

3rd **PROCEEDINGS**
of the
Small
Systems
Simulation
Symposium
2010

February 12-14
Faculty of Electronic Engineering
Niš, Serbia

organized by the
Yugoslav Simulation Society
and the
Faculty of Electronic Engineering Niš



Acknowledgments

The organization of the Third Small System Simulation Symposium, SSSS 2010, was supported by

The Serbian Ministry of Science and Technological Development
The Town of Niš
The Community Mediana, Niš
Regional Center for Education, Niš
Rhode & Schwartz, Belgrade
Irvan International, Niš
Uno-Lux, Belgrade

Publisher:

Faculty of Electronic Engineering, Niš
P.O.Box 73, 18000 Niš
<http://www.elfak.ni.ac.rs>

Editor:

Vančo Litovski

CIP – Каталогизација у публикацији
Народна библиотека Србије, Београд

519.876.5(082)
004.942(082)

SMALL Systems Simulation Symposium (3 ; 2010 ; Niš)

3rd. Proceedings of the Small Systems
Simulation Symposium 2010, February 12-14,
Niš, Serbia / organized by Yugoslav
Simulation Society and Faculty of Electronic
Engineering ; [editor Vančo Litovski]. – Niš
: Faculty of Electronic Engineering, 2010
(Niš : Unigraf). – 93 str. : ilustr. ; 27 cm

Tekst štampan dvostubačno. – Tiraž 100. –
Bibliografija uz svaki rad. – Registar.

ISBN 978-86-6125-006-4

1. Yugoslav Simulation Society (Niš)
 2. Faculty of Electronic Engineering (Niš)
 - a) Симулација – Зборници
- COBISS.SR-ID 172762124

Printed by: “Unigraf”, Niš

STEERING COMMITTEE

A. Belić, Institute of Physics, Belgrade (Serbia)
S. Bojanić, Universidad Politecnica de Madrid (Spain)
M. Jevtić, University of Niš (Serbia)
M. Damnjanović, University of Niš (Serbia)
B. Dokić, Faculty of Electrical Engineering, University of Banja Luka (Bosnia and Herzegovina)
G. S. Đorđević, University of Niš (Serbia)
N. Janković, University of Niš (Serbia)
V. Katić, University of Novi Sad (Serbia)
V. Litovski, University of Niš (Serbia)
O. Nieto, Universidad Politecnica de Madrid (Spain)
D. Maksimović, YSS (Switzerland)
S. Mijalković, University of Delft (Netherlands)
S. Milenković, YSS (United Kingdom)
Ž. Mrčarica, YSS (Switzerland)
P. Petković, University of Niš (Serbia)
D. Trajanov, St. Cyril and Methodius University in Skopje (Macedonia)
V. Zerbe, Technical University of Ilmenau (Germany)
M. Zwolinski, University of Southampton (United Kingdom)

ORGANIZING COMMITTEE

M. Andrejević Stošović, University of Niš (Serbia)
S. Bojanić, Universidad Politecnica de Madrid (Spain)
M. Dimitrijević, University of Niš (Serbia)
S. Đorđević, University of Niš (Serbia)
B. Jovanović, University of Niš (Serbia)
V. Litovski, University of Niš (Serbia)
S. Milenković, YSS (United Kingdom)
J. Milojković, YSS (Serbia)
D. Milovanović, University of Niš (Serbia)
P. Petković, University of Niš (Serbia)
Z. Petković, University of Niš (Serbia)

SYMPOSIUM SECRETARY

Miona Andrejević Stošović
Faculty of Electronic Engineering
Aleksandra Medvedeva 14
18000 Niš
Serbia
Tel: +381 18 529 321
miona.andrejevic@elfak.ni.ac.rs

CONTENTS

| | |
|---|----|
| Desktop Supercomputing: Simulation With Multi-and Many-Core Processors, Invited lecture Zwolinski M. | 1 |
| Simulating Electrostatic Discharge Maksimović, D., Notermans, G. | 6 |
| BDD-based Cryptanalysis of LFSR Stream Ciphers Đorđević, S., Bojanić, S. and Nieto-Taldriz, O. | 13 |
| Modeling and Simulation of Digital Systems in different Domains Paunović, I., Zerbe V. | 17 |
| GinisED - Geographic Information System for Support of Evidencing, Maintenance, Management and Analysis of Electric Power Supply Network Stoimenov, L., Stanimirović, A., Bogdanović, M. , Davidović, N., Krstić, A. | 23 |
| An Agent-Based Simulation model of Stock Market Filiposka, S., and Trajanov, D. | 27 |
| Twenty years of ANN research and application in LEDA Litovski, V. | 33 |
| Multistep forecasting in electronics based on reduced information Milojković, J., and Litovski, V. | 42 |
| Low power digital design in Integrated Power Meter IC Jovanović, B., Zwolinski, M., Damnjanović, M. | 49 |
| Analysis of Real-Time Systems Timing Constrains Đošić, S., and Jevtić, M. | 56 |
| An approach to Digital Low-Pass IIR Filter Design Jovanović, B., and Jevtić, M. | 61 |
| Galvanotechnical Manufacture of Parts of Electrical Components using Pulse-reversed current Stević, Z., Rajčić-Vujasinović, M., and Topisirović, D. | 67 |
| Parallelizing Electronic Circuit Simulation on Multicore Computer Cluster Andelković, B., Dimitrijević, M., Andrejević Stošović, M., Litovski, V. | 71 |
| High Level Simulator of Spatial to Auditory Mapping System for Blind and Visually Impaired Petković, M., and Đorđević, G. | 75 |

| | |
|--|----|
| Computer Based Power Factor and Distortion Measuring for Small Loads Dimitrijević, M., and Litovski, V. | 81 |
| Strategies Against Side-Channel-Attack Stanojlović, M., and Petković, P. | 86 |
| Multi channel $\Delta\Sigma$ A/D converter for integrated power meter Mirković, D., and Petković, P. | 90 |

Desktop Supercomputing: Simulation With Multi- and Many-Core Processors

Invited Paper

Mark Zwolinski

Abstract—Processor speed has remained constant for several years, but the number of CPUs per chip has increased. Furthermore, graphics cards now include tens of processors. Using these resources for scientific computing is a new challenge. A number of standards have appeared that simplify much of the mechanics of writing parallel programs. The fundamental challenge of exactly how to exploit parallelism remains. This paper shows how two technologies, OpenMP and OpenCL, have been used to accelerate different aspects of circuit simulation.

Keywords—Circuit simulation, Parallel algorithms, SPICE.

I. INTRODUCTION

In 1965, Gordon Moore [1] observed that the number of transistors on an integrated circuit was doubling each year. Although subject to some revision, notably that *performance* doubles every 18 months, “Moore’s Law” has become a self-fulfilling prophecy. In recent years, there has been an important qualification to this law. The number of transistors continues to increase at the same rate as before, but clock speeds have stalled at less than 4 GHz. While clock speed should not be used as an absolute measure of performance, it is clear that the throughput of individual CPUs is increasing much more slowly than the transistor count.

The explanation for this discrepancy is, of course, that the number of CPU cores per integrated circuit is increasing. Ideally, therefore, the throughput per chip is continuing to increase in line with Moore’s Law. In practice, this speed increase can only be achieved if the applications are trivially parallel – in other words, if there is no communication between concurrent processes.

The number of cores per chip is currently 4 to 6 for Intel and AMD devices. A PC or server might contain four such ICs, allowing perhaps 16 cores to share memory. High Performance Computing (HPC) systems include many (tens of thousands) of such servers, which communicate through message passing protocols. On the other hand, graphics cards include tens or hundreds of small, dedicated processors (Graphics Processing Units or GPUs). Each processor

is capable of floating-point operations. Thus, graphics cards can be utilised for general purpose numerical programming (General Purpose GPUs or GPGPUs).

While transistor counts have been growing as anticipated by Moore’s Law, the productivity of designers has been advancing more slowly. There is currently a “design gap” between the output of designers and the productivity expected for fulfillment of the Moore’s Law prophecy. Simulators are important tools for bridging the design gap, but to date, most simulators for electronics design have been written with a single execution thread. This presents the designers of simulation tools with a new challenge. For many years, it has been possible to rely on increasing CPU speed to drive simulator performance. That is no longer possible and simulators must now be designed to exploit concurrency.

Three standards have emerged for the three types of parallelism introduced above. MPI (Message Passing Interface) [2] provides a mechanism for loosely-coupled processes to communicate. OpenMP [3] allows concurrent threads on parallel processors to communicate through shared memory. OpenCL [4] is a new standard that allows a programmer to exploit the power of GPUs. The three technologies can be mixed in the same software, allowing both homogeneous and heterogeneous systems to be built.

In this paper, the three programming technologies will be briefly described in the next section. An example of parallelising a simulator program using OpenMP will then be described. Finally, an example of the use of OpenCL will be given.

