Application Note

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

Self-adaptive LMS filter

ÆTHER SVP/SEP demo

*zdenek.pohl@utia.cas.cz (ÆTHER FP6-IST-**027611**)*

Contents

Revision history

Rev.	Date	Author	Description
0	15.9.2009	Z.P.	Initial revision
1			
2			

1. Introduction

This demo is functional sample of the UTIA HW accelerators integrated to the computational cloud by the SVP/SEP distributable framework API developed under the AETHER project. It focuses on demonstration of distributed computing which can dynamically adapt to the changing availability of processing elements in the computational cloud.

The integration of FPGA hardware to the AETHER SVP/SEP layer allows building large computational cloud from multiple heterogenous FPGA boards. Consequently, the client applications can benefit from huge parallelisms and work offloads in such system.

2. Description

In this demonstrator we use the LMS filter in standard noise-cancellation setup, i.e. filter estimates unknown room response from known input and output sound waveforms. In the noise cancellation it is assumed that the room response can change over the time. The speed how the LMS filter can track the changes is given by its parameter μ, see Figure 1.

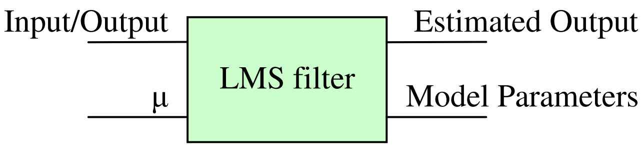


Figure 1: LMS filter

The quality of estimation can be improved by implementation of the self-adaptive LMS filter. The filter uses multiple LMS filter instances computing estimation with one μ “hypothesis” each. After that, the results are compared and the best is chosen. Consequently, new “hypoteses” for μ are formulated. The block diagram of the self-adaptive LMS filter is in Figure 2. It can be seen that for maintaining four μ “hypotheses” the same number of LMS filters must be computed.

