt/cdrive/d_145_7z/d_7z020_fp12_4x8/SDK_Workspace/edkdsp_cc/a$

See these steps in Figure 34.

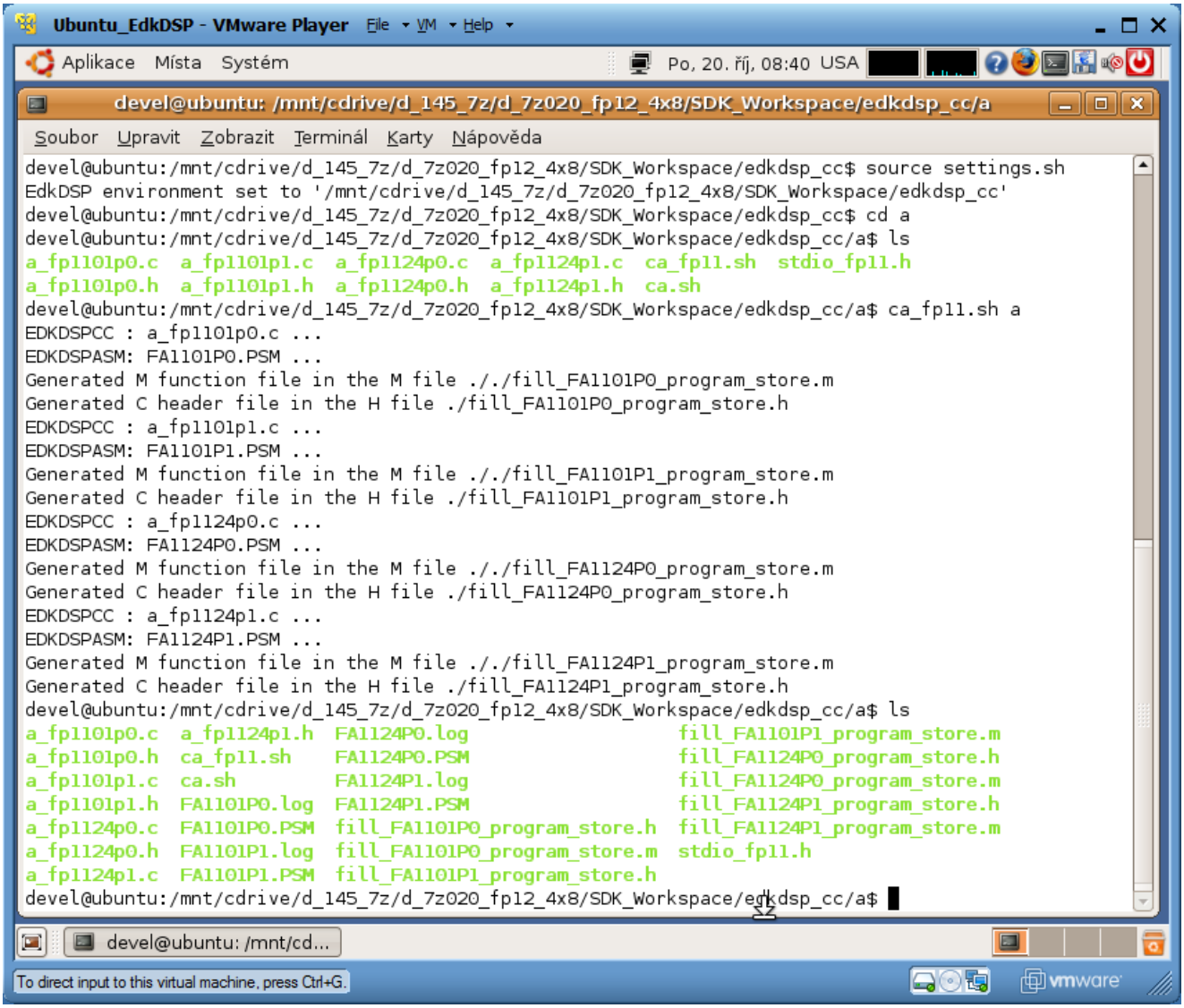

Figure 34: Compilation of EdkDSP firmware in Ubuntu.