II. PARALLEL PROGRAMMING

Parallel programming has always been difficult and remains so. In this section, we will briefly look at three standards that assist with the mechanics of parallel programming. None of these approaches is a solution to the problem of *how* to convert a sequential algorithm into a parallel form.

It is possible to combine two or more of these standards in a single application, in order to make full use of the available resources.

Mark Zwolinski is with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK, Email:mz@ecs.soton.ac.uk

A. MPI

MPI [2] is the standard used in High-Performance Computing (HPC). As the name implies, MPI provides a standard method for passing messages between processes running on concurrent processors. Most of the functionality is provided by daemon processes running on each processor, thus, from a programmer's point of view, it is only necessary to include function calls, such as `MPI_Send` and `MPI_Recv`.

Because MPI relies on message passing that is slow and unpredictable with respect to time, it is only effective if the application is either sufficiently decoupled or sufficiently large that the overheads are not significant compared with the computation. (This is generally true of all parallel processing, but the speed of message passing is particularly significant.) Thus MPI is suited to simulations of large physical systems. In general, however, applications such as circuit simulation do not map easily to MPI.

B. OpenMP

OpenMP [3] is an applications programming interface for implementing shared memory, parallel programming. Within a program, a number of threads may be created, to run in parallel on separate cores. The shared memory model is particularly relevant to the multi-core processor systems that are now appearing as workstations.

A significant advantage of OpenMP over other coding styles is that it does not necessarily require major rewriting of existing code. The basic OpenMP model is that code is annotated with directives to show parallel sections.

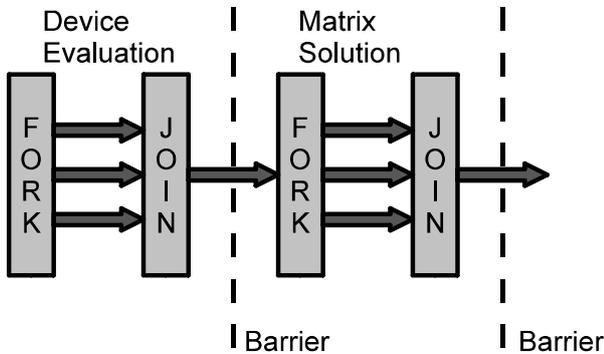


Fig. 1. Parallel thread execution

The execution model is that shown in Fig. 1. Threads are forked and joined according to the directives given by the programmer. For many applications this model of parallel sections interleaved with single threaded sections is appropriate. However, creating a new thread will take a certain amount of time. Therefore simply adding directives to existing code, without considering the overall program flow is unlikely to achieve a major speed-up.

The latest version of OpenMP, version 3.0, was published in May 2008. A significant enhancement is the ability

to label arbitrary loops and function calls with a `task` directive. For example, each element of a linked list of indeterminate length could be processed by a different thread using the `task` directive:

```
#pragma omp parallel
#pragma omp single
for (p=start; p; p=p->next)
#pragma omp task
    task(p);
```

The first directive, `#pragma omp parallel` is needed to set up the parallel environment. `#pragma omp single` specifies that the `for` loop incrementing is only done once, while the tasks are forked off to individual threads with the third directive. In this example, there is no need to qualify the parameter passed to each task as there is (apparently) no interaction between elements in the list.

The `task` directive can be applied to any statement, although the caveat about thread creation costs clearly applies. In particular, it can be applied to functions that process arbitrary data structures such as linked lists or trees.

It should also be noted that execution continues along the main thread at the same time as any forked threads. If the main thread completes a task before any forked threads, execution will proceed to the next statement. It may, therefore, be necessary to specify an explicit join point. In OpenMP 3.0, this is done with the `#pragma omp taskwait` directive.

C. OpenCL

In recent years, there has been a trend to move graphics processing onto specialised graphics cards. These contain 10 or more small-scale processors, each capable of floating-point operations. Attention has turned to the possible use of these Graphical Processing Units (GPUs) for numerical processing. The leading vendors have each produced their own development kits, but in 2009, a common programming interface – OpenCL [4] – was released. OpenCL is now an important part of Apple's Mac OS X.

The architecture of GPUs has been used in more powerful General Purpose GPUs (GPGPUs), that have a larger number of processors and which may not even include graphics outputs. Examples include the NVidia Tesla range [5] and the ATI/AMD Firestream cards [6].

Each processing unit on a graphics card has a limited amount of memory and limited processing power. The OpenCL language is a subset of C, designed to allow part of a problem to run on each processor. For example, a function to square the elements of a vector can be written as:

```
__kernel void square(
    __global float* input,
    __global float* output,
    const unsigned int count)
```

```
{
  int i = get_global_id(0);
  if(i < count)
    output[i] = input[i] * input[i];
}
```

Because OpenCL is intended to be portable between different GPGPUs, the *kernel* code is compiled “on the fly”. The programming interface, therefore, consists of routines to determine the hardware resources, to set up the computing environment, including input and output buffers, and to compile and run the kernel code.

The use of OpenCL is not limited to GPGPUs. The CPU of a system, perhaps with multiple cores, can be used as an OpenCL resource, running the same kernel code. As presently implemented, this is unlikely to be efficient, but could, in principle, obviate the need for OpenMP.

III. MULTI-THREADED CIRCUIT SIMULATION USING OPENMP

A. Hierarchical Circuit Simulation

In general terms, the equations for a nonlinear circuit may be expressed as a function [7]:

$$f(x, \dot{x}, t) = 0 \quad (1)$$

where x is the vector of unknown circuit variables, \dot{x} is the time derivative of x and t is time. This equation cannot be solved analytically and therefore it is discretized in time, such that a nonlinear set of equations is solved at each time point:

$$g(x^n) = 0 \quad (2)$$

where $x^n = x(t^n)$.

The nonlinear equation (2) is linearized using the Newton-Raphson (N-R) method:

$$A^m x^{m+1} = A^m x^m - g(x^m) = b^m \quad (3)$$

where A^m is the matrix of partial derivatives of g with respect to x at iteration m at time point t^n . x^{m+1} is the vector of unknown circuit variables. The iteration proceeds until convergence, $x^{m+1} \approx x^m$.

Calculating the entries of A^m and b^m can be done in parallel for each device in the circuit, because there is no interaction between the devices. Techniques exist for the parallel solution of matrices. The device evaluation phase must complete before matrix solution can start and the matrix solution must complete before the device evaluation in the next iteration can begin. So there are two barriers that limit the amount of parallel execution that may be performed, Fig. 1.

A different approach is to partition the circuit and to solve each partition in parallel. The idea of maintaining the hierarchical partitioning of a circuit for simulation was first proposed in the mid-1970s [8]. The basic idea is that of node-tearing.

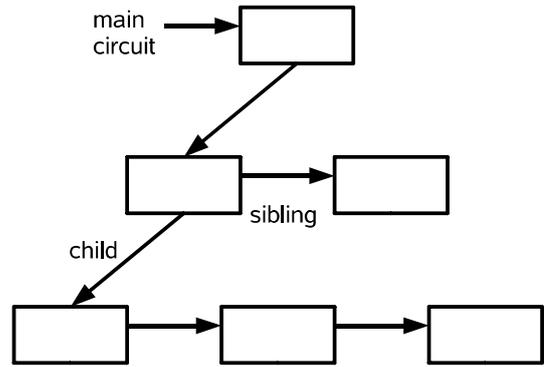


Fig. 2. Sub-circuit hierarchy

The sub-circuit hierarchy can be represented as binary tree, Fig. 2. Solving the circuit equations at one N-R iteration at one time point requires two traversals of this tree. This can be done using recursive procedures as illustrated in the following two algorithms.

Algorithm 1 ForwardElim(subcct *ptr)

```
1: if ptr->child then
2:   ForwardElim(ptr->child)
3: end if
4: EvaluateDevices(ptr)
5: GaussFore(ptr)
6: if ptr->sibling then
7:   ForwardElim(ptr->sibling)
8: end if
```

Algorithm 2 BackSubst(subcct *ptr)

```
1: GaussBack(ptr)
2: if ptr->child then
3:   BackSubst(ptr->child)
4: end if
5: if ptr->sibling then
6:   BackSubst(ptr->sibling)
7: end if
```

Algorithm 3 Simulation(subcct *maincircuit)

```
1: while t < tMAX do
2:   repeat
3:     ForwardElim(maincircuit)
4:     BackSubst(maincircuit)
5:   until convergence
6:   UpdateTimestep
7: end while
```

The two algorithms are called, in turn, for the main, top-level circuit until convergence is reached at each time point, Algorithm 3. EvaluateDevices calculates the contribution of each device to the sub-circuit matrix equation. GaussFore

performs the forward phase of the Gaussian Elimination for each sub-circuit and GaussBack does the back substitution. It can be seen, therefore, that the overwhelming majority of the computation effort is expended in Algorithm 1.

