





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

Asymmetric Multiprocessing with MicroBlaze, EdkDSP Accelerator and Toshiba Sensor Video Processing for low cost Zynq on TE0720-03-1CF SoM on TE0701-05 Carrier

Jiří Kadlec, Zdeněk Pohl, Lukáš Kohout

<u>kadlec@utia.cas.cz</u>, <u>xpohl@utia.cas.cz</u>, <u>kohoutl@utia.cas.cz</u> phone: +420 2 6605 2216 UTIA AV CR, v.v.i.

Revision history:

Rev.	Date	Author	Description
1	09.07.2016	Jiří Kadlec	Evaluation package for Xilinx SDK 2015.4
2	14.07.2016	Jiří Kadlec	Updated data in conclusions section

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

Asymmetric Multiprocessing with MicroBlaze, EdkDSP Accelerator and Toshiba Sensor Video Processing for low cost Zynq on TE0720-03-1CF SoM on TE0701-05 Carrier	1
1. Summary	4
1.1 Key features	4
1.2 Project sh01: EdkDSP accelerator with edge detection in single HLS accelerator	6
1.3 Project sh02: EdkDSP accelerator with edge detection in two HLS accelerators	7
1.4 Project sh03: EdkDSP accelerator with edge detection in three HLS accelerators	8
1.5 Project md01: EdkDSP accelerator with motion detection in HLS accelerators	2
2. Installation of evaluation package1	4
2.1 Import of SW projects in Xilinx SDK 2015.41	4
2.2 HW setup	8
2.3 Test demos 1	9
2.4 EdkDSP C compiler	9
3. Conclusions	1
4. References	2
5. Evaluation license	3
Disclaimer 3	4

2/34



http://zs.utia.cas.cz



Table of figures

Figure 2: Project sh01 - Edge detection with single HW accelerator
Figure 3: Project sh01 - Energy per frame reduction and used HW resources. 6 Figure 4: Project sh02 - Edge detection with two HW accelerators 7 Figure 5: Project sh02 - Energy per frame reduction and used HW resources. 7
Figure 4: Project sh02 - Edge detection with two HW accelerators 7 Figure 5: Project sh02 - Energy per frame reduction and used HW resources 7
Figure 5: Project sh02 - Energy per frame reduction and used HW resources
Figure 6: Project sh03 - Edge detection with three HW accelerators
Figure 7: Project sh03 - Energy per frame reduction and used HW resources
Figure 8: Project sh01 – EdkDSP and edge detection in one Arm SW function
Figure 9: Project sh01 – EdkDSP and edge detection in one HLS accelerator
Figure 10: Project sh02 – EdkDSP and edge detection with two Arm SW functions
Figure 11: Project sh02 – EdkDSP and edge detection with two HLS accelerators
Figure 12: Project sh03 – EdkDSP and edge detection with three Arm SW functions
Figure 13: Project sh03 – EdkDSP and edge detection with three HLS accelerators 11
Figure 14: Project md01 - Motion detection with single HW accelerator data path
Figure 15: Project md01 - Energy per frame reduction and HW resources
Figure 16: Project md01 – EdkDSP and motion detection in Arm SW functions
Figure 17: Project md01 – EdkDSP and motion detection with single HLS accelerator path
Figure 18: Select the SDK Workspace 14
Figure 19: Import Existing Projects into Workspace15
Figure 20: Select "Copy projects into workspace" and finish the import of all projects
Figure 21: All projects are compiled in debug mode 17
Figure 22: Serial cables USB based for Arm and Jtag. RS232 with Pmod for Microblaze
Figure 23: Serial console. Reset board and stop auto boot by any key
Figure 24: Download bitstream to the PL part of Zynq
Figure 25: Select demo application for debug21
Figure 26: Demo app is booted to Arm and the debugger is waiting on the first executable line
Figure 27: Arm is waiting on HW Mutex for the MicroBlaze start
Figure 28: Select the Microblaze application for debug
Figure 29: MicroBlaze application is loaded and debugger stops on the first instruction
Figure 30: Arm is running. It indicates the number of frames per second
Figure 31: Microblaze is running. It indicates MFLOPs
Figure 32: Accelerated edge detection Toshiba sensor and Zynq with EdkDSP
Figure 33: Edge detection (Sobel filter) output on Full HD monitor

3/34



http://zs.utia.cas.cz



1. Summary

1.1 Key features

This application note describes HW platform performing integration of the runtime reprogrammable EdkDSP floating point accelerator with edge detection and motion detection video processing for Toshiba Full HD colour video sensor with fixed resolution (1920x1080p60) with low cost Zynq device.

- The Xilinx low cost Zynq device xa7z020-1c has two Arm Cortex A9 processors (operating at 666 MHz), memory controller with two levels of caches and also with high performance DDR3 memory access ports. It provides also the programmable logic area used for:
 - UTIA EdkDSP (8xSIMD) floating point processor (operating at 100 MHz) connected to Xilinx MicroBlaze 32bit processor (operating at 75 MHz).
 - Input chain of video processing IPs is connecting Full HD Toshiba video sensor to input video frame buffers. The input video DMA (VDMA) controller is operating at 150 MHz.
 - Area reserved for HLS HW accelerators and data movers defined in Xilinx SDSoC 2015.4 environment. These accelerators can be controlled from Arm Cortex A9 C programs compiled in SDK 2015.4 C projects. These HLS accelerators are operating at 120 MHz.
 - Chain of output video processing IPs is connecting output frame buffers to the Full HD display connected by HDMI cable. The output VDMA controller is operating at 150 MHz.
- UTIA EdkDSP is 8xSIMD floating point accelerator reprogrammable in runtime by change of firmware of build in PicoBlaze6 8bit controller. This is serving as a scheduler of vector operations performed in the EdkDSP is 8xSIMD floating point processor data paths. This scheduler is programmed by simple C programs compiled by simple C compiler and assembler, respecting the minimal resources of the PicoBlaze6 controller.
- UTIA EdkDSP is 8xSIMD floating point accelerator is controlled by the 32bit MicroBlaze processor. The MicroBlaze processor is executing C programs from the DDR3 memory. It executes complex C algorithms. Algorithms can benefit from execution of selected operations effectively on the EdkDSP coprocessor connected to the MicroBlaze by local dual ported memories. MicroBlaze C programs can take benefit of overlap of data communication from DDR3 to the EdkDSP dual-ported memories with parallel computations in the EdkDSP accelerator.
- Platform includes also the video processing chain of IPs controlled by Arm Cortex A9 processor.
- Arm Cortex A9 processor of Xilinx Zynq is performing initialisation and synchronisation of the video processing chain. Program and the FPGA image is downloaded to the board from the Xilinx SDK 2015.4 via USB JTAG to the 1GB DDR3 located on the Zynq system on module. System can be also started directly from the SD card. Arm processor initiates the IP cores in the programmable logic (PL) part of the Zynq. It also initiates the Toshiba video sensor and the video output to the Full HD monitor with fixed 1920x1080p60 resolution and standard Full HD pixel clock 148.5 MHz.

4/34



ŪΠΑ

http://zs.utia.cas.cz





Figure 1: Toshiba Full HD evaluation platform HW.