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

Copy and paste the compiled headers into the src directory of the MicroBlaze project "edkdsp_fp12_1x8_all" of the SDK 14.5 AMP demo.

See the initial list of C firmware files for the EdkDSP in the SDK before compilation in Figure 35.
See the generated header files with compiled firmware after the compilation in Figure 36.

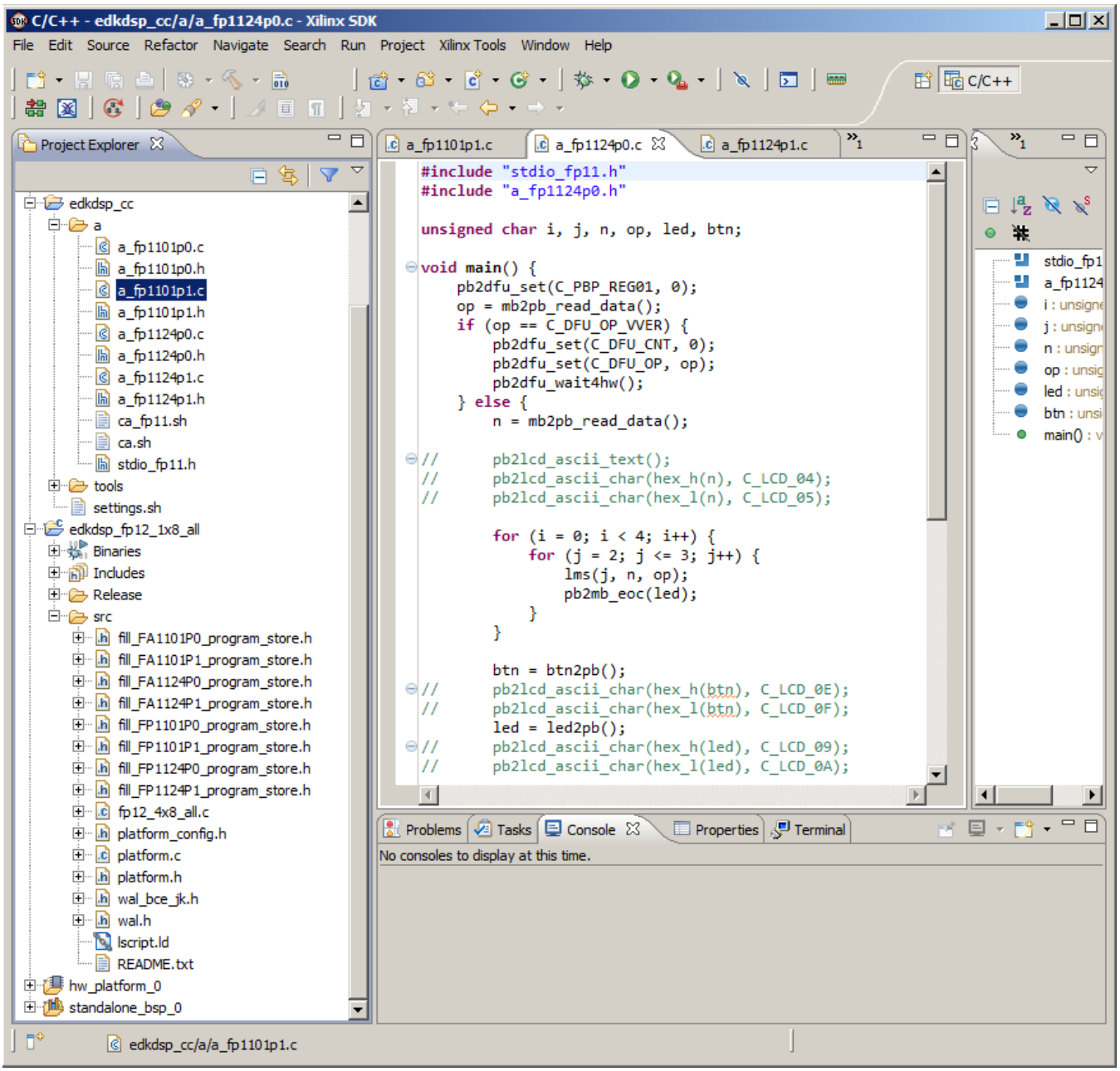

Figure 35: Initial firmware files and C listening of the LMS filter firmware for the EdkDSP.