This hierarchical solution approach has been implemented in a circuit simulator. If all subcircuits use a common timestep, the results obtained from a hierarchically partitioned simulation are mathematically the same as for a non-partitioned circuit. There may, however, be numerical differences because of a different evaluation order. It should also be noted that in order to perform the internal node suppression, Gaussian Elimination is used, in contrast to LU factorization, as in SPICE.

B. Simulator Acceleration

The application of OpenMP to the hierarchical circuit simulator is motivated by a simple observation: the processing for one sub-circuit can be done at the same time as that for any of its siblings. Therefore, in principle, a new thread can be created for each sibling at each level of the hierarchy. It is, however, true that a child must be processed before its parent during the Forward Elimination phase (Algorithm 1). Therefore, there is no useful purpose in creating a new thread for the first child of any parent.

There is a cost to creating a new thread. The application of OpenMP has therefore been restricted to the Forward Elimination phase. This allows parallelization of both the model evaluation and matrix factorization.

Algorithm 1 is therefore rewritten as Algorithm 4.

Algorithm 4 ForwardElim(subcct *ptr)

```

1: if ptr->sibling then
2:   #pragma omp task
3:   ForwardElim(ptr->sibling)
4: end if
5: if ptr->child then
6:   ForwardElim(ptr->child)
7: end if
8: EvaluateDevices(ptr)
9: GaussFore(ptr)
10: #pragma omp taskwait

```

As can be seen, the changes are minimal. The call to process any sibling is made at the start of the routine. This does not affect the functionality in any way. A breadth-first, rather than a depth-first traversal is made, but children are always processed before their parent. The change is made to allow a new thread to be created at the start of the algorithm, so that it will execute concurrently with the remainder of the routine.

The OpenMP directive `#pragma omp task` is used to indicate that the call to ForwardElim for the sibling should be executed as a separate thread. Because this call will be executed for all the siblings at one level, all siblings would

TABLE I
RUN TIMES FOR PCHIP

| Threads | Run Time (s) |
|---------|--------------|
| 1 | 68.0 |
| 2 | 60.9 |
| 3 | 54.0 |
| 4 | 46.8 |
| 5 | 38.9 |
| 6 | 30.7 |
| 7 | 23.0 |
| 8 | 15.2 |

therefore be processed concurrently in separate threads. It is possible to attach attributes to the OpenMP directives to indicate the data scope and hence to protect data against corruption by other threads. In this case, because of the design of the data structures and because of the way in which models are evaluated and matrix values are updated, there is no interaction between siblings and hence there is no need to add extra attributes. Data from siblings is collected by their parent and hence any interaction between siblings occurs after they have all completed their execution.

A second OpenMP directive is needed at the end of the routine to ensure synchronization. `#pragma omp taskwait` causes the calling routine to wait until any threads that it has created have completed. Omitting this directive could allow processing to start on the parent before the children have completed and hence lead to incorrect or corrupted data.

In addition to these two directives, the two OpenMP directives `#pragma omp parallel` and `#pragma omp serial` need to be included in the main calling routine to set up parallel regions and to ensure that the timing and N-R loop control statements are only executed once, respectively.

C. Results

This example is taken from the CircuitSim90 [9] collection of benchmark circuits. The pchip circuit has 1029 transistors. The input and output buffers were not considered. Eight instances of the circuit were used, but this time they were chained together, to avoid any suggestion of trivial parallelism. The number of threads can be set by the environment variable `OMP_NUM_THREADS`. By default, this is equal to the number of cores, in this case 8. The run times for the operating point analysis are given in Table I and plotted in Fig. 3.

The trend in Fig. 3 clearly shows that the run time decreases monotonically with the number of available threads. In this case, the complexity of the computation far outweighs the cost of thread creation. There is no load balancing, so, in effect, this shows the time required to process 8 sub-circuits down to one sub-circuit per thread. The speed-up is 4.47 times for 8 threads.

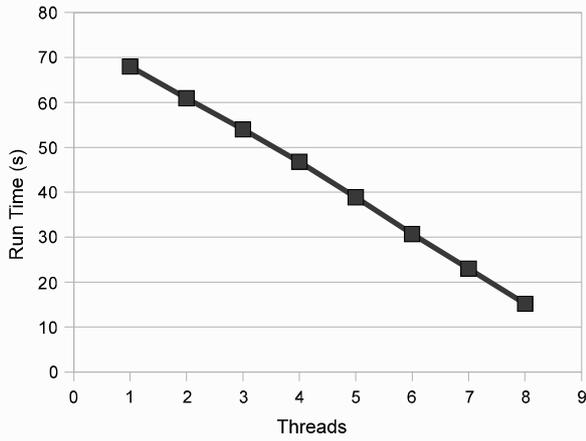


Fig. 3. Run time vs. No. Threads

IV. MATRIX FACTORISATION USING OPENCL

In circuit simulation, at each N-R iteration, equation (3) is solved by factorizing A^m into lower and upper triangular matrices, L and U , and forward and back substituting to give x^{m+1} . J is usually very sparse (because in general, electronic components are connected to only 2 or 3 other components) and therefore the solution time is typically $O(N^{1.5})$ or better, where N is the number of circuit nodes. On the other hand, in circuit simulation, the matrix is asymmetric (because circuits have gain), so methods such as Cholesky decomposition are not appropriate.

Crout's algorithm [10] is used to factorise a matrix. Implicitly $l_{ii} = 1, i = 1, \dots, N$,

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}, i = 0, \dots, j \quad (4)$$

and

$$l_{ij} = \frac{1}{u_{ii}} \left[a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj} \right], i = j + 1, \dots, N - 1. \quad (5)$$

It can be seen that there is dependency between the two computations. Thus matrix factorisation is usually performed in a serial manner.

In order to parallelise the process, it is necessary to divide the matrix into sub-matrices [11]. These sub-matrices are then coupled together in a final step. Each of the sub-matrices can be factorised in parallel. This is exactly equivalent to thinking of the circuit being partitioned into sub-circuits, as in Figure 2.

Figure 4 shows the speed increase that can be achieved for large matrices. The experiment was performed on an NVidia Tesla card. The test data was a diagonally-banded matrix. In fact, the example was coded using the precursor to OpenCL – CUDA. It can be seen that the GPU version of the code is about 13 times faster for matrices of dimension 8000. The cross-over occurs at about 500; below that the

overhead required to move data on and off the GPU tends to dominate.

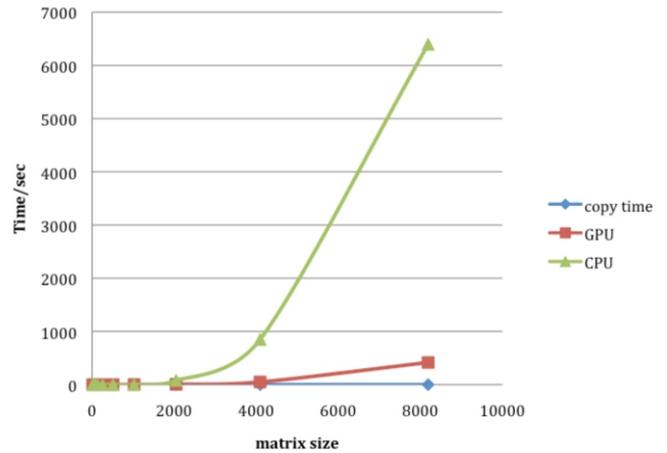


Fig. 4. GPU vs. CPU Run Time

V. CONCLUSIONS

Parallel programming remains one of the most significant challenges in the development of new EDA tools. While new technologies allow multiple CPUs and GPUs to be exploited, they do not solve the problem of how to partition a problem. Nevertheless, by using these technologies, either singly or together, we now have the opportunity to simulate much larger systems on a desktop machine than would be possible using a single CPU.

ACKNOWLEDGEMENTS

The results for matrix factorisation were obtained by Wang Yuyang as part of his MSc dissertation project.

REFERENCES

- [1] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, 1965.
- [2] <http://www.open-mpi.org/>.
- [3] <http://www.openmp.org>.
- [4] <http://www.khronos.org/opencl/>.
- [5] http://www.nvidia.com/object/tesla_computing_solutions.html.
- [6] <http://www.amd.com/us/products/technologies/stream-technology/Pages/stream-technology.aspx>.
- [7] V. Litovski and M. Zwolinski, *VLSI Circuit Simulation and Optimization*. Chapman and Hall, 1997.
- [8] N. Rabbat and H. Hsieh, "A latent macromodular approach to large-scale sparse networks," *Circuits and Systems, IEEE Transactions on*, vol. 23, no. 12, pp. 745–752, Dec 1976.
- [9] <http://www.cbl.ncsu.edu:16080/benchmarks/>.
- [10] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge, UK: Cambridge University Press, 1992.
- [11] C.-C. Chen and Y.-H. Hu, "Parallel LU factorization for circuit simulation on an MIMD computer," *Proceedings of the 1988 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pp. 129–132, Oct 1988.