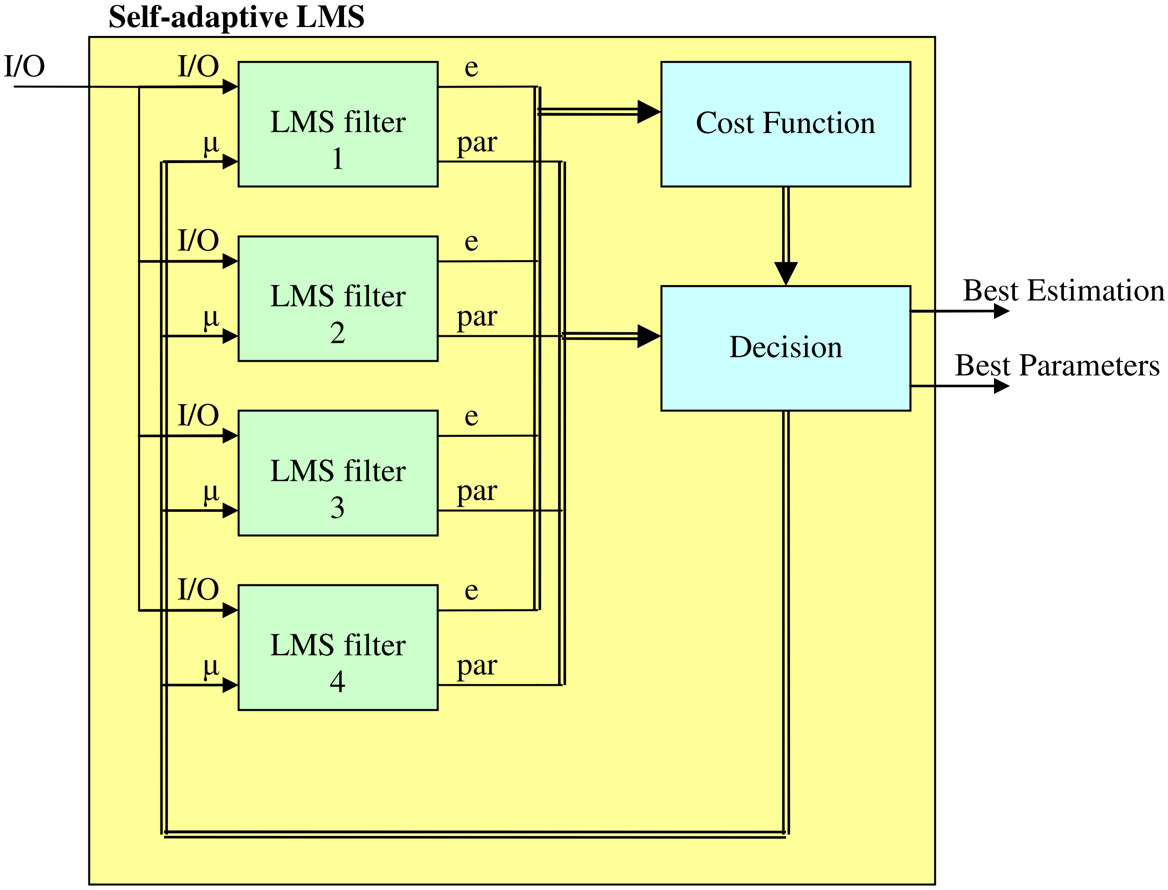


Figure 2: Self-adaptive LMS

2.1 Embedded platform

The basic processing element of the demo is the RISC microprocessor (MicroBlaze) with the UTIA HW accelerators [1]. The accelerator is capable to perform SIMD floating-point operations according to its configurable microcode. One Microblaze can connect up to 15 such HW accelerators, depending on the FPGA device size.
The microcodes for HW accelerators supporting following operations were implemented:

- VPROD for computation of the cost function
- LMS filter for computation of one μ “hypothesis”
- SA-LMS, internally optimized self-adaptive LMS filter (Figure 2)

MicroBlaze processor is running Petalix OS [2]. It makes benefit from its ability to run SVP/SEP framework with TCP/IP connectivity. Moreover, the VGA interface was added to the system in order to implement visualization of processes happening in the demo.

Complete embedded design for running the demo is prepared for Xilinx ML402 board with Virtex4 SX35 FPGA device.

2.2 Software components

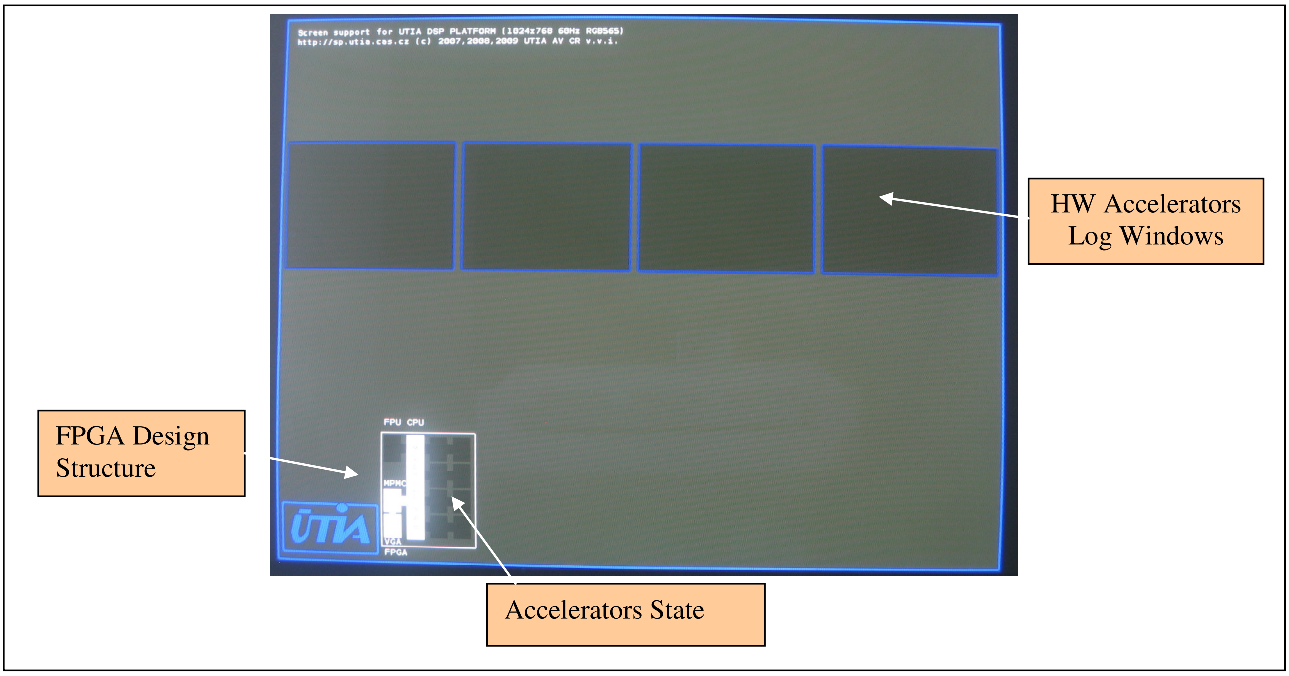
The demo consists from four components. First three are the “SANEs” which are servers providing computational places in the SVP/SEP distributed framework. The last one is “client” which is using the SVP/SEP framework API and implements the filter from Figure 2 using the resources found in the computational cloud.