The EdkDSP firmware before compilation is presented in Figure 35. See also the C code listing of the firmware for computation of the LMS in the (8xSIMD) EdkDSP platform.

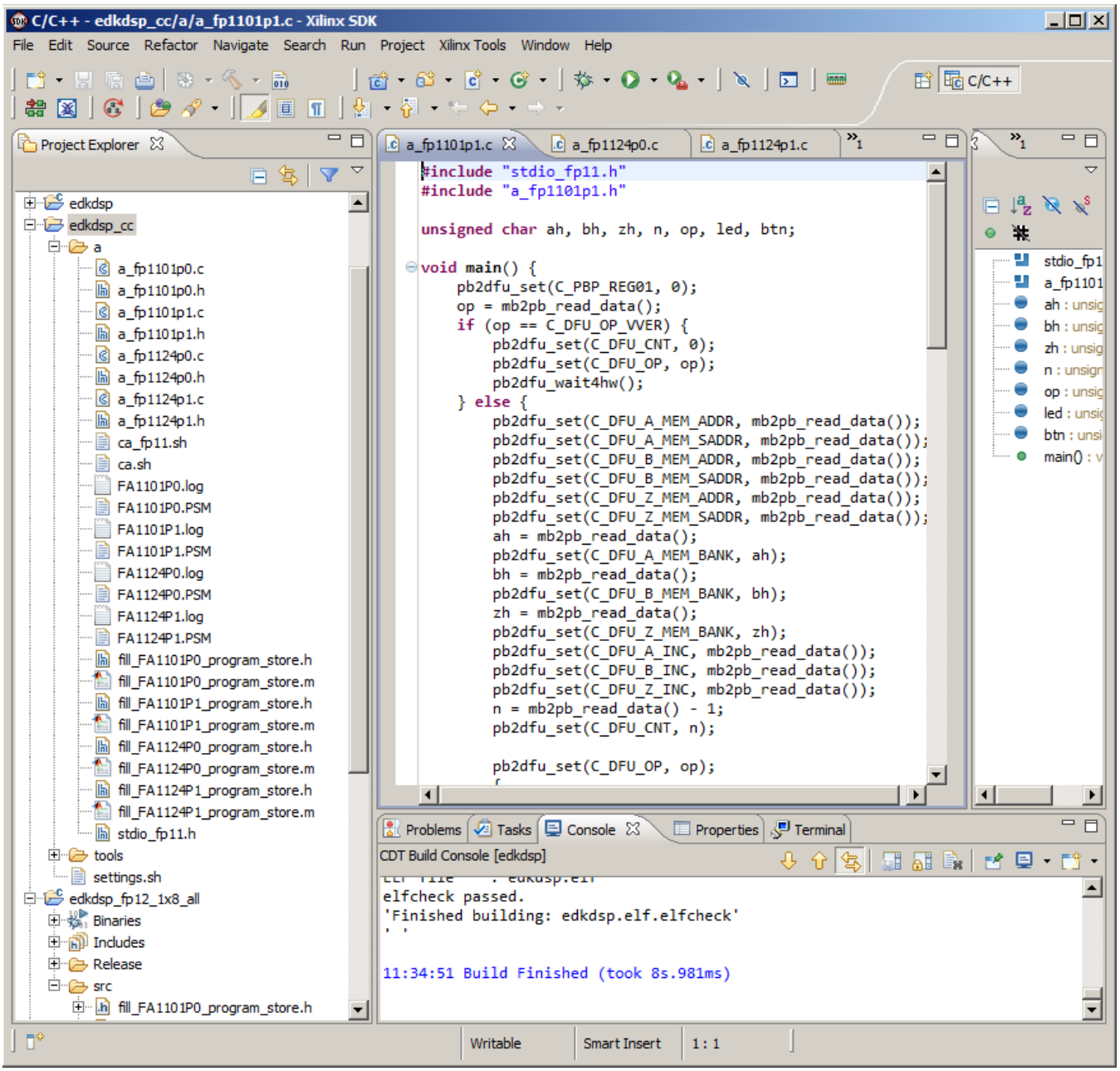

Figure 36: Initial setup of shared directory with Windows 7 and Sourcing of PATH

The EdkDSP firmware after the compilation is presented in Figure 36.
See also the C code listing of the firmware for the basic test of vector operations in the (8xSIMD) EdkDSP platform. To use the compiled headers in the SDK project, use the copy and paste to copy 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_1x8_all/src/ and recompile the MicroBlaze project "edkdsp_fp12_1x8_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.

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] ZC702 Evaluation Board for the Zynq-7000 XC7Z020 All Programmable SoC User Guide UG850 (v1.3) June 4, 2014;
http://www.xilinx.com/support/documentation/boards_and_kits/zc702_zvik/ug850-zc702-eval-bd.pdf

[3] Zynq-7000 All Programmable SoC ZC702 Evaluation Kit Quick Start Guide
http://www.xilinx.com/support/documentation/boards_and_kits/zc702_zvik/xtp310-zc702-quickstart.pdf

[4] Zynq-7000 All Programmable SoC: ZC702 Evaluation Kit and Video and Imaging Kit (ISE Design Suite 14.5) Getting Started Guide UG926 (v4.0) May 14, 2013
http://www.xilinx.com/support/documentation/boards_and_kits/zc702_zvik/14_5/UG926_Z7_ZC702_Eval_Kit.pdf

[5] 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

[6] 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

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

[8] 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