Simulating Electrostatic Discharge

Dejan Maksimović, Guido Notermans

Abstract - In this paper we justify the necessity of the electrostatic discharge simulation. We give an overview of the ESD stress standards and the ESD protection devices. We further describe the modelling of the ESD devices and give a case study which shows the importance of timely ESD simulation for the design success.

Keywords – ESD, simulation, modelling, HBM, MM, CDM, HMM.

I. INTRODUCTION

Electrostatic discharge (ESD) occurs between two bodies at different electrostatic potentials. The charging of these bodies can occur either by triboelectricity or by induction. ESD is characterized by a short duration (0.1ns to 100ns), high current (1A to 30A) pulse. Such high current can damage the semiconductor device. Failures can be thermally induced due to the high power dissipated during the ESD event: silicon melting can be observed as well as metal or polysilicon resistance blow-up if the metal/poly line is not designed wide enough. Gate oxide breakdown can also be observed due to the large voltage drop built-up by the ESD current.

ESD can occur any time in the life of the product: during manufacturing, assembly, testing, shipment and in the final application and it is a key issue for the reliability of the integrated circuits (ICs).

Two approaches are used together to fight against the ESD. The first one is to prevent the ESD events. Special dissipative materials are used in clean rooms and labs, ionizers, proper grounding of the equipment, wearing of a wrist strap during the tests etc. The second approach is to implement efficient ESD protection on the IC. The ideal ESD protection circuit is similar to a switch: it is highly resistive during the normal operation of the IC, but it is able to detect an ESD event and to become low resistive when it occurs. In such a way the ESD device shunts the ESD current with the lowest possible voltage drop.

Dejan Maksimović and Guido Notermans are with the ST-Ericsson, Binzstrasse 44, 8045 Zurich, Switzerland, E-mail: dejan.maksimovic@stericsson.com, guido.notermans@stericsson.com.

II. MODELING THE ESD EVENTS

There are many ESD models, three of them being the most widely used: human body model (HBM), machine model (MM) and charged device model (CDM).

2A. Human Body Model (HBM)

This model corresponds to the discharge of a charged human being into the IC. The capacitance of the average human body (to ground) is 100pF. The average skin resistance is 1.5kOhm. The body capacitance is charged to a certain voltage level and discharged through the skin resistance and the device under test (DUT) to the ground. The electronic circuit representing this event is shown in Figure. 2.1.

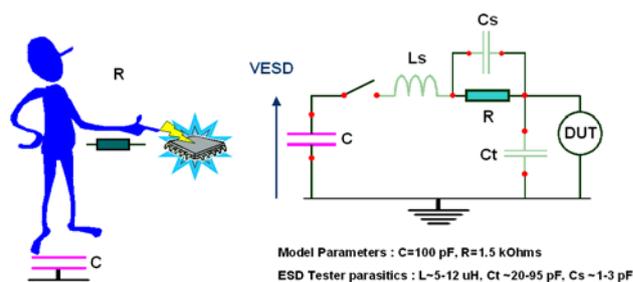


Fig. 2.1. Human Body Model (HBM).

This ESD model is defined by the JEDEC standard JESD22-A114F [1].

2B. Machine Model (MM)

This model emulates the discharge that can occur in automatic assembly lines between a machine and the IC. The charged machine has a higher capacitance of 200pF, while the contact resistance is very low, almost zero, often considered as a few ohms. Due to the low resistance, this model is strongly dependent on the parasitic inductance which has to be fixed and is in the order of 0.5uH. The electronic circuit representing this kind of ESD event is shown in Figure. 2.2.

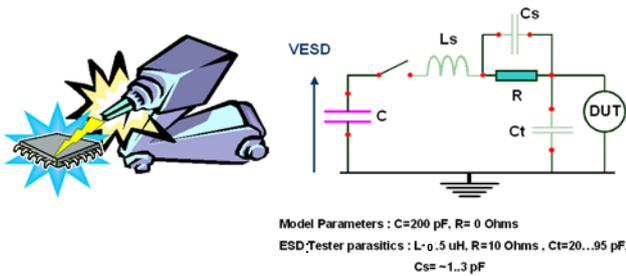


Fig. 2.2. Machine Model (MM).

This ESD model is defined by the IEA/JEDEC standard IEA/JESD22-A115-A [2].

2C. Charged Device Model (CDM)

This model emulates the discharge of a charged IC to the ground which occurs when one pin of the IC touches a grounded surface. The whole IC is charged and the discharge is determined by many device parameters such as the package type and the die size. The CDM event is very short (rise time is less than 0.5ns) high current pulse in order of tens of amperes. It mostly causes the gate oxide failures due to the overvoltage caused by such a high current. A typical CDM current waveform is shown in Figure 2.3.

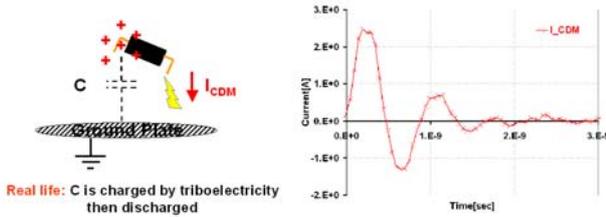


Fig. 2.3. Charged Device Model (CDM).

During the CDM test the device is placed in a "dead bug" position on a charging plane connected to a high voltage source. Above the device there is a ground plane. The discharge occurs when the pogo pin connected to the ground plane touches one pin of the IC. This ESD model is defined by the JEDEC standard JESD22-C101C [3].

While the HBM and MM events occur between two pins of the IC and the circuit can be designed in such a way that the ESD current path is predictable, during the CDM stress the current comes from the silicon substrate and distributes in an unpredictable way through the metal lines and devices towards the stressed pin.

2D. System Level ESD Stress (Gun Test)

This model corresponds to the "real-world" discharge that happens when the final user handles the product that contains the IC. The stress levels are much higher and the test is performed using the ESD gun. The discharge is

applied to every possible exposed surface of the product, such as metal connectors, displays, case, etc. The ESD current flows from the stressed point to the system ground (and another way around), which is similar to the CDM discharge. The current waveform is shown in Fig. 2.4. It consists of a very short CDM-like first pulse of very high amplitude and a HBM-like second pulse of the amplitude higher than that of the HBM pulse for the same stress level. The gun pulse parameters for different stress levels are given in Table 2.1.

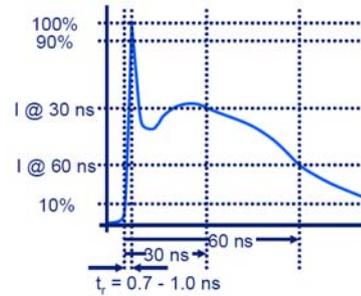


Fig. 2.4. Gun test current waveform.

TABLE 2.1
 GUN TEST CURRENT WAVEFORM PARAMETERS FOR DIFFERENT STRESS LEVELS

| Level | Indicated voltage [kV] | First peak current of discharge ±10% [A] | Rise time tr contact discharge [ns] | Current (±30%) at 30 ns [A] | Current (±30%) at 60 ns [A] |
|-------|------------------------|--|-------------------------------------|-----------------------------|-----------------------------|
| 1 | 2 | 7.5 | 0.7-1.0 | 4 | 2 |
| 2 | 4 | 15 | 0.7-1.0 | 8 | 4 |
| 3 | 6 | 22.5 | 0.7-1.0 | 12 | 6 |
| 4 | 8 | 30 | 0.7-1.0 | 16 | 8 |

TABLE 2.2
 HBM PEAK CURRENT VERSUS THE FIRST PEAK AMPLITUDE IN THE GUN TEST

| Applied voltage [kV] | HBM peak current [A] | Gun test first peak current [A] |
|----------------------|----------------------|---------------------------------|
| 2 | 1.33 | 7.5 |
| 4 | 2.67 | 15.0 |
| 6 | 4.00 | 22.5 |
| 8 | 5.33 | 30.0 |
| 10 | 6.67 | 37.5 |

This ESD model is defined by the IEC standard 61000-4-2 [4]. Table 2.2 compares the maximum peak current during the HBM and gun test for the same voltage levels [5].

2E. Transmission Line Pulse (TLP) Measurement

The ESD tests described so far are pass/fail measurements. They do not give any information on the behaviour of the IC during the ESD event. To obtain I(V) characteristics of the ESD protection circuits and devices a special tool called TLP is used [6].

During the TLP measurement, the DUT is subjected to a trapezoidal positive current pulse. Once the transients in

the device are over, the current through and the voltage over the DUT are measured. This produces one I/V data point. To check if the device is still not damaged, the DC leakage through the DUT is measured after each I/V data point extraction. If no degradation is observed, the amplitude of the current pulse is increased and the next data point is measured. In this way the I(V) curve of the device can be constructed starting from the low ESD currents and finishing after the device is damaged.