Details of the Full HD video processing video chain:

- Raw video data are provided by the Toshiba video sensor.
- Data are processed into the YCrCb 16 bit per pixel format and stored by Video DMA (VDMA) to input video frame buffers (VFBs) defined in the DDR3.
- HW DMA controller(s) send data from/to the VFBs to the processing accelerators. Clock is 120 MHz.

Projects described in next section are summarising the energy per frame measured on the platform for Different accelerated image processing algorithms as defined by individual C projects in these main configurations:

- 1. MicroBlaze with EdkDSP coprocessor is computing Floating point FIR filter (in parallel to the dedicated video processing accelerator chain).
- 2. MicroBlaze with EdkDSP coprocessor is computing Floating point LMS adaptive filter (in parallel to the dedicated video processing accelerator chain).
- 3. MicroBlaze is computing in SW (only with its Floating point unit) FIR or LMS filter (in parallel to the dedicated video processing accelerator chain) but EdkDSP accelerator s not used.
- 4. MicroBlaze and EdkDSP is not present in the PL logic and only the dedicated video processing accelerator chain is processing the Full HD video from the Toshiba sensor.

SW figures indicate the energy/pixel consumed by the complete system in case of computation in Arm. C/C++ code was compiled with -O3 optimisation (but without NEON) in the SDSoC 2015.4 environment. No HLS accelerators present.

The evaluation designs with HLS accelerators have been created from these C/C++ functions in SDSoC 2015.4.



http://zs.utia.cas.cz



1.2 Project sh01: EdkDSP accelerator with edge detection in single HLS accelerator



Figure 2: Project sh01 - Edge detection with single HW accelerator

Energy per pixel (nJ/p = nano Joule/pixel) **Reduced:** EdkDSP FIR SW: 515.54 nJ/p HW: 103.89 nJ/p **4.96 x** EdkDSP LMS SW: 513.96 nJ/p HW: 103.57 nJ/p **4.96 x** Filter by MB SW: 504.49 nJ/p HW: 101.63 nJ/p **4.96 x** Without MB SW: 487.91 nJ/p HW: 98.23 nJ/p **4.97 x**





Figure 3: Project sh01 - Energy per frame reduction and used HW resources.



6/34

http://zs.utia.cas.cz

1.3 Project sh02: EdkDSP accelerator with edge detection in two HLS accelerators



Figure 4: Project sh02 - Edge detection with two HW accelerators

Energy per pixel (nJ/p = nano Joule/pixel)Reduced: te0701-05 te0720-1c: FPS EdkDSP FIR SW: 512.48 nJ/p HW: 64.33 nJ/p 7.97 x 7,34 SW EdkDSP LMS SW: 510.90 nJ/p HW: 64.14 nJ/p 7.97 x 60 HW Filter by MB SW: 501.44 nJ/p HW: 62,98 nJ/p 7.96 x 0 20 Without MB SW: 484.88 nJ/p HW: 60.96 nJ/p 7.99 x 40 60 te0701-05 te0720-1c: Slices [%] te0701-05 te0720-1c: BRAMs [%] 66,64 51,79 SW/ SW/ 67,5 94,54 HW HW 0 0 20 100 20 40 60 80 100 40 60 80 te0701-05 te0720-1c: LUTs [%] te0701-05 te0720-1c: FFs [%] 45,46 26,23 SW SW 64.63 42.23 HW HW 0 20 40 60 80 0 20 40 60 100 100 80

Figure 5: Project sh02 - Energy per frame reduction and used HW resources.



7/34

http://zs.utia.cas.cz

AV ČR, v.v.i. All disclos

1.4 Project sh03: EdkDSP accelerator with edge detection in three HLS accelerators



Figure 6: Project sh03 - Edge detection with three HW accelerators



Figure 7: Project sh03 - Energy per frame reduction and used HW resources.

8/34



http://zs.utia.cas.cz

© 2016 ÚTIA AV ČR, v.v.i. All disclosure and/or reproduction rights reserved

Akademie ved Ceske republiky Ústav teorie informace a automatizace AV ČR, v.v.i.



Figure 8: Project sh01 – EdkDSP and edge detection in one Arm SW function



Figure 9: Project sh01 – EdkDSP and edge detection in one HLS accelerator



9/34

http://zs.utia.cas.cz



Figure 10: Project sh02 – EdkDSP and edge detection with two Arm SW functions



Figure 11: Project sh02 – EdkDSP and edge detection with two HLS accelerators



10/34

http://zs.utia.cas.cz



Figure 12: Project sh03 – EdkDSP and edge detection with three Arm SW functions



Figure 13: Project sh03 – EdkDSP and edge detection with three HLS accelerators



http://zs.utia.cas.cz

1.5 Project md01: EdkDSP accelerator with motion detection in HLS accelerators



Figure 14: Project md01 - Motion detection with single HW accelerator data path

Energy per pixel (nJ/p = nano Joule/pixel) **Reduced:** EdkDSP FIR SW: 3227.0 nJ/p HW: 121.4 nJ/p **26.58 x** EdkDSP LMS SW: 3217.2 nJ/p HW: 121.1 nJ/p **26.57 x** Filter by MB SW: 3158.4 nJ/p HW: 118.9 nJ/p **26.56 x** Without MB SW: 3055.4 nJ/p HW: 115.2 nJ/p **26.52 x**





Figure 15: Project md01 - Energy per frame reduction and HW resources.

12/34



http://zs.utia.cas.cz

© 2016 ÚTIA AV ČR, v.v.i. All disclosure and/or reproduction rights reserved



Figure 16: Project md01 – EdkDSP and motion detection in Arm SW functions



Figure 17: Project md01 – EdkDSP and motion detection with single HLS accelerator path

signar j	department of Drocessing		http://zs.utia.cas.cz
		13/34	
ŪTIA	Akademie věd České republiky Ústav teorie informace a automatizace AV ČR, v.v.i.		© 2016 ÚTIA AV ČR, v.v.i. All disclosure and/or reproduction rights reserved

2. Installation of evaluation package

2.1 Import of SW projects in Xilinx SDK 2015.4

Unzip the evaluation package to directory of your choice. The directory C:\VM_07 will be used in this application note. C:\VM_07\t20c1tm1_V54_IMPORT

Create empty directory for Xilinx SDK workspace. C:\VM_07\t20c1tm1

Start Xilinx SDK 2015.4 and select the directory for the SDK 2015.4 workspace. See Figure 18. Select C:\VM_07\t20c1tm1

Workspace	Launcher	
Select a wo	rkspace	
Xilinx SDK st Choose a we	ores your projects in a folder called a workspace. orkspace folder to use for this session.	
Workspace:	C:\VM07\t20c1tm1	- Browse
🔲 Use this a	s the default and do not ask again	OK Cancel

Figure 18: Select the SDK Workspace

14/34

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

File -> Import -> General -> Existing Projects into Workspace Click on Next button. See Figure 19.



http://zs.utia.cas.cz



SOK Import	
Select Create new projects from an archive file or directory.	
Select an import source:	
type filter text	
 General Archive File Existing Projects into Workspace File System Preferences C/C++ Git Finstall Remote Systems Run/Debug Team Tracing 	
Rext > Finish	Cancel

Figure 19: Import Existing Projects into Workspace

Type directory with projects to be imported. See Figure 20.

