Application Note

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

DSP Library for UTIA BCE platform architecture

Zdeněk Pohl, Jan Kloub
zdenek.pohl@utia.cas.cz, kloub@utia.cas.cz

Revision history

Rev.	Date	Author	Description
0	25.10.2011	Z.P.	Initial draft
1			
2			

1. Introduction

The library contains a set of the DSP operations implemented for worker cores of the UTIA BCE multicore platform [1,2,3,4]. It contains the optimized solutions of the FIR [5] and LMS [6,7] filters and of the fully pipelined error-feedback least-square lattice filter [8]. In addition, the FFT radix 2 and radix 4 implementations and DVB-T2 P1 symbol detection [9] are included.

Enclosed package contains simple pre-compiled benchmarking applications which are providing detailed profiling information about each DSP operation in the library.

2. Description

The UTIA master-worker platform for FPGA consists from the MicroBlaze, a soft processor core for FPGA designed by Xilinx, and from the coprocessor cores denoted as *Basic Computing Elements* (BCE) or *workers.* The workers designed by UTIA can support customized functions, use different arithmetic etc. For that reason, the workers are characterized by family identification, number of SIMD layers and by their special capabilities. The reconfiguration of the worker for different job is done by changing its firmware. Thus, the operations in this library are in fact firmware with the information about necessary data communication between main memory and worker.

The workers used for implementation of the DSP operations in this library are using for its operation complex worker and floating point worker with two independent memory layers (SIMD). Table 1 summarizes the firmwares in the DSP library and their requirements to worker cores.

Operation	Worker Family	Min. SIMD Layers	Required Capabilities
FIR	float 32-bit	2	VCPY, DPROD
LMS	float 32-bit	2	VCPY, VADD, VSUB, VMUL, DPROD
Lattice	float 32-bit	1	VCPY, VMUL, VMAC, VMSAC, VDIV
FFT	complex 16-bit	1	VORDS, VMUL, VADD, VSUB, VCPY
P1 detect	complex 16-bit	1	CSUMA, VCPY, VMUL, VMULC, CSUM, SUM, VSUB

Table 1: Requirements of the DSP operations on the worker capabilities

Parameters and performance figures of provided operations are introduced in Table 2. The information in the table was obtained from the operation parameters and from the profiling information.

Operation	Dimension [-]	Execution Time [ns/N]	Latency [-]	Performance [MFLOPS]
FIR	2048	5.47	1	365.5
LMS	2046	20.23	1	98.9
Lattice	256	459.5	N	58.7
FFT radix 2	2048	2231	1	24.7
FFT radix 4	2048	1315	1	56.7
P1 detect	4096	177	1	189.5

Table 2: Parameters of supported operations

3. Used tools and resources

Xilinx tools: iMPACT, XMD
Hardware: Ethernet cable, RS232 cable
Special hardware: ML506 Xilinx Development Board, Xilinx JTAG cable
Software: Matlab, serial console, FTP

4. How to Run

This step-by-step guide assumes you have prepared and installed all tools and resources from Section 3.

1. Power on the ML506 board, connect the JTAG cable, Ethernet cable between the board and PC, and serial cable if you intend to work from serial console.
2. Use the impact to download “download.bit” FPGA bitstream. It can be found in the ‘bit_ml506’ directory.
3. Start the XMD and connect to the board. Download and run petalinux image image.bin from the same directory as FPGA bitstream. XMD console commands are:
    a. connect mb mdm
    b. dow –data image.bin 0x50000000
    c. con 0x50000000
4. Connect to the board using the FTP client. The IP address of the board is 192.168.0.102.
5. Transmit all demo files located in directory ‘bin’. All files should be transmitted to the /tmp directory on the board
6. Transmit the drivers from ‘driver’ directory to the ‘/tmp’ directory on the board.
7. Login to the petalinux from serial console. Login is ‘root’, password ‘root’.
8. Continue on the console
9. Change to the /tmp directory
10. Load the drivers by commands in following order:
    a. insmod uio_bce.ko
    b. insmod uio_xdma.ko
11. Change permissions to the run script: chmod 700 run
12. Start the demo by running ‘run’ script in the ‘/tmp’ dir.
13. Wait all computations are finished profiling information is written to files.
14. Connect by the FTP and copy all profile files from board’s ‘/tmp’ dir to PC directory ‘profile’
15. Start Matlab in 'matlab' directory.
16. Run ‘show_all’ script to view the profiles.

4.1 Interpretation of the Profile Analysis Result

Figure 1 shows the sample FIR profile analysis provided by the 'plot_analysis.m' script. First table displays data of our interest, the clean times of all operations needed to filter one block of input data. The second table shows the execution times if the resource management middle layer would be used for programming the Linux application. Each time value is given in clock cycles. In our demo, the length of one clock cycle is 10 ns (100MHz clock frequency).

The 'datain' field of the table stands for time to store input data block from DDR memory to the worker local memory. The 'statein' is time for needed to initialize state of the fir filter and its parameters. The 'codein' time reflects the time for booting the FIR algorithm firmware to the worker. Clean time of the computation within the worker is denoted as the 'exec' time. The polling value says how long the calling code must wait for the result. Analogically to the input, the output data time and output state times are presented respectively.