The TLP can vary the rise time and/or the width of the current pulse. The most commonly used parameter values are $T_r=10\text{ns}$, $T_w=100\text{ns}$. The I/V data point is usually taken at 90% of the T_w , i.e. after 90ns.

The current and voltage waveforms during the TLP measurement look like those shown in Fig. 2.5. The waveforms in Fig. 2.5 were obtained by the electrical simulation though.

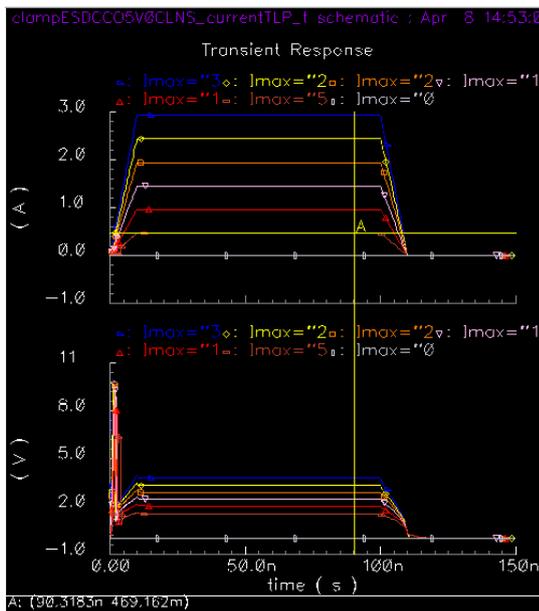


Fig. 2.5. The TLP waveforms obtained by the simulation. The measurement point at 90% of pulse width is shown.

A typical measured TLP curve of an ESD device is shown in Fig. 2.6.

2F. Definition of ESD Parameters

The most important ESD parameters are shown on the TLP curve in Fig. 2.6. They are as follows:

Vt1: Trigger voltage. This voltage must not be higher than the (gate oxide or PN junction) breakdown voltage of the circuitry connected in parallel to the ESD device. In case the device does not exhibit the snap-back (like in the case of the diode), this parameter is sometimes called V_{on} .

Vh: Hold voltage. Minimum voltage on the ESD device is important since if it is below the supply voltage of the

IC, the ESD event can cause the latch-up (large leakage from the supply to the ground that can be stopped only if the supply is turned off).

It2 – Failure current. This is the maximum ESD current that the ESD device can conduct without being damaged.

Vt2 – Voltage at failure level. This parameter is important only if $V_{t2} > V_{t1}$.

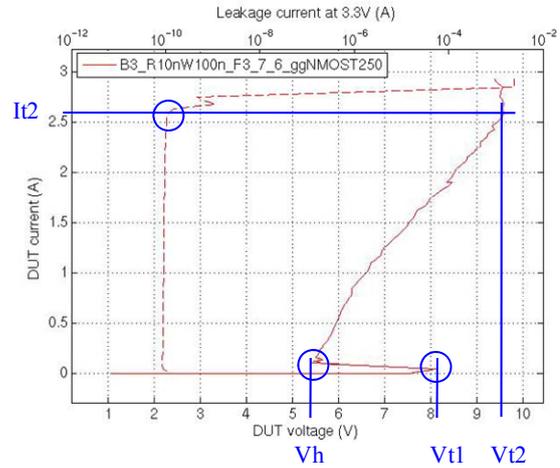


Fig. 2.6. Typical TLP characteristic of a ggNMOST with the ESD parameters marked.

2G. Correlation between different ESD stresses and TLP

A 100ns current pulse delivers a thermal stress equivalent to the HBM stress. A good correlation has been reported between the TLP It_2 value and the HBM fail level: the device which fails at 2kV HBM level will also fail at 1.33A TLP current if 100ns wide pulse is applied. Hence, there is a correlation factor of approximately $2\text{kV}/1.33\text{A} = 1.5\text{kV}_{\text{HBM}}/\text{A}_{\text{TLP}}$ between the It_2 in TLP curve and the HBM fail level.

The HBM and MM produce similar IC failures and there is a relation between them: typically 2kV HBM level corresponds to 100V MM level (ratio 20 times). Consequently, the correlation factor between the MM fail level and the It_2 is approximately $75\text{V}_{\text{MM}}/\text{A}_{\text{TLP}}$.

The relation between IEC and TLP stress is: 1A TLP corresponds to 600V IEC gun test [7]. This results in 13.3A TLP current for 8kV IEC.

III. ESD PROTECTION DEVICES

Early in the design process, the ESD protection concept is being discussed and planned. There are two main approaches: local ESD protection and rail-based ESD protection. If the ESD protection is implemented locally, then each pin of the IC has its own ESD protection element

(also known as "local ESD clamp"). In rail-based concept, the protection elements are connected only between the power and ground rails (rail clamps). The ESD current is diverted from each input/output (I/O) pin using the ESD diodes connected between the I/O pin and the power rail and between the I/O pin and the ground rail.

There are many different types of ESD protection devices. Some of them will be shortly described here.

3A. Diodes

Diodes are used in forward bias to conduct large ESD currents. The forward biased diode can conduct between 5mA and 30mA per micrometer of the junction perimeter (typically 10ma/um). In the inverse (Zener) breakdown the diode can however conduct very small ammount of ESD current.

The problem that often arises when one wants to simulate the ESD is that the diodes are not properly modelled in forward bias. This because the diodes are very rarely used in functional part of the IC – they are only the parasitic junctions and are supposed to always remain inverse biased. ESD diodes are however an exception to this rule.

3B. Snap-back devices

The most popular snap-back devices are ggNMOST and thyristor (also known as silicon-controlled rectifier – SCR). They have the I(V) characteristic similar to that in Fig. 2.6. Due to the negative resistance region in the curve, they cannot be simulated. SCRs have much deeper snap-back (lower Vh) than the ggNMOSTs.

It is important to say that each NMOST (or PMOST) can be driven into the bipolar mode and experience the snap-back. For this to happen, the voltage between their drain and source must reach the inverse breakdown of the drain/bulk junction. Of course, this is an unwanted event and it is sometimes possible to simulate the surrounding circuitry to discover if the critical voltage is reached on the MOS transistor.

3C. BigFETs

If a large NMOS transistor is conducting in parallel to the protected circuitry during the ESD pulse, it can take over all the ESD current and such protect the rest of the circuit. This ESD device is known as bigFET (or RC triggered FET) and is used as a rail clamp in most of the modern CMOS technologies. BigFET triggers at around 1V (slightly higher than a diode) and has a special RC circuit which is responsible to switch it off after the ESD pulse is over. Since all the circuitry in this clamp is working in

normal MOS operation mode, the device can be simulated.

IV. ESD SIMULATION

To be able to simulate the ESD event two models are necessary:

- the model of the ESD current pulse
- the model of all the ESD (and other) components in the circuitry connected to the stressed IC pin.

Both models can be easily developed for the analog simulator. The stimulus generator is made using a circuit such as that in Fig. 2.1 and 2.2. The waveform is adjusted to that prescribed by the corresponding ESD test standard.

The ESD devices are usually produced on a test wafer and TLP measured to extract their electrical parameters. Then the electrical model is built using these parameters and used in ESD simulation.

A few examples of ESD simulations on the device level will be given in following paragraphs.

4A. Generating the TLP characteristic of the bigFET clamp

The TLP tester is used for on-wafer measurements. The test circuit often contains additional resistance of metal connections from the ESD device to the measurement pads. Therefore the measured TLP curve has higher resistance than the ESD device itself.

If the ESD device is of the kind that can be simulated, such as a bigFET clamp, the simulation can show us the real resistance. The Fig. 4.1 shows a simple simulation testbench with such a bigFET rail clamp.

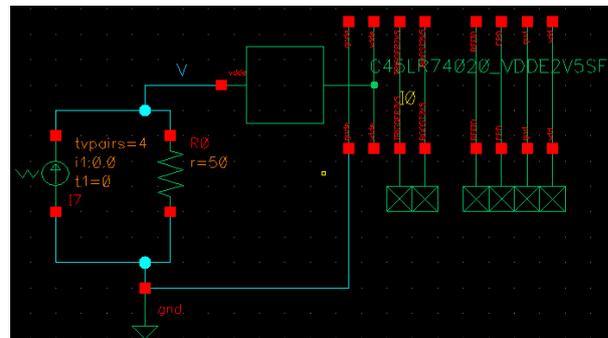


Fig. 4.1. Testbench for generating the TLP characteristic of the clamp by the simulation.

The trapezoidal current generator and the 50ohm resistor model the TLP system. The current and voltage waveforms obtained by the simulation are shown in Fig. 2.5. The parametric analysis is used to change the amplitude of the current pulse with constant step. Same as in TLP system, the I/V points are measured at 90ns simulation time.