SW SANE
The SANE provides basic operations for client application to run. Its implementation is pure in software, thus it can run on the Microblaze which is not equipped by the HW accelerators. It provides one computational place (supported operations: LMS, VPROD; HW places = 0, SW places = 1).

HW SANE
The second SANE is interfacing four HW accelerators to the distributed computational cloud. The size of the design consisting from MicroBlaze and 4 HW accelerators fits the size of FPGA on demo target board (Xilinx ML402). When this sane joins the cloud, the client application will be able to use four additional independent computational places accelerated by HW. Each place can be configured to support VPROD or LMS function. In contrary to the SW SANE, the switching between VPROD and LMS requires reconfiguration of the accelerator microcode (supported operations: LMS, VPROD; HW places = 4, SW places = 0).

Preferred SANE
The third SANE provides one HW accelerator for SA-LMS (the SA-LMS uses four HW accelerators hardwired inside, thus in this case we can provide only one SA-LMS on the ML402 board), the SANE is also providing VPROD and LMS (supported operations: SA-LMS, VPROD, LMS; HW places = 1, SW places = 0)

The example VGA output from the HW SANE is shown in Figure 3. It consists from four LOG windows each dedicated to one HW accelerator. Simple FPGA icon at the bottom highlights active regions while the demo is running. The accelerator painted in green stands for the one which is computing while the one painted in red is reconfigured.



Screen support for UTIA DSP PLATFORM (1024x768 60Hz RGB565)
http://sp.utia.cas.cz (c) 2007,2008,2009 UTIA AV CR v.v.i.

Figure 3: HW SANE VGA Display

Client

The client application is implemented using the SVP/SEP API. It asks for necessary resources and implements the self-adaptive LMS algorithm shown in Figure 2. It is programmed to work with a different number of resources according to functions actually available from the SEP layer.
Minimal requirements of the client:

1x LMS, 1x VPROD (i.e. at least one SW SANE)

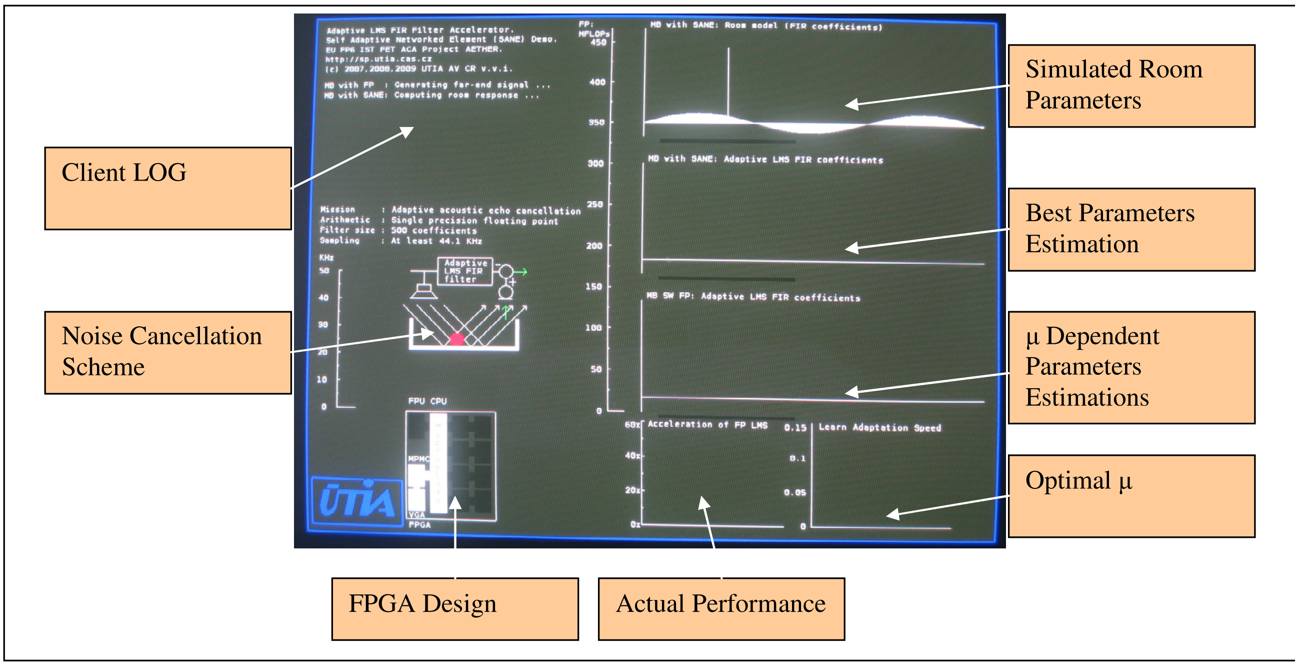
In this case the client will execute 4 LMS filters sequentially, then VPROD operation is demanded for computation of the cost function (by 32 VPROD calls).