C:\VM_07\t20c1tm1_V54_IMPORT

Set the "**Copy projects into workspace**" check box. Click on Finish button. See Figure 20.

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.



15/34

http://zs.utia.cas.cz



sok Import			- 0 X
Import Projects Select a directory to searc	h for existing Eclipse projects.		
Select root directory:	C:\VM07\t20c1tm1_V54_IMPORT	•	Browse
Select archive file:		-	Browse
Projects:			
 edkdsp (C:\VM07 md01_bsp (C:\VM md01_edkdsp_fp1 md01_rows_fixed. md01_rows_fixed. md01_standalone sh01_bsp (C:\VM0 sh01_edkdsp_fp12 sh01_hw_platforn sh01_rows_fixed_1 sh01_rows_resize_ sh01_standalone_ sh02_bsp (C:\VM0 sh02_hw_platforn sh02_rows_fixed_1 sh03_bsp (C:\VM0 sh03_edkdsp_fp12 sh03_hw_platforn 	<pre>\t20c1tm1_V54_IMPORT\edkdsp) 107\t20c1tm1_V54_IMPORT\md01_bsp) L2_1x8_all (C:\VM07\t20c1tm1_V54_IMPORT\md01_edkdsp_fp12_1x8_all) m_0 (C:\VM07\t20c1tm1_V54_IMPORT\md01_nows_fixed_100) e_bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\md01_standalone_bsp_0) 07\t20c1tm1_V54_IMPORT\sh01_bsp) 2_1x8_all (C:\VM07\t20c1tm1_V54_IMPORT\sh01_edkdsp_fp12_1x8_all) n_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh01_edkdsp_fp12_1x8_all) n_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh01_rows_fixed_100) 25_to_100 (C:\VM07\t20c1tm1_V54_IMPORT\sh01_rows_resize_25_to_100) bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh01_rows_resize_25_to_100) bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh01_standalone_bsp_0) 07\t20c1tm1_V54_IMPORT\sh02_edkdsp_fp12_1x8_all) n_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_fixed_100) 25_to_100 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_resize_25_to_100) bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_resize_25_to_100) bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_fixed_100) 25_to_100 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_fixed_100) 25_to_100 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_fixed_100) 25_to_100 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_resize_25_to_100) bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_fixed_100) 25_to_100 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_resize_25_to_100) bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_resize_25_to_100) bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_resize_25_to_100) bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh02_rows_resize_25_to_100) bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh03_etkdsp_fp12_1x8_all) n_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh03_hw_platform_0)</pre>		Select All Deselect All Refresh
 sh03_rows_resize_ sh03_standalone_ options Search for nested progodies Copy projects into working sets Add project to working working sets: 	25_to_100 (C:\VM07\t20c1tm1_V54_IMPORT\sh03_rows_resize_25_to_100) bsp_0 (C:\VM07\t20c1tm1_V54_IMPORT\sh03_standalone_bsp_0) jects orkspace	-	Select
?	< Back Next > Finish		Cancel

Figure 20: Select "Copy projects into workspace" and finish the import of all projects.



Son C/C++ - Xilinx SDK			
File Edit Source Refactor Navigate Search Project	Xilinx Tools Run Window Help		
📑 🕶 🗔 💿 📥 🛛 🕲 🕶 🔦 🖛 🗖 🐹 🖬 🖉	X @ + @ + C + Ø + Ø + Q + Q +	۳ 🗉 🖌 👏	
		Ouick Accord	
		Quick Access	
🔁 Project Explorer 🛛 📄 🔄 🔽 🦳 🔽			🗄 0 🛛 💿 M 🖓 🗖
> 🔝 edkdsp			
⊳ 🏙 md01_bsp			An outline is not available.
md01_edkdsp_fp12_1x8_all			
▷ [] md01_hw_platform_0			
md01_rows_rixed_100			
sh01 bsp			
b Se sh01 edkdsp fp12 1x8 all			
▷ 1 sh01_hw_platform_0			
b Sh01_rows_fixed_100			
sh01_rows_resize_25_to_100			
b 100 sh01_standalone_bsp_0			
▷ 200 sh02_bsp ▷ 200 sh02_bsp			
snoz_edkasp_ipiz_ixo_all			
Sh02_rws_platorn_0 Sh02_rws_fixed 100			
sh02_rows_resize_25_to_100			
b 10 sh02_standalone_bsp_0			
⊳ 🏙 sh03_bsp			
▷ 2 sh03_edkdsp_fp12_1x8_all			
▷ U sh03_hw_platform_0			
▷ >>>> sh03_rows_fixed_100			
snu3_rows_resize_25_to_100			
p 2 subj_standalone_osh_o			
	lame 🖉 Tacke 🔲 Concolo 🕅 🔲 Proportion 🗖 SDK Terminal		
N C Linux TCE Agent	ales to display at this time		
QEMU TcfGdbClient	bles to display at this time.		19:30:41 INFO : Lau A
			19:50:42 INFO : XSD
			Ψ.
			<

Figure 21: All projects are compiled in debug mode.

17/34

SDK 2015.4 compiles SW of all imported demos in debug mode.



http://zs.utia.cas.cz

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

2.2 HW setup

HW setup is using commercially accessible components [1], [2], [3], [4], [5], [6]:

TE0720-03-1CF; Part: XC7Z020-1CLG484C; 1 GByte DDR;Price: \in 219,00 [1]Heatsink for TE0720, spring-loaded embedded;Price: \in 19.00 [2]**TE0701-05 Carrier Board** for Trenz Electronic 7 Series;Price: \in 249.00 [3]**AES-FMC-HDMI-CAM-G** FMC card with HDMI I/O and CAM interfacePrice: \$250.00 [4]**Toshiba Industrial 1080P60 Camera Module**Price \$229.00 [5]**PmodRS232:** Serial converter & interfacePrice \$13.54 [6]

HW Options:

TE0701-05 can be replaced by **TE0701-04**

(Same Price, both boards from Trenz) [3].

Trenz TE0701-04 or TE0701-05 carriers require modifications to run the FMC Imageon carrier AES-FMC-HDMI-CAM-G with Zynq TE0720-03-1CF system on module. The modification is related to the swapped polarity of the differential clock signal for the FMC board. Evaluation HW systems with carriers TE0701-04 or TE0701-05 provided by UTIA have these modifications already done.

UTIA can implement these HW modifications for the original Trenz TE0701-04 and TE0701-05 carriers. This requires written e-mail request to <u>kadlec@utia.cas.cz</u>. Request will be first confirmed by UTIA. The interested party has to cover the cost of shipment of the carrier board to/from UTIA. Modification can be done in 5 working days and it is offered free of charge.

18/34



ŪTĬA

http://zs.utia.cas.cz

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

2.3 Test demos

To test demos follow these steps:

- Insert the Toshiba Full HD video sensor to the connector on the Imageon board.
- Connect Full HD (or DVI) monitor by HDMI cable to the HDMI OUT on the Imageon FMC card.
- Switch the monitor ON.
- Connect the carrier board by USB-to-microUSB cable to PC to support JTAG serial link and the standard serial terminal.
- Connect the PmodRS232 Serial converter & interface module to the carrier board as indicated in Fig. xx . Connect the RS232 cable to COM1 serial terminal of your PC. This serial line will support serial terminal for the Microblaze processor.