The resulting TLP curves of a few different types of clamps are shown in Fig. 4.2. It must be stressed that the transistor models are valid only in the nominal supply range (up to 5V in this case). The simulations we did here by far exceed this range. This means the curves can be trusted only up to 5V. Nevertheless, the TLP measurements and long experience in this technology process show us that the clamp does not change the resistance until very close to the destruction level. The failure level can only be determined by the TLP measurement on the silicon. The simulation cannot give us this information. However, even the TLP can measure only up to 10A and above this point we cannot judge the behaviour of the devices which are designed for higher currents.

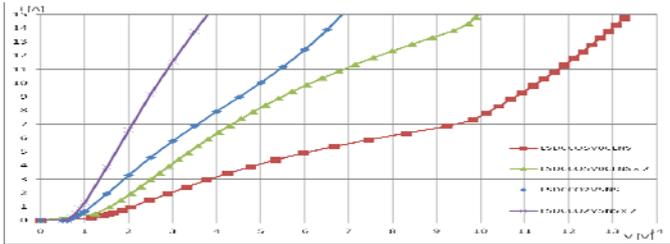


Fig. 4.2. Simulation waveforms.

To eliminate the simulation artefacts (invalid simulation results due to too high voltages), we can interpolate the clamp curves obtained by the simulation in the valid range as shown in Fig. 4.3. This enables us to estimate the voltage drop on the clamp in case of different ESD stresses and help us decide if we need to use more than one clamp in parallel to reduce the voltage.

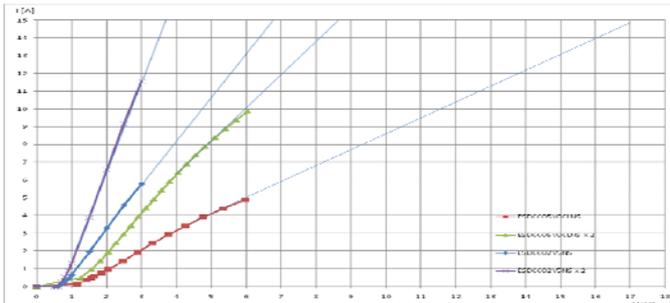


Fig. 4.3. Linear interpolation of the TLP curves generated by the simulation outside of the valid range of the circuit models.

B. Simulating a bigFET clamp under the CDM stress

The testbench schematic for the simulation of the CDM stress influence on the clamp is shown in Fig. 4.4. The sinewave current generator emulates the CDM pulse. The CDM pulse is approximately 2ns long positive pulse of certain amplitude. The negative half-period of the used sine generator is not important – the reverse diode of the clamp will conduct during it. The results of the simulation are shown in Fig. 4.5.

First of all, during the negative half-period of the

current pulse, the simulation shows only 1V over the reverse diode of the clamp. This is definitely an underestimation – the diode model is obviously not valid in forward bias.

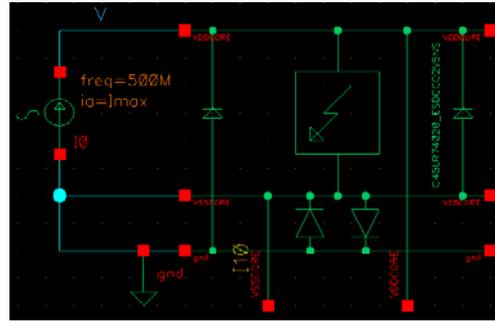


Fig. 4.4. A simple simulation testbench for the CDM stress.

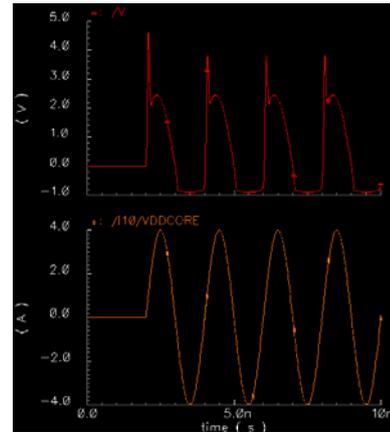


Fig. 4.5. CDM simulation waveforms.

During the positive half-period of the current pulse the voltage on the clamp shows a short, very high peak. This peak is the consequence of the bad design of the clamp. The clamp cannot trigger fast enough to conduct the CDM pulse. Some time later, the clamp triggers and this pulse falls down (clamp resistance is reduced) to 2V. This voltage is realistic to expect for this kind of the clamp when conducting a 4A current.

From this simulation we can conclude that the clamp triggering is not fast enough if 4A CDM peak current is to be expected from the IC.

V. A CASE STUDY

A simplified schematic of the output stage of the radio antenna in an IC is shown in Fig. 5.1. The power amplifier has a PMOST connected between the VDD_PA and the antenna pin Int_Ant. The bulk diode of the PMOST (Dpmost) is also shown in the schematic. The wire resistance between the PMOST drain and the Int_Ant pad is around 2ohm. The voltage at the antenna pin can vary between -2V and +2V. Therefore, the four-diode string (D11-D14 and D21-D24) is used as the ESD protection on this pin. The rest of the circuit is protected with a standard

rail-based ESD concept, which assumes the rail clamps (bigFETs) between the power (VDD_PA) and ground (VSS_PA) rail. In the shown voltage domain two such clamps are connected (C1 and C2). Each rail clamp has the reverse diode (Drc1 and Drc2).

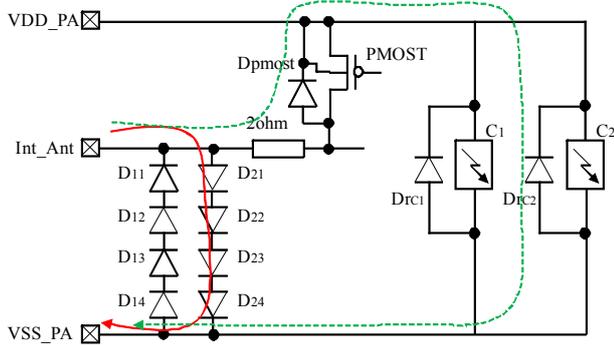


Fig. 5.1. ESD protection and functional circuitry at the antenna pin of an FM radio IC.

The Int_Ant pin is supposed to withstand 8kV system level stress (gun test) without adding any additional external ESD protection on the application board. The intended ESD current path for the current between Int_Ant and VSS_PA is through the 4-diode string (depicted with red line in Fig. 5.1). There is however the alternative path through Dpmost and clamps C1/C2, as shown by the dashed green line in Fig. 5.1. This unwanted current could damage some of these devices. To prevent this, the circuit was modeled and simulated before the production.

The 4-diode strings are designed to be able to conduct 14A of the 100ns TLP current. Their TLP characteristics measured on a separate test wafer are shown in Fig. 5.2. The TLP tester could produce maximum current of 10A, hence the diodes could not be damaged. The diode curve was estimated up to 14A by removing the connection resistance (depicted red in Fig. 5.2). The simulation model was generated with $V_{on}=4V$ and $R_{on}=0.25\Omega$, same in both directions.

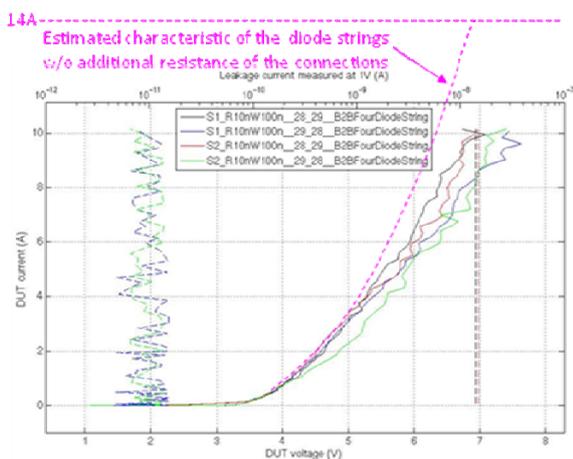


Fig. 5.2. Measured TLP characteristics of the 4-diode string, two samples measured in both directions on a test silicon.

In a similar way the rail clamps C1 and C2 were measured and their electrical model was created. In the forward direction the clamps have $V_{on}=0.35V$ and $R_{on}=1\Omega$. In the reverse direction (Drc1 and Drc2) the equivalent circuit contains $V_{on}=0.6V$ and $R_{on}=1\Omega$. The bulk diode of the PMOST was also modeled by $V_{on}=1\Omega$ and $R_{on}=1\Omega$. The resulting simulation model of the circuit is shown in Fig. 5.3.