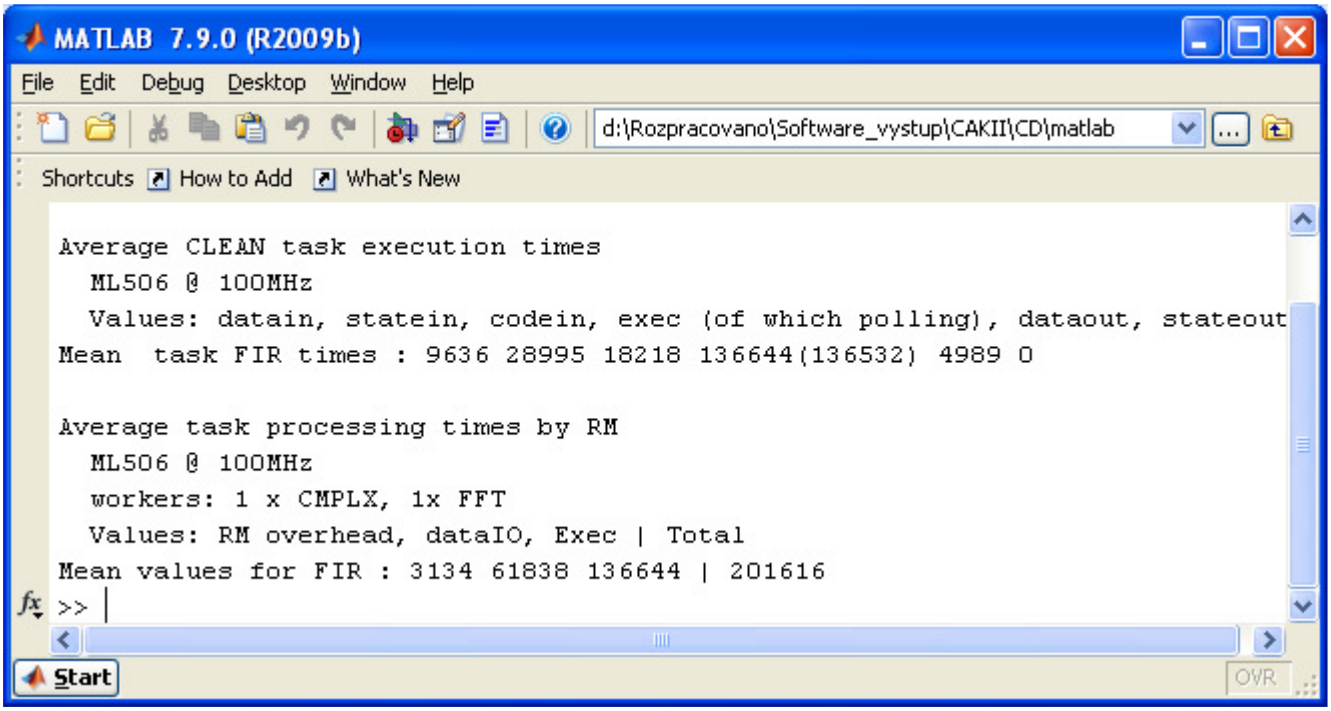

Figure 1: FIR filter profile analysis

From the profile data for the FIR algorithm it can be seen, that the execution time covers the data input and output time, thus the filter can be continuously fed by input data while the computation of previous data block is running. By the detailed observation of all provided operations, it can be seen, that also for other DSP operations can be I/O communication masked behind the execution of the worker.

5. Package contents

```
cdrom    - bin           Binary executables of demo algorithms
                         for petalinux, run script
         - bit_ml506     ML506 FPGA bitstream and DDR image
                         of the petalinux
```

```
- doc            This documentation
- driver         Linux drivers for workers
- matlab         Matlab scripts for profile analysis
- profiles       pre-computed profile examples
```

6. Troubleshooting

If the demo is started by the 'run' command on petalinux without the worker and xdma driver pre-loaded, the demo can't work properly. For recovery it is needed to restart from point 2 of Section 4.

7. References

[1] J. Kadlec, R. Bartosinski, and M. Danek, "Accelerating MicroBlaze floating point operations," in *Int. Conf. Field Prog. Logic and Applications*. IEEE, 2007, pp. 621–624.

[2] J. Kadlec, M. Danek, and L. Kohout, "Proposed architecture of configurable, adaptable SoC," in *The Inst. of Eng. and Tech. Irish Signals and Systems Conf., ISSC*, Galway, IE, 2008.

[3] J. Kadlec, "Design flow for reconfigurable Microblaze accelerators," in *Int. Workshop Reconfigurable Communication Centric System-on-Chips*, 2008.

[4] M. Danek, J. Kadlec, R. Bartosinski, and L. Kohout, "Increasing the level of abstraction in FPGA-based designs," in *Int. Conf. Field Prog. Logic and Applications*. IEEE, 2008, pp. 5–10.

[5] Rabiner, Lawrence R., and Gold, Bernard, 1975: Theory and Application of Digital Signal Processing (Englewood Cliffs, New Jersey: Prentice-Hall, Inc.) ISBN 0139141014

[6] Monson H. Hayes: *Statistical Digital Signal Processing and Modeling,* Wiley, 1996, ISBN 0-471-59431-8

[7] Paulo S.R. Diniz: *Adaptive Filtering: Algorithms and Practical Implementation,* Kluwer Academic Publishers, 1997, ISBN 0-7923-9912-9

[8] A. H. C. Carezia, P. M. S. Burt, M. Gerken, M. D. Miranda, and M. T. M. Da Silva. A stable and efficient DSP implementation of a LSL algorithm for acoustic echo cancelling. In ICASSP '01, volume 2, pages 921–924. IEEE, May 7–11, 2001.

[9] EN 302 755 V1.1.1 - Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2), http://www.etsi.org/deliver/etsi_en/302700_302799/302755/01.01.01_60/en_302755v010101p.pdf