[9] 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/

5. Evaluation version of the AMP demo on ZYNQ with (8xSIMD) EdkDSP package.

The enclosed **Evaluation version of the AMP demo on ZYNQ with UTIA (8xSIMD) EdkDSP package** can be downloaded from UTIA www pages free of charge and used for evaluation asymmetric multiprocessing on ZYNQ [2] together with the four UTIA (8xSIMD) EdkDSP accelerators.

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

10 precompiled designs with UTIA (8xSIMD) EdkDSP accelerators for Xilinx ZC702 board [2], compiled in Xilinx EDK 14.5 and XPS (ISE) 14.5 [6]. 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 evaluation package includes SDK 14.5 [7] 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 Xilinx ZC702 board [2].

The evaluation package includes this compiled library:

libwal.a	EdkDSP api (SDK 14.5, MicroBlaze) for EdkDSP accelerators on ZC702 board.

This library has no time restriction. The evaluation license is provided by UTIA only for the use with the family of UTIA EdkDSP accelerators designed for the Xilinx ZC702 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.

These binary applications have no time restriction. The user of the evaluation package has 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 10 precompiled designs for the Xilinx ZC702 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 ZC702 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 ZC702 board without the need to install the UTIA compiler binaries and the Ubuntu (x86 PC) OS image under the VMware Player.

On email request to kadlec@utia.cas.cz , UTIA will send 2 DVD CDs (8GB) with the Ubuntu (x86 PC) image for the VMware Player free of charge.