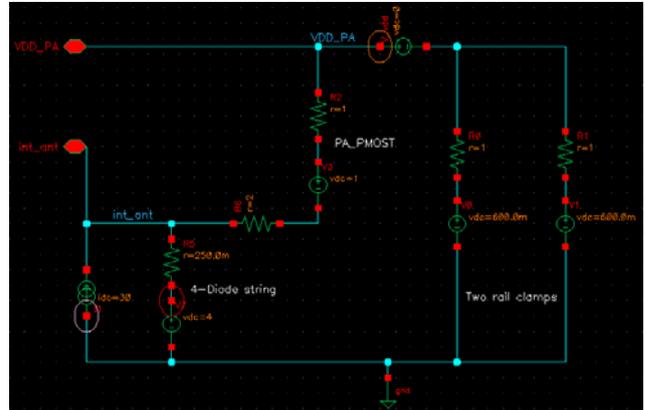


Fig. 5.3. Simulation model of the system for the positive IEC pulse at Int_Ant pin.

The 8kV IEC stress consists of the first short 30A current peak which can generate the overvoltages in the circuit and the second 16A current peak which can damage the components by inducing thermal damage. One simple approach to the simulation is to use a DC current source of a maximum current value (30A and 16A) and check the DC currents that flow through all the circuit elements.

The result of such a simulation is shown in Fig. 5.4. The second 16A current pulse was brought to the circuit. The simulation shows that there are no large overvoltages in the circuit nodes. The maximum node voltage is $V(\text{Int_Ant})=7.44V$, which is lower than the breakdown voltage (8.2V) of the gate oxide connected to this node.

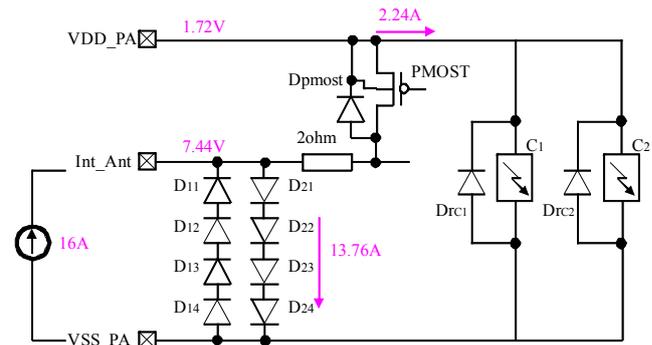


Fig. 5.4. Simulation model of the system for the positive IEC pulse at Int_Ant pin.

The maximum of 13.76A flows through the diode string D21-D41. This is safe since the diodes were designed for 14A. The simulation also shows that 2.24A flows through the unwanted path – through the Dpmost and two clamps C1/C2. Based on the size of the PMOST (720um) it can be

estimated that D_{pmost} can stand as much as $720\mu\text{m} \cdot 10\text{mA}/\mu\text{m} = 7.2\text{A}$. Hence, D_{pmost} will not be damaged. Two rail clamps C1/C2 are designed to be able to survive as much as 4A of 100ns TLP current (to stand 200V MM stress). Obviously, they will also be able to conduct 2.24A current.

The 2ohm resistor represents the estimated resistance of the relatively long and thin metal connection between the Int_Ant pad and the PMOST. Unfortunately, this connection was not carefully reviewed during the ESD review of the circuit before the production. As a result, the gun test on the produced IC showed circuit failure at 6.5kV stress level instead of targeted 8kV. The failure analysis (FA) of the failing samples showed that the connection towards the PMOST was lost. The weakest point on the metal connection were three 1.9um wide lines in M3 which melted, as visible in Fig. 5.5.

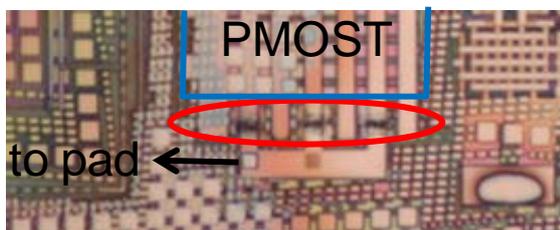


Fig. 5.5. The failure analysis result, three melted M3 lines at PMOST's drain.

A M3 line in the used 45nm process can conduct maximum 225mA of the 100ns TLP current per micrometer of the line width. That means that these three parallel M3 wires could conduct maximum $3 \cdot 1.9\mu\text{m} \cdot 225\text{mA}/\mu\text{m} = 1.28\text{A}$, which confirms the root cause of the fail.

VI. CONCLUSION

In this paper we defined the pre-requisites for ESD simulation. The modelling of the ESD devices is explained and a case study is given that proves the usefulness of the ESD simulations. In this case, we show that quite simple DC simulation of the circuit is able to explain the root cause of the gun test fail.

One should be careful when simulating ESD since the currents and/or voltages often exceed the valid range of the component models. Therefore it is always recommended to calibrate the simulation results using the measured TLP characteristics of the silicon test structures.

The bigFETs (which are the most popular ESD devices in technologies below 90nm) can be accurately simulated. The simulation of the pad-ring is obligatory when defining the ESD rules in the rail-based ESD concept.

Only the snap-back ESD devices cannot be simulated due to the negative resistance region in their I/V curve. They can be modelled by the V_{t1} and R_{on} , but the designer must be aware that the simulation result becomes invalid as soon as the voltage on the snap-back device exceeds V_{t1} .

REFERENCES

- [1] *Electrostatic Discharge (ESD) Sensitivity Testing Human Body Model (HBM)*, JEDEC standard JESD22-A114F, December 2008.
- [2] *Electrostatic Discharge (ESD) Sensitivity Testing Machine Model (MM)*, IEA/JEDEC standard IEA/JESD22-A115-A, October 1997.
- [3] *Field-Induced Charged-Device Model: Test Method for Electrostatic-Discharge-Withstand Thresholds of Microelectronic Components*, JEDEC standard JESD22-C101C, December 2004.
- [4] *Electromagnetic compatibility (EMC) – Part 4-2: Testing and measurement techniques – Electrostatic discharge immunity test*, IEC standard 61000-4-2, edition 2.0, December 2008.
- [5] *Human Body Model (HBM) vs. IEC IEC61000-4-2*, California micro Devices, white paper, January 2008.
- [6] J. Barth, K. Verhaege, L. G. Henry, J. Richner, "TLP calibrarion, correlation, standards and new techniques", Proc. EOS/ESD Symposium, pp. 85-96, 2000.
- [7] T. Smedes, J. van Zwol, G. de Raad, T. Brodbeck, H. Wolf, "Relations between system level ESD and (vf-)TLP", EOS/ESD Symposium Proc., 3A1, pp. 136-143, 2006.

BDD-based Cryptanalysis of LFSR Stream Ciphers

Srdan Đorđević, S. Bojanić and O. Nieto-Taladriz

Abstract - Binary Decision Diagram (BDD) is binary variant of the data structure called Decision diagram that is aimed to discrete function representation. BDDs algorithms are used as an effective way to represent Boolean functions and are very efficient in terms of space and time complexity. They raised a lot of interest in cryptanalysis of Linear Feedback Shift Register (LFSR) that is one of the most important and widely used building blocks for keystream generators. LFSR is well suited for hardware and software implementations and produce very uniformly distributed output streams. Best known LFSR-based stream ciphers are E_0 used in Bluetooth wireless LAN, A5/1 and A5/2 used in GSM standard of cell phones, the shrinking generator, etc. In this paper, different cryptanalytic variants of BDD like Free Binary Decision Diagram (FBDD), Ordered BDD (OBDD), Zero-suppressed BDD (ZBDD), etc. are discussed and further research directions are outlined.

Keywords –Cryptanalysis, Binary Decision Diagrams.

I. INTRODUCTION

A stream cipher generates bit by bit a keystream, which is used to encrypt the plaintext. The keystream produced by a stream cipher should be as random looking as possible in order to make it more resistant to attacks. The difficult task is to make stream cipher secure and at the same time provide excellent software or hardware performance. Much recent cryptography research has been focused on stream ciphers that offer faster performance in some architecture (8, 16, 32 or 64-bit) and smaller hardware implementation in terms of gates, area or power consumption.

In 2002, Krause introduced the concept of BDD-based attacks [1]. Attacks on several generators were presented, including A5/1, E_0 and the self-shrinking generator. Leater Shaked and Wool introduced their OBDD-based attack to E_0 key stream generator.

The time complexity of the algorithms is determined by the space complexity of the synthesized Binary Decision Diagrams throughout the entire process of construction.

Srdan Đorđević is with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: srdjan.djordjevic@elfak.ni.ac.rs

Slobodan Bojanić and Octavio Nieto-Taladriz are with Universidad Politecnica de Madrid, Departamento de Ingenieria Electronica, Ciudad Universitaria s/n, 28040 Madrid, Spain E-mail: slobodan@die.upm.es (S. Bojanić), nieto@die.upm.es (O. Nieto-Taladriz)

In Section 2 we concern LFSR-based keystream generators. Section 3 is dedicated to the Binary Decision Diagram (BDD) and its variants used in cryptanalysis. Brief introduction to BDD-attack against LFSR-based key stream generators is given in section 4. Two adapted and optimized attacks of Krause algorithm for the specific details of E_0 system, OBDD-attack and ZBDD-attack are described in Sections 5 and 6 respectively.