Best performance of the client:

1x SA-LMS (with Preferred SANE)

At this configuration the client will benefit from internally optimized self-adaptive LMS filter.

When no preferred SANE (the one with SA-LMS) is available the client constructs the self-adaptive LMS from LMS and VPROD elements as shown in Figure 2. Depending on the number of LMS and VPROD operations available in the computational cloud the client performs parallel calls in order to reach best performance. During its work the client displays its intermediate results through VGA display, see Figure 4.



```
Adaptive LMS FIR Filter Accelerator.
Self Adaptive Networked Element (SANE) Demo.
EU FP6 IST FET ACA Project AETHER.
http://sp.utia.cas.cz
(c) 2007,2008,2009 UTIA AV CR v.v.i.

MB with FP  : Generating far-end signal ...
MB with SANE: Computing room response ...
```

```
Mission     : Adaptive acoustic echo cancellation
Arithmetic  : Single precision floating point
Filter size : 500 coefficients
Sampling    : At least 44.1 KHz
```

Client LOG
Noise Cancellation Scheme
Simulated Room Parameters
Best Parameters Estimation
µ Dependent Parameters Estimations
Optimal µ
FPGA Design
Actual Performance
FP: MFLOPs
450
400
350
300
250
200
150
100
50
0
MB with SANE: Room model (FIR coefficients)
MB with SANE: Adaptive LMS FIR coefficients
MB SW FP: Adaptive LMS FIR coefficients
KHz
50
40
30
20
10
0
Adaptive LMS FIR filter
FPU CPU
MPMC
VGA
FPGA
ÚTIA
Acceleration of FP LMS
60x
40x
20x
0x
Learn Adaptation Speed
0.15
0.1
0.05
0

Figure 4: Self-adaptive LMS (Client) VGA Display

3. Used tools and resources

For running the demo following tools are required:

- CF card writer with mkdosfs utility
- At least two ML402 boards (three boards in ideal case), each board connected to one VGA display
- Two or three CF cards of size 64MB at least
- Ethernet switch
- PC running linux for running the root_sep application

4. Implementation

Setup configuration files for ML402 boards:

- Format each CF card with mkdosfs tool with its default settings
- Copy provided `petaX.ace` to the root of each CF card, `peta1.ace` to the firs one, `peta2.ace` to the second one etc. (in one SVP/SEP cloud cannot appear two identical images)
- Make sure that the SW12 is in SYSACE position.
- The jumper J26 must be in 1-2 position to proper JTAG chain setup for ML402 without daughter board.

Setup network:

- Set the connection from PC to switch. The setting of the IP protocol at PC side must be:
  - IP address: 192.168.0.1
  - Net mask: 255.255.255.0
- Connect boards to switch

Start the demo:

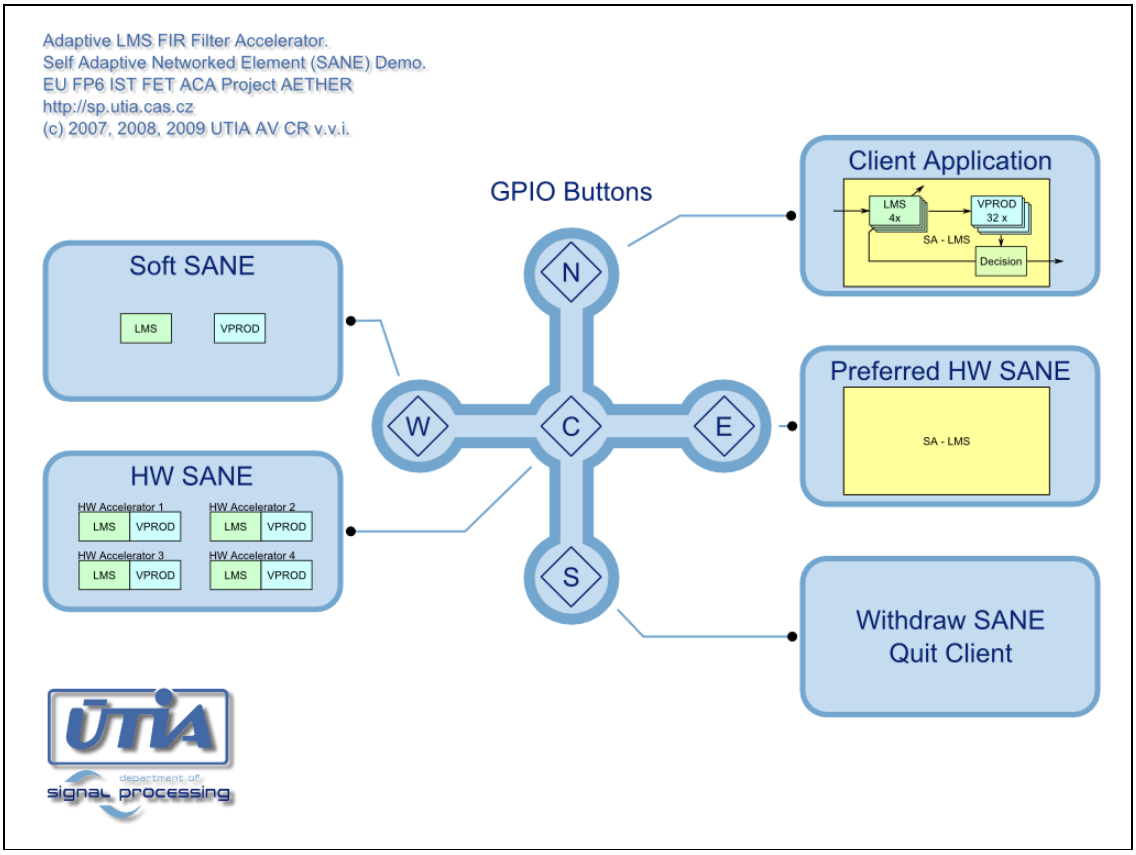
- Turn ML402 boards on and wait until the petalinux boots up and main menu appears on each VGA display (see Figure 5 for the main menu appearance)
- At PC side run the root_sep with following commadline parameters:

  ```
  root_sep --dutc_addr 192.168.0.1 --dutc_port 2000
  ```

- Choose the SW SANE by pressing GPIO_SW_W button at the first board (it should be seen that its resources has been registered at root_sep terminal)
- Choose the client at another ML402 board (GPIO_SW_N button)

Operating the demo:

- Pressing GPIO_SW_S once when the SANE is running initiates its withdraw from the cloud. IMPORTANT: The SANE withdrawal must be cooperational. User must wait until main menu appears after some time necessary to complete the operation. Green LED at each “SANE” button signals the intention to withdraw.
- The client doesn’t react to quit button (GPIO_SW_S) immediately. The user must hold it down until green LED becomes ON. Then the button can be released and main menu appears at the board.



Adaptive LMS FIR Filter Accelerator.
Self Adaptive Networked Element (SANE) Demo.
EU FP6 IST FET ACA Project AETHER
http://sp.utia.cas.cz
(c) 2007, 2008, 2009 UTIA AV CR v.v.i.

Figure 5: Main menu

5. Troubleshooting

To restart when the system freeze please follow this steps:

- Power off the boards
- Quit root_sep by pressing “q” or “ctrl-c”
- Power cycle the ethernet switch.
- Start root_sep again
- Turn on boards.

NOTE: After some time the license of the HW accelerator expires. It causes automatic withdrawal of the sane from the system. To recover the board must be power cycled.

6. Package contents

mkdosfs.exe	CF card formatting utility
peta1.ace	Bitstream for the first board – IP address 192.168.0.101
peta2.ace	Bitstream for the second board – IP address 192.168.0.102
peta3.ace	Bitstream for the last board – IP address 192.168.0.103
sep_root	Root server to be run on linux PC
salms_demo.pdf	This document

7. References

[1] J.Kadlec, *Design Flow for Reconfigurable Microblaze Accelerators*. ReCoSoC 2008 4th Reconfigurable Communication-centric Systems-on-Chip workshop, Eds: Moreno Manuel J., Madrenas Jordi, Sassatelli Gilles, Hübner Michael, Zipf Peter, Barcelona, ES, 09-11 Jul 2008