6. AMP projects for ZYNQ ZC702 board with evaluation version of (8xSIMD) EdkDSP for the Artemis EMC2 project partners.

The **release version of the AMP HW/SW projects on ZYNQ ZC702 board with evaluation version of the (8xSIMD) EdkDSP for the partners in the Artemis EMC2 project [9]** 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 also send to the EMC2 project partner (by email) and by the standard mail the invoice for:

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

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

10 precompiled designs with UTIA (8xSIMD) EdkDSP accelerators for Xilinx ZC702 board, compiled in Xilinx EDK 14.5 and XPS (ISE) 14.5 [6]. 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 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 all 10 EDK design projects demonstrating the asymmetric processing on ZYNQ and the evaluation versions of the UTIA (8xSIMD) EdkDSP accelerators provided in form of netlisted pcores generated in Xilinx ISE 14.5 [6]:

bce_fp11_1x8_0_axiw_v1_10_a
bce_fp11_1x8_0_axiw_v1_20_a
bce_fp11_1x8_0_axiw_v1_30_a
bce_fp11_1x8_0_axiw_v1_40_a
bce_fp12_1x8_0_axiw_v1_10_a
bce_fp12_1x8_0_axiw_v1_20_a
bce_fp12_1x8_0_axiw_v1_30_a
bce_fp12_1x8_0_axiw_v1_40_a

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** [9] have license from UTIA to integrate these evaluation netlists into their own ISE 14.5 designs and to compile them to unlimited number of bit-streams for the asymmetric multiprocessing designs on Xilinx ZYNQ FPGAs. This 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 14.5 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 ZC702 board.

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

libwal.a	EdkDSP api (SDK 14.5, MicroBlaze) for EdkDSP accelerators on ZC702 board.

This library has no time restriction. The evaluation license is provided by UTIA only for the use with the family of UTIA EdkDSP accelerators designed for the Xilinx ZC702 board. Source code of this libray 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 (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.

These binary applications have no time restriction. The Artemis EMC2 project partners have 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 10 precompiled designs for the Xilinx ZC702 board. The 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 Xilinx ZC702 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 ZC702 board without the need to install the UTIA compiler binaries and the Ubuntu (x86 PC) OS image under the VMware Player.

The release package deliverables also includes two DVDs 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 on Windows 7 (32bit or 64bit) in the VMware Player.

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 package.

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

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

10 precompiled designs with UTIA (8xSIMD) EdkDSP accelerators for Xilinx ZC702 board [2], compiled in Xilinx EDK 14.5 and XPS (ISE) 14.5 [6]. 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 ZC702 without limitations of the evaluation package.

The release package includes source code of all 10 EDK design projects demonstrating the asymmetric processing on ZYNQ. The UTIA (8xSIMD) EdkDSP accelerators are provided in the form of netlist pcores generated in Xilinx ISE 14.5 [6]:

bce_fp11_1x8_0_axiw_v1_10_a
bce_fp11_1x8_0_axiw_v1_20_a
bce_fp11_1x8_0_axiw_v1_30_a
bce_fp11_1x8_0_axiw_v1_40_a
bce_fp12_1x8_0_axiw_v1_10_a
bce_fp12_1x8_0_axiw_v1_20_a
bce_fp12_1x8_0_axiw_v1_30_a
bce_fp12_1x8_0_axiw_v1_40_a

These UTIA (8xSIMS) EdkDSP netlist pcores have **no HW limit on number of vector operations.** The user of the release package has license from UTIA to integrate these netlists into its own ISE 14.5 designs and to compile them to unlimited number of bit-streams for the asymmetric multiprocessing designs on Xilinx ZYNQ FPGAs. This 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 14.5 SW projects in source code for MicroBlaze as described in this application note. Projects support the family of UTIA (8xSIMD) EdkDSP accelerators for Xilinx ZC702 board [2].

The release package includes the library:

libwal.a	EdkDSP api (SDK 14.5, MicroBlaze) for EdkDSP accelerators on ZC702 board.

This library has no time restriction. The evaluation license is provided by UTIA only for the use with the family of UTIA EdkDSP accelerators designed for the Xilinx ZC702 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.

These binary applications have no time restriction. The user of the evaluation package has 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 10 precompiled designs for the Xilinx ZC702 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 ZC702 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 ZC702 board without the need to install UTIA compiler binaries and the Ubuntu (x86 PC) OS under the VMware Player.

The release package deliverables also includes two DVDs 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 on Windows 7 (32bit or 64bit) in the VMware Player.

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.