Figure 22: Serial cables USB based for Arm and Jtag. RS232 with Pmod for Microblaze .

- Connect power supply (DC 12V).
- Open and configure the standard serial terminal client (PuTTY or similar) on PC for the Arm serial terminal (USB emulated).

(Speed: 115200 baud; Data bits: 8; Stop bits: 1; Parity: None; Flow control: None).

- Open and configure the standard serial terminal client (PuTTY or similar) on PC for MicroBlaze It is COM1. (Speed: 115200 baud; Data bits: 8; Stop bits: 1; Parity: None; Flow control: None).
- Reset the board. Board will start first stage boot loader from internal flash as set up by Trenz. It is writing messages to the serial terminal. On request, "Hit any key to stop autoboot" type any key to stop the auto-boot of Linux.
- If you need to switch-off the power, close first the serial terminal on the PC. This will help to avoid problems



19/34

http://zs.utia.cas.cz





Figure 23: Serial console. Reset board and stop auto boot by any key.

Download bitstream to the board. Demo **sh01_rows_fixed_100** will be used as an example. The **bitstream.bit** for demo **sh01** is located in the directory:

C:\VM_07\t20c1tm1\sh01_hw_platform_0



20/34

http://zs.utia.cas.cz

Program FPGA					-\$-
BMM/MMI file is n write_mem_info <1	ot specified, BRA filepath> in Vivao	M will not be initialized. do.	To generate it, use	command	ÓЮ
Hardware Configura	tion				
Hardware Platform:	sh01_hw_platfo	rm_0	•		
Connection:	Local		•	New	
Device:	Auto Detect Select				
Bitstream:	C:\VM07\t20c1	tm1\sh01_hw_platform_()\bitstream.bit	Search	Browse
Partial Bitstream					
BMM/MMI File:				Search	Browse
Software Configurati	ion				
Processor		ELF/MEM File to Initial	ize in Block RAM		
microblaze_0		bootloop			
~					

Figure 24: Download bitstream to the PL part of Zynq.

Select Program to download the bitstream to the PL part of Zynq via the USB cable in JTAG mode.

Debug Configurations		×
Create, manage, and run configurations		
bype filter text Performance Analysis Target Communication Framework Xilimx C/C++ application (GDB) \$\$\$\$ shol_rows_fixed_100 Debug \$	Name: sh01_rows_fixed Target Setup Debug Type: Standald Connection: Local Device: Auto De	L100 Debug Application & Source StDIO Connection Debugger Options Common ne Application Debug New tect Select
	Hardware platform: Processor: Bitstream file: Initialization file:	sh01_hw_platform_0 p57_contexa9_0 bitstream.bit p57_init.tcl Search Browse Browse Browse Browse Browse
	Reset Processor Program FPG/ Run ps7_init Run ps7_post_cor Enable Cross-Trig	Summary of operations to be performed Following operations will be performed before launching the debugger. I. Reset processor. Reset processor. I. Run ps7_init. (Only first time after System reset or board power ON) I. Run ps7_post_config. (Only first time after System reset or board power ON) I. C.\VM07t20c1tml\sh01_rows_fixed_100.Debug\sh01_rows_fixed_100.elf' will be downloaded to the processor 'ps7_cortexa9_0'
Filter matched 6 of 16 items		Apply
0		Debug Close

Figure 25: Select demo application for debug.



Debug - sh01 rows fixed 100/src/main.c - Xilinx SDK		
File Edit Source Refactor Navigate Search Project Xiliny Tools Run Window Help		
	*** • • • • • • • • • • • • • • • • • •	
	پ، ۱۵۰۰ ۲۵۰۰ ۲۵۰۰ ۲۵۰۰ ۲۵۰۰ ۲۵۰۰ ۲۵۰۰ ۲۵۰۰	iick Access
Å Dehug ☆	(X)= Variables 💥 🤷 Breaknoints 1010 Registers 🕅 XM	ID Console 🖬 XSDB Console 🛋 Modules 👘 🗖
Scholl roug fixed 100 Debug (Viling C/C++ application (GDD))	valuates to breakpoints intracisters and this	
XMD Target Debug Agent (13.07.16.7/49) (Suspended)		2 🕫 🖃 🖉 🛪 % 🗖 🖸 🖄
A thread [1] (Suspended: Breaknoint hit.)	Name	Value
$\equiv 1 \text{ main}() \text{ main}(c;287 0x00100 ce0)$	b ConfigPtr	0x001a23e0
arm-xilinx-eabi-gdb (13.07.16 7:49)	(×)= Status	1630728
C:\VM07\t20c1tm1\sh01_rows_fixed_100\Debug\sh01_rows_fixed_100.elf (13.07.16 7:49) [Console not co	(x)= TimeoutCount	6
	(x)= val	4
	(x)= status	1/21384
		· · · · · · · · · · · · · · · · · · ·
		A
		Ŧ
	•	4
🔁 main.c 🕱	- 8	🗄 Outline 🛛 🕞 🔩 💘 💊 🙀 🗸 🗖 🗖
⊖ /*		stdio h
* Init mutex and start microblaze		stdlib.h
*/		yil cache.h
White Castle *CastleDta		yparameters.h
XStatus Status:		platform.h
u32 TimeoutCount = 0;		ui.h
		img_filters.h
⊖ /* * toolway and firmentian data in the during and firmentian toble.		# I2C_BASE
* Lookup configuration data in the device configuration table.		# DDR_MEM_BASE
* driver instance.		sds_lib.h
*/		clock_start : unsigned long long
ConfigPtr = XMutex_LookupConfig(MUTEX_DEVICE_ID);		clock_end_processing : unsigned long long
1+ (ConfigPtr == (XMutex_Config *) NULL) {		clock_end_frame : unsigned long long
}		# TIME_STAMP_INIT
	-	# TIME_STAMP_1 +
· · · · · · · · · · · · · · · · · · ·	Þ	۰ III +
🍓 Target Conn 🛛 😑 🗖 📮 Console 🕅 ⁄ Tasks 📮 SDK Terminal 🔝 Problems 💽 Executables	🚺 Memory 🗖 🗖 📗 SDK Log 🛛	
	💴 🖻 🖛 📑 👻 07:40:01 INFO : Launch	ing XSDB server: xsdb.bat C:/pf/Xilinx/SC
▷ And	1\sh01 rows fixed 100\ 07:40:02 INFO : XSDB set	erver has started successfully.
▶ > Linux TCF Agent Process STDIO not connected to console.	07:43:21 INFO : Connect	ted to target on host '127.0.0.1' and por
De DEMU TcfGdbClient If you'd like to see UART output in this console, please mo	dify STDIO sett: 07:43:21 INFO : 'target	ts -set -filter {jtag_cable_name =~ "JTA(
	07:49:06 INFO : PPGA CO	it is completed
	07:49:06 INFO : ps7_po	st_config is completed.
	07:49:06 INFO : Process	sor reset is completed for ps7_cortexa9_(
	·	v
<	• •	•

Figure 26: Demo app is booted to Arm and the debugger is waiting on the first executable line.

22/34



http://zs.utia.cas.cz

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



Figure 27: Arm is waiting on HW Mutex for the MicroBlaze start.

Image: Second	Name: sh01_edkdsp_fp12_1x8_all Debug Image: Target Setup Application Source Target Setup Connection: Common Debug Type: Standalone Application Debug Image: Connection: Connection: Local Image: Connection: New Device: Auto Detect Select Image: Connection: Image: Connectio: Image:
Filter matched 7 of 17 items	Apply Revert
0	Debug Clos



We are downloading program for Microblaze while Arm is already running.

- Unselect "Run ps7_init"
- Unselect "Run ps7_post_config"
- Select No reset

Click on "Apply" button.

Click on "Debug" to download the **sh01_edkdsp_fp12_1x8_all.elf** to DDR3 as program for MicroBlaze.

The debugger will download this code by JTAG (connected to PC by the USB cable shared with the serial terminal) and stop Microblaze at the first executable instruction. See Figure 29.

Sex Debug - sh01_edkdsp_fp12_1x8_all/src/fp12_1x8_all.c - Xilinx SDK			
File Edit Source Refactor Navigate Search Project Xilinx Tools Run Window Help			
····································			
	-		
🏇 Debug 🛛 🙀 😰 🗖 🗖	🗱 Variables 🔀 🤏 Breakpoints 👬 Registers 🐹 XM	1D Console 🔳 XSDB Console 🛋 Modules 🛛 🖓 🗖	
Sh01_rows_fixed_100 Debug [Xilinx C/C++ application (GDB)]		🗄 🐗 🕞 🕈 💥 🙀 📑 🖆 💙	
XMD Target Debug Agent (13.07.16 7:49)	Name	Value	
Thread [1] (Running)	A i timer 0 Timer	{} ·	
arm-xilinx-eabi-gdb (13.07.16 7:49)	▷ ➡ ConfigPtr	0x9c44c86b	
ε. (vivio) (zociani (shor_lows_ined_too(bebug(shor_lows_ined_too.en (15.07.107.45) [consolenot co ε shot edddsn fn12 1y8 all Debug [Xilinx C/C++ application (GDB)]	(×)= Status	-67305473	
A See XMD Target Debug Agent (13.07.16 7:52) (Suspended)	(x)= TimeoutCount	3218809855	
▲ 🔊 Thread [1] (Suspended: Breakpoint hit.)	(x)= val	301989912	
1 main() fp12_1x8_all.c:2997 0x3000c258	(III III	7378630/067	
📕 mb-gdb (13.07.16 7:52)		A	
C:\VM07\t20c1tm1\sh01_edkdsp_fp12_1x8_all\Debug\sh01_edkdsp_fp12_1x8_all.elf (13.07.16 7:52) [Con			
	4	\k	
€ fp12_1x8_all.c ⊠	<u> </u>	🗄 Outline 🛛 📄 🖓 💘 🔍 🗮 👋 🗖	
⊖int main() {	*	stdio.h	
<pre>// static XIntc intc;</pre>		■ platform.h	
<pre>static XTmrCtr axi_timer_0_Timer;</pre>		++ print(cnar): void	
init nlatform().		xtmrctr.h	
		tmrctr_header.h	
XMutex_Config *ConfigPtr;		# TMRCTR_DEVICE_ID	
XStatus Status; u32 TimeoutCount = 0:		# TIMER_COUNTER_0	
		TimerCounter : XTmrCtr	
⊖ /* * to but out in the in the duing out in the bull * to but out in the bull		xmutex.h	
* LOOKup configuration data in the device configuration table. * Use this configuration info down below when initializing this		# MUTEX_DEVICE_ID	
* driver instance.		Mutex : XMutex	
*/ Confights - YMutov LookunConfig(MUTEY DEV/ICE ID);		yparameters.h	
if (ConfigPtr == (XMutex Config *) NULL) {	-	uwal.h	
	4	🚽 wal_bce_jk.h 👻	
📥 Target Conn 👋 😑 🗖 🔲 Console 🖄 🖉 Tasks 🖻 SDK Terminal 🚇 Problems 📭 Executables	Memory 🖵 🗖 🗐 SDK Log 🖄		
	c1tm1\sh01 edkden fn 07:40:02 INFO : Launch	erver has started successfully.	
De maravare server sing eukopping ind an been grain kerver application (sobj) Civimo/Lacitmismoi_eakopping of neurol and or sob server nas started successfully.			
QEMU TcfGdbClient If you'd like to see UART output in this console, please model	dify STDIO sett: 07:43:21 INFO : 'targe	ts -set -filter {jtag_cable_name =~ "JTA(
	07:43:24 INFO : FPGA configured successfully with bitstream "C		
07:49:06 INFO : ps7_post_config is completed.			
07:49:06 INFO : Processor reset is completed for ps7_cortexa9_(
	· · · · · · · · · · · · · · · · · · ·	- T	
	•		

Figure 29: MicroBlaze application is loaded and debugger stops on the first instruction.

- Arm Thread [1] is running.
- Microblaze Thread [1] is currently suspended at breakpoint hit. See Figure 29.

Click on the |> icon to start the execution of MicroBlaze. The handshake of Arm and Microblaze on HW mutex IP is completed and both processors start to run uninterrupted.

Arm will initiate the Toshiba Full HD video sensor and all Video processing IP cores. It controls in SW status of VDMA units and sets correct pointers to the active video frame buffers. Video

http://zs.utia.cas.cz

24/34



and back to output video frame buffers by HW data mover IPs. These IPs are set-up by the Arm SW via the Axi-Lite.

Input HW data movers act as HW masters controlling the DMA engines moving data from DDR3 as input to the chain(s) of HLS IP cores. Output HW data movers act as HW masters controlling the DMA engines moving data from the output of chain(s) of HLS IP cores to DDR3 output video frame buffers. See

B COM8 - PuTTY	- • • × •	🚱 COM8 - PuTTY
zynq-uboot> ARMCPU0: place 0xb8000000 at start of MB	0 vectors	AXI VDMA - Checking Error Flags
		S2MM_DMASR - ErrIrg
		S2MM_DMASR - SOFLateErr
***********		S2MM_DMASR - DMAIntErr
* Signal Processing Dept. *		AXI_VDMA - Clearing Error Flags
* UTIA AV CR, v.v.i. *		AXI_VDMA - Partial Register Dump (uBaseAddr = 0x43000000):
* Iosniba Sensor in IMAGEON *		PARKPIR = 0x03020000
* [[Sobel eage detection]] *		$S_{2}MM_{DM}CP = 0_{V}000100CP$
* C:/S/t20c1tm1/bio *		S2MM DMASR = 0x00011000
* MicroBlaze 8xSIMD EdkDSP *		S2MM STRD FRMDLY = $0x00001000$
*****		S2MM START ADDR0 = 0x20000000
		S2MM START ADDR1 = $0x20800000$
		S2MM START ADDR2 = 0x21000000
IIC IMAGEON Initialization OK		S2MM_HSIZE = 0x00000F00
HDMIO Initialization OK		$S2MM_VSIZE = 0x00000438$
Initialize Timing Controller 1920x1080p60 OK		
Initialize VDMA Common Init TX Init OK		$MM2S_DMACR = 0x0001008B$
TCM3232DB Init Chin ID: 0x0854 OK		MM2S STRD FRMDLY = 0x00001000
Running sobel		MM2S START ADDR0 = 0x20000000
AXI VDMA - Partial Register Dump (uBaseAddr = 0x4300)	0000):	MM2S START ADDR1 = $0x20800000$
PARKPTR = 0x00000000		MM2S START ADDR2 = $0x21000000$
		MM2S $HSIZE$ = $0x00000F00$
S2MM_DMACR = 0x000100CB		MM2S VSIZE = 0x00000438
$S2MM_DMASR = 0x00014810$		
S2MM_STRD_FRMDLY = 0x00001000		S2MM_HSIZE_STATUS= 0x00000000
$S2MM_START_ADDR0 = 0x20000000$		S2MM_VSIZE_STATUS= 0x00000000
S2MM START ADDR1 = 0x20800000		
S2MM + START ADDR2 = 0x210000000		Trace processing time: 0 Total FDS: 0 019081
S2MM VSIZE = 0x00000438		Image processing time: 18642698, Total FPS: 35.750568
		Image processing time: 18638366, Total FPS: 35.759415
MM2S DMACR = $0 \times 0001008B$		Image processing time: 18638128, Total FPS: 35.759941
MM2S DMASR = 0x00011000	E	Image processing time: 18637946, Total FPS: 35.760372
MM2S_STRD_FRMDLY = 0x00001000		Image processing time: 18637990, Total FPS: 35.760277
MM2S_START_ADDR0 = 0x20000000		Image processing time: 18638034, Total FPS: 35.760139
$MM2S_START_ADDR1 = 0x20800000$		Image processing time: 18637992, Total FPS: 35.760250
MM2S START ADDR2 = 0x21000000		Image processing time: 18637954, Total FPS: 35.760326
MM2S MSIZE = 0x0000000000000000000000000000000000		Image processing time: 18637054 Total FPS: 35.760170
		Image processing time: 18638000, Total FPS: 35.760212
S2MM HSIZE STATUS= 0x00000000		Image processing time: 18637862, Total FPS: 35.760452
S2MM_VSIZE_STATUS= 0x00000000		Image processing time: 18637966, Total FPS: 35.760326
		Image processing time: 18637940, Total FPS: 35.760357
AXI_VDMA - Checking Error Flags		Image processing time: 18637834, Total FPS: 35.760574
S2MM_DMASR - ErrIrq		Image processing time: 18637862, Total FPS: 35.760502
S2MM_DMASR - SOFLateErr		Image processing time: 18637966, Total FPS: 35.760262
SZMM_DMASR - DMAINTEFF		Image processing time: 1863/918, lotal FPS: 35./60422
AXI_VDMA - Partial Register Dump (uBaseAddr = 0x4300)	00001 :	Image processing time: 18637916, Total FPS: 35.760365
$= \frac{1}{PARKPTR} = 0x03020000$		Image processing time: 18637904, Total FPS: 35.760422
		Image processing time: 18637946, Total FPS: 35.760384
S2MM_DMACR = 0x000100CB		Image processing time: 18637898, Total FPS: 35.760475
$S2MM_DMASR = 0x00011000$		Image processing time: 18637988, Total FPS: 35.760239
S2MM_STRD_FRMDLY = 0x00001000		Image processing time: 18637898, Total FPS: 35.760429
S2MM START ADDR0 = 0x20000000		Image processing time: 18637918, Total FPS: 35.760395
$\frac{S2MM}{START} = 0x20800000$		Image processing time: 18638020 Total FPS: 35.760525
S2MM + SIZE = 0x00000F00	-	Image processing time: 18637906, Total FPS: 35.760437

Figure 30.



Ústav teorie informace a automatizace AV ČR, v.v.i.

http://zs.utia.cas.cz

25/34

🚱 COM8 - PuTTY		B COM8 - PuTTY	x
zynq-uboot> ARMCPU0: place 0xb8000000 at start of MB0 vectors	~	AXI VDMA - Checking Error Flags	-
		S2MM_DMASR - SOFLateErr	
*****		S2MM_DMASR - DMAIntErr	
* Signal Processing Dept. *		AXI_VDMA - Clearing Error Flags	
* UTIA AV CR, v.v.i. *		AXI_VDMA - Partial Register Dump (uBaseAddr = 0x43000000):	
* Toshiba Sensor In IMAGEON *		PARKPTR = 0x03020000	
* [[Sobel edge detection]] *			
* Full HD HDMI Out IMAGEON *		S2MM DMACR = 0x000100CB	
* C:/S/t20cltml/hio *		S2MM DMASR = 0x00011000	
* MicroBlaze 8xSIMD EdkDSP *		$S2MM_STRD_FRMDLY = 0x00001000$	
******************		S_{2MM} START ADDR0 = 0x20000000	
		S2MM START ADDRI = 0x20000000000000000000000000000000000	
TIC IMAGEON Initialization OK		S2MM HST7F = 0x00000F00	
HDMIO Initialization OK		$S2MM_VST7F = 0x00000438$	
Initialize Timing Controller 1920x1080n60 OK			
Initialize VDMA Common Init TX Init OK		MM25 DMACR = 0x0001008B	
RX Init OK		MM2S DMASR = 0x00011000	
TCM3232PB Init Chip ID: 0x0854 OK		MM2S STRD FRMDLY = 0×00001000	
Running sobel		MM2S START ADDR0 = 0x20000000	
AXI VDMA - Partial Register Dump (uBaseAddr = 0x43000000):		MM2S START ADDR1 = 0x20800000	
$= 0 \times 00000000$		MM2S START ADDR2 = 0x21000000	
		MM2SHSIZE = 0x00000F00	
S2MM DMACR = 0×000100 CB		MM2SVSIZE = 0x00000438	
$S2MM_DMASR = 0x00014810$			
S2MM_STRD_FRMDLY = 0x00001000		S2MM_HSIZE_STATUS= 0x00000000	
$S2MM_START_ADDR0 = 0x20000000$		S2MM_VSIZE_STATUS= 0x00000000	
$S2MM_START_ADDR1 = 0x20800000$			
S2MM_START_ADDR2 = 0x21000000		Parking started	
$S2MM_{HSIZE} = 0x00000F00$		Image processing time: 0, Total FPS: 0.019081	
$S2MM_VSIZE = 0x00000438$		Image processing time: 18642698, Total FPS: 35.750568	
		Image processing time: 18638366, Total FPS: 35.759415	
MM2S DMACR = 0x0001008B		Image processing time: 18638128, Total FPS: 35.759941	
MM2S DMASR = 0x00011000	=	Image processing time: 18637946, Total FPS: 35.760372	
MM2S STADT ADDDO = 0x20000000000000000000000000000000000		Image processing time: 1863/990, lotal FPS: 35./602//	
MM2S = START ADDR0 = 0x20000000		Image processing time: 18632002, Total FPS: 35./60139	
MM2S START ADDR2 = 0x21000000		Image processing time: 18637952, lotal FPS: 35.760230	
MM2S HSIZE = 0x00000F00		Image processing time: 18638006 Total FPS: 35.760320	
MM2S VSIZE = 0x00000438		Image processing time: 18637954, Total FPS: 35,760296	
		Image processing time: 18638000. Total FPS: 35.760212	
S2MM HSIZE STATUS= 0x00000000		Image processing time: 18637862, Total FPS: 35.760452	
S2MM_VSIZE_STATUS= 0x00000000		Image processing time: 18637966, Total FPS: 35.760326	
		Image processing time: 18637940, Total FPS: 35.760357	
AXI_VDMA - Checking Error Flags		Image processing time: 18637834, Total FPS: 35.760574	=
		Image processing time: 18637862, Total FPS: 35.760502	-
S2MM_DMASR - SOFLateErr		Image processing time: 18637966, Total FPS: 35.760262	
S2MM_DMASR - DMAIntErr		Image processing time: 18637918, Total FPS: 35.760422	
AXI_VDMA - Clearing Error Flags		Image processing time: 18637914, Total FPS: 35.760410	
AXI_VDMA - Partial Register Dump (uBaseAddr = 0x43000000):		Image processing time: 18637916, Total FPS: 35.760365	
PARKPTR = 0x03020000		Image processing time: 18637904, Total FPS: 35.760422	
		Image processing time: 1863/946, Total FPS: 35.760384	
$S2MM_DMACR = 0x000100CB$		Image processing time: 18637898, Total FPS: 35.760475	
S2MM STRD FRMDLY = 0x00001000		Image processing time: 1863/988, lotal FPS: 35.760239	
S2MM STAPT ADDPO = 0x20000000000000000000000000000000000		Image processing time: 1863/898, lotal FPS: 35./60429	
S2MM START ADDRU = 0x20000000		Image processing time: 1063/918, lotal FPS: 35./60395	
$S2MM_START_ADDR1 = 0x21000000$		Image processing time: 18638020 Total FPS: 35.760525	
S2MM HSTZE = 0x00000F00	-	Image processing time: 18637906, Total FPS: 35.760437	
		The second secon	

Figure 30: Arm is running. It indicates the number of frames per second.

The MicroBlaze processor executes in parallel program from DDR3 and communicated firmware and data to the (8xSIMD) EdkDSP floating point accelerator.

It is testing basic floating point operations and compares EdkDSP results with MicroBlaze floating point results.

In next stage it programs EdkDSP to perform FIT filter and LMS adaptive filter. The performance of the combination of MicroBlaze with EdkDSP accelerator is measured by HW timer instantiated as Microblaze AXI-Lite IP core. See Figure 31.



🛃 COM1 - PuTTY

Initialize TmrCtr for axi timer 0... MB0 : (EdkDSP 8xSIMD) Write firmware ... MB0 : (EdkDSP 8xSIMD) Capabilities1 = 13ffff MBO : (HW FP unit) Far-end signal ... MB0 : (EdkDSP 8xSIMD) FIR room response ... 923 MFLOPs MB0 : (HW FP unit) Add near-end signal ... MB0 : (EdkDSP 8xSIMD) LMS Identification ... 603 MFLOPs MB0 : (HW FP unit) LMS Identification ... 2 MFLOPs MB0 : (EdkDSP 8xSIMD) OK MB0 : (EdkDSP 8xSIMD) Write firmware ... MB0 : (EdkDSP 8xSIMD) Capabilities1 = 13ffff MB0 : (EdkDSP 8xSIMD) VZ2A 'worker1' OK MB0 : (EdkDSP 8xSIMD) VB2A 'worker1' OK MB0 : (EdkDSP 8xSIMD) VZ2B 'worker1' OK MB0 : (EdkDSP 8xSIMD) VA2B 'worker1' OK MB0 : (EdkDSP 8xSIMD) VADD 'worker1' OK MB0 : (EdkDSP 8xSIMD) VADD BZ2A 'worker1' .. OK MB0 : (EdkDSP 8xSIMD) VADD_AZ2B 'worker1' .. OK MB0 : (EdkDSP 8xSIMD) VSUB 'worker1' OK MB0 : (EdkDSP 8xSIMD) VSUB BZ2A 'worker1' .. OK MB0 : (EdkDSP 8xSIMD) VSUB AZ2B 'worker1' .. OK MB0 : (EdkDSP 8xSIMD) VMULT 'worker1' OK MB0 : (EdkDSP 8xSIMD) VMULT BZ2A 'worker1' . OK MB0 : (EdkDSP 8xSIMD) VMULT AZ2B 'worker1' . OK MB0 : (EdkDSP 8xSIMD) VPROD 'worker1' OK MB0 : (EdkDSP 8xSIMD) VMAC 'worker1' OK MB0 : (EdkDSP 8xSIMD) VMSUBAC 'worker1' OK MB0 : (EdkDSP 8xSIMD) VPROD S8 'worker1' ... OK MB0 : (EdkDSP 8xSIMD) VDIV 'worker1' OK MB0 : (EdkDSP 8xSIMD) Write firmware ... MB0 : (EdkDSP 8xSIMD) Capabilities1 = 13ffff MBO : (HW FP unit) Far-end signal ... MBO : (EdkDSP 8xSIMD) FIR room response ... 922 MFLOPs MBO : (HW FP unit) Add near-end signal ... MB0 : (EdkDSP 8xSIMD) LMS Identification ... 603 MFLOPs

Figure 31: Microblaze is running. It indicates MFLOPs.

27/34



http://zs.utia.cas.cz

_ D X

.

© 2016 ÚTIA AV ČR, v.v.i. All disclosure and/or reproduction rights reserved

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



Figure 32: Accelerated edge detection Toshiba sensor and Zynq with EdkDSP.



Figure 33: Edge detection (Sobel filter) output on Full HD monitor.



http://zs.utia.cas.cz

28/34

- All evaluation demos can be also compiled into release versions with optimisation set to -O2 or -O3. These optimisations can be set for Arm and for MicroBlaze.
- Demo **sh01_rows_fixed_100** works on complete frame with single HW accelerator data path.
- Demo sh01_rows_resize_25_to_100 works with identical HW. But SW scales dynamically the number of lines to be processed. This is scaling from ¼ of frame to the complete frame. Part of the frame which is not processed is automatically propagating the input video signal via the cyclic structure of 8 video frame buffers. The HW data movers are instructed about the number of lines to be processed. SW is writing this information to an AXI-lite configuration register of the data mover IP core.
- Demos sh02_rows_fixed_100 and sh02_rows_resize_25_to_100 work with 2 data paths.
- Demos sh03_rows_fixed_100 and sh03_rows_resize_25_to_100 work with 3 data paths.
- Demos md01_rows_fixed_100 works with one HW video processing chain with fixed set of processed lines.

2.4 EdkDSP C compiler

This section describes how to use the UTIA EdkDSP C compiler. It cross-compiles (on PC) simple C programs for the PicoBlaze6 controller. This controller acts as programmable finite state machine in the (8xSIMD) EdkDSP accelerator. It is setting the wide instructions for the 8xSIMD floating point data path of the EdkDSP accelerator.

The evaluation package includes also precompiled firmware files. These files can be used without the need to install the EdkDSP C compiler to your PC.

The UTIA EdkDSP C compiler is included as Ubuntu binaries. The "VMware player" software with compatible Ubuntu image is needed to run the UTIA EdkDSP C compiler on Windows 7 PC.

The Ubuntu image used in UTIA needs two DVD (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 <u>kadlec@utia.cas.cz</u> 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.

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.

29/34

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/t20c1tm1/edkdsp



http://zs.utia.cas.cz



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/t20c1tm1/edkdsp/a\$

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

ca_fp11.sh a

This will compile and assemble four C firmware programs to header files with the firmware binary code for the EdkDSP accelerator:

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

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

edkdsp/a/ fill_FA1101P0_program_store.h edkdsp/a/ fill_FA1101P1_program_store.h edkdsp/a/ fill_FA1124P0_program_store.h edkdsp/a/ fill_FA1124P0_program_store.h

to the SDK project directory (in case of sh01_edkdsp_fp12_1x8_all):

C:\VM_07\t20c1tm1\sh01_edkdsp_fp12_1x8_all\src

Recompile the MicroBlaze project "**sh01_edkdsp_fp12_1x8_all**". The compiled firmware for the (8xSIMD) EdkDSP will be used by the MicroBlaze C code of the demo as data for the runtime (re)configurations of the (8xSIMD) EdkDSP accelerator PicoBlaze6 controller.

The change of firmware is demonstrated by the runtime change of firmware for computation of FIR and LMS filters in the EdkDSP accelerator.



http://zs.utia.cas.cz

30/34

3. Conclusions

This application note documents following general observations and conclusions:

- Programmable logic part of the Zynq xc7z020-2I device is capable implement in parallel the UTIA (8xSIMD) EdkDSP floating point accelerator together with the Full HF video processing chain for the Toshiba color sensor.
- The total power consumption for the HW accelerated video processing in Full HD (measured at the 12V DC power supply) is up to 8.18 W for HW accelerated edge detection with MB and EdkDSP computing FIR filter in floating point. This is relatively high power for passive cooling in small space even if the dedicated passive heat sink is used.
- The total power consumption for the SW solution without HLS Video IPs (measured at the 12V DC power supply) is up to 7.90 W for SW edge detection with MB and EdkDSP computing FIR filter in floating point. This is also relatively high power for passive cooling.
- The energy per pixel savings for the complete system are significantly reduced for the HW accelerated designs with HLS IP accelerators. Energy per pixel reduction up to 26 x can be reached, for chained HLS IP cores (motion detection).
- Main source of the energy per pixel saving is the increased frame rate of the video processing.
- The combination of 32bit MicroBlaze with the (8xSIMD) EdkDSP floating point accelerator brings additional capability to compute in floating point (single precision) with performance 0.925 GFLOP/s (in case of the FIR filter) at the expense of relatively moderate increase of total power consumption:
 - 7.76 W without MicroBlaze + (8xSIMD) EdkDSP 0 GFLOP/s
 - 8.18 W with MicroBlaze + (8xSIMD) EdkDSP 0.925 GFLOP/s (this is + 420 mW)
- Instantiation of MicroBlaze + (8xSIMD) EdkDSP takes significant part of Zynq PL resources. This limits the possibilities for design of video systems with increased number of parallel video processing chains.
- Bill of material for the system [1]-[6] is €980,00.
- The Toshiba Industrial 1080P60 Camera Module is connected to the FMC card by connector. This
 open space for possible replacement of the Toshiba module by Python 1300 colour video sensor
 module (provided by Avnet) with resolution 1280x1024p60. Designs included in this evaluation
 package and the corresponding SW projects are designs only for support Toshiba module.

This application note documents how designs debugged and developed in the high level SDSoC 2015.4 environment can be exported to the end-user in form of SDK 2015.4 projects.

Enclosed SDK 2015.4 projects provide space for the end-user to make some SW adaptations and customisations of the final application without the need to disclose to the end-user complete low level details about used IP cores Vivado 2015.4 project and the SDK 2015.4 board support package.



31/34

http://zs.utia.cas.cz



4. References

- [1] TE0720-03-1CF; Part: XA7Z020-1CLG484C; 1 GByte DDR; Price: €219,00. <u>https://shop.trenz-electronic.de/de/TE0720-03-1CF-Xilinx-Zynq-module-XC7Z020-1CLG484C-com.-temp.-range-1-Gbyte</u>
- [2] Heatsink for TE0720, spring-loaded embedded; Price: €19.00. https://shop.trenz-electronic.de/en/26922-Heatsink-for-TE0720-spring-loaded-embedded?c=38
- [3] TE0701-05 Carrier Board for Trenz Electronic 7 Series; Price: €249.00. <u>https://shop.trenz-electronic.de/en/TE0701-05-Carrier-Board-for-Trenz-Electronic-7-Series</u>
- [4] AES-FMC-HDMI-CAM-G Price: \$250.00. http://products.avnet.com/shop/en/ema/3074457345623664802
- [5] Toshiba Industrial 1080P60 Camera Module; Price \$229.00. http://zedboard.org/sites/default/files/product_briefs/PB-AES-CAM-TOSH-1080P-G-v5-web.pdf
- [6] PmodRS232: Serial converter & interface; Price €13.54. <u>https://shop.trenz-electronic.de/de/23331-PmodRS232-Serial-converter-und-interface?c=215</u>

32/34

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



ŪTĬA

http://zs.utia.cas.cz

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

5. Evaluation license

The **evaluation version of the package** can be downloaded from UTIA www pages free of charge for evaluation of EdkDSP accelerator with HW accelerated edge detection and motion detection algorithms for the Toshiba Full HD video sensor [5] on TE0720-03-1CF module [1] located on TE0701-05 carrier [3] with FMC card [4].

The evaluation package includes SDK 2015.4 SW projects with C source code for Arm Cortex A9 processor (32bit) in standalone mode, C source code for MicroBlaze and C source code for the EdkDSP Picoblaze6 controller.

The evaluation package includes these static libraries for Arm Cortex A9 processor (32bit) for standalone mode:

libfmc_imageon.a	SDK 2015.4 UTIA static library with interface functions for video IP cores
libwal.a	SDK 2015.4 UTIA static library with EdkDSP API for MicroBlaze
libsh01.a	SDSoC 2015.4 static library for HW accelerator in project sh01
libsh02.a	SDSoC 2015.4 static library for HW accelerator in project sh02
libsh03.a	SDSoC 2015.4 static library for HW accelerator in project sh03
libmd01.a	SDSoC 2015.4 static library for HW accelerator in project md01
libmd02.a	SDSoC 2015.4 static library for HW accelerator in project md02

These libraries have no time restriction. Source code of these libraries is not provided in this evaluation package.

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 2015.4 SW projects with source code for MicroBlaze processor and ARM processor. SW projects support the family of UTIA (8xSIMD) EdkDSP accelerators for the Trenz TE0720-03-1Q Xilinx Zynq module [1] on Trenz TE701-05 Carrier Board board [3].

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, nontransferable 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 Xilinx TE0720-03-1CF module on TE0701-05 carrier 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 TE0720-03-1CF module on TE0701-05 carrier board without the need to install the UTIA compiler binaries and the Ubuntu image under the VMware Workstation 12 Player [7]. On email request to <u>kadlec@utia.cas.cz</u>, UTIA will send DVD with the Ubuntu image for the VMware Workstation 12 Player [7] free of charge.

33/34



http://zs.utia.cas.cz



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.

34/34



http://zs.utia.cas.cz