II. LFSR-BASED STREAM CIPHERS

A stream cipher is a symmetric key cipher which generates a pseudorandom bit stream (keystream) used to encrypt the plaintext. Digits of the plaintext (usually bits or bytes) are combined with a keystream by an exclusive-or operation to produce ciphertext.

Linear feedback shift register (LFSR) is a shift register whose input bit is a linear combination of its previous state. Implementation of LFSR, shown in Fig. 1, consists of a simple shift register in which a binary-weighted modulo-2 sum of the taps is fed back to the input. For any given tap, weight g_i is either 0, meaning "no connection," or 1, meaning it is fed back.

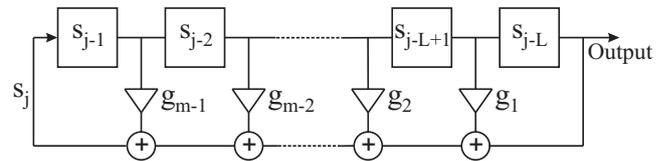


Fig. 1. Model of the linear feedback shift register (LFSR)

LFSR is widely used in stream ciphers as a pseudo-random number generator. It is well suited for hardware and software implementations and produce very uniformly distributed output streams with long periods.

The LFSR-based keystream generators consists of two components, a linear bitstream L and a nonlinear compression function C . They generate keystream according to the rule $y=C(L(k))$ for the cipher key k . Linear bitstream generator produce a linear bitstream $L(k)$ by one or more parallel LFSRs.

Since LFSR is a linear system, cryptanalysis of their output sequences is very simple. Additional block of LFSR-based keystream generators is a nonlinear compression function, C that provide cryptographic security.

Best known LFSR-based stream ciphers are E_0 used in Bluetooth wireless LAN, A5/1 and A5/2 used in GSM standard of cell phones, the shrinking generator. Most of

the known cryptographic attacks are key recovery attacks. There are two categories of key stream attacks, short and long, according to the required known key-stream.

E_0 is a LFSR-based key stream generator used in the Bluetooth protocol. It employs 4 shift registers of differing lengths (25, 31, 33, 39 bits) and a nonlinear combiner logic, that consists of summation combiner logic and blend machine. Nonlinear combiner logic is usually represented as a 4 bit finite state machine.

The sum of the four output bits of the LFSR's is input into the finite state machine to update the state of the machine. At each clock tick E_0 generates output bit of the encryption system stream using outputs of the shift registers and two internal states, each 2 bits long. Practically, the output of the four LFSRs is xor-ed with the output bit of the nonlinear combiner logic. The secret key length is generally 128 bits.

III. BINARY DECISION DIAGRAM BDD

A binary decision tree (BDT) is a tree data structure that is used to represent a Boolean function. Each nonterminal node has exactly two children. Nonterminal nodes are labeled by boolean variables, while sink nodes, or leaves are labeled by 0 or 1. Every two different vertices on a single path are labeled by distinct variables. BDT determines a unique boolean function from the variables in nonterminal nodes, while consumes as much space as truth table.

Binary Decision Diagrams (BDDs) [2] are canonical directed acyclic graphs. A graph is canonical if multiple identical nodes are not allowed. Two nodes are identical if they have the same label, and their respective child nodes are also identical. Important property of the BDD is that computation results are stored for future reuse.

A BDD is derived from a BDT by elimination two obvious kinds of redundancies. Transformations to eliminate redundancies of BDT are:

- (1) *elimination of redundant tests*: Nonterminal vertex whose two children vertices are the same can be removed.
- (2) *merging isomorphic subgraphs*: Repeated appearance of the same subtrees are merged into one subtree.

Fig. 2 shows an example of a BDT and corresponding BDD, that is obtained by implementation of the above transformations.

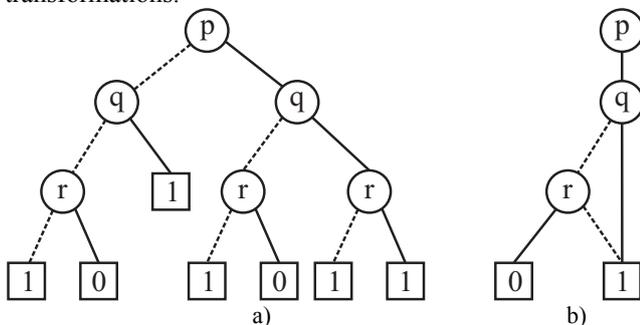


Fig. 1. a) Binary Decision Tree b) Binary Decision Diagram

It is obvious improvement of the data structure compactness.

Variants of BDD used in cryptography analysis so far are Free Binary Decision Diagram (FBDD), Ordered Binary Decision Diagram (OBDD) and Zero Decision Diagram (ZDD).

FBDD is a BDD in which along each path variables appears at most once. An Ordered BDD is a BDD in which each variable is encountered no more than once in any path and the order of variables is same along each path.

A Reduced BDD (ROBDD) is an OBDD that is reduced by two reduction rules: deletion rule and merging rule. These Reduction rules remove redundancies from the OBDD.

Zero-suppressed Binary Decision Diagrams (ZBDDs) are a special type of BDDs which were introduced for efficient manipulation of sparse item combinations [3].

An itemset p can be represented by a n -bit binary vector (x_1, x_2, \dots, x_n) , where $x_i = 1$ if item i is contained in p . A set S of itemsets can be represented by a characteristic function $X_s(p): \{0, 1\}^n \rightarrow \{0, 1\}$ where $X_s(p) = 1$ if $p \in S$ and 0 otherwise. More specifically, a ZBDD is a BDD with two reduction rules:

1. Merging rule: merge identical subtrees (to obtain canonicity);
2. Zero-suppression rule: delete nodes whose 1-child is the sink-0, and replace them with their 0-child.

By utilising these rules, a sparse collection of item combinations, which can be seen as a boolean formula, can be represented with high compression.

IV. FBDD BASED CRYPTANALYSIS OF LFRS-BASED KEYSTREAM GENERATORS

The cryptanalysis of the keystream generators consists of finding a secret key k fulfilling $y = E(k)$, for a given keystream y and given encryption algorithm E . In the algorithm proposed by Krause [1] the problem of finding a secret key k is reduced to problem of finding the minimal FBDD P for the decision whether k fulfils $y = E(k)$. The important properties of FBDD is that they can be efficiently minimized and allow an efficient enumeration.

The algorithm consists of three steps:

- Construction of minimal FBDD Q_m which decided whether bit stream Z , $z \in \{0, 1\}^m$ for all $m \geq 1$ produce a prefix of keycipher k_{cipher} .
- Construction of minimal FBDD R_m based on the feedback polynomials that decided whether Z can be produced by Linear bitstream generator.
- Construction of third set of FBDD's P which are result of intersection of the Q_m and R_m .

Kraus algorithm incrementally computes FBDD for increasing number of bits. The cryptanalysis algorithm proposed by Kraus is very efficient against LFSR-based

keystream generators. The weakness of LFSR-based generators can be explained by small memory of the compressor C as a result of online manner of keystream generation.

Synthesis operation bounds the size of the synthesis result as:

$$|\text{SYNTH}(P, Q)| \leq |R| \cdot |Q| \quad (1)$$

where $|Q|$ is size of the OBDD representing the compressor $|R|$ is an LFSR consistency check OBDD.

The authors of the algorithm reported, based on some estimation, that their attack against E_0 requires $O(2^{77})$ space complexity and $O(2^{81})$ time complexity.

V. OBDD BASED CRYPTANALYSIS OF E_0 KEY STREAM GENERATOR

Optimization of general OBDD-based attack for specific details of E_0 system was introduced by Shaked and Wool [4], which proposed OBDD-based cryptanalysis. This algorithm uses OBDD instead of FBDD and propose a new composable BDD for the compressor.

OBDD based attack uses regularities of E_0 key generator, reflected in fact that every clock tick each LFRS is stepped once and from each LFSR one output bit is input to the compressor. Variable ordering of the internal bitstream is expressed in terms of the clock tick index m ($0 \leq m \leq 127$) and index of the LFSR ($1 \leq i \leq 4$), as $j = 4 \cdot m + i - 1$. This indexing method leads to separate four equations for linear bitstream assigned to each of the four shift generators. Consequently algorithm includes construction of OBDD's for each bit of the bitstream and associated to a distinct LFSR. Different length of the shift generators requires adjustment of the algorithm.

Each internal bit is produced by one of the LFSR's, and depends on four earlier bits of the same shift register. Algorithm produces BDD, according to the LFSR's feedback polynomial, that make decision whether the internal bit z_k is consistent with the prefix $\{z_j\}_{j=1}^{k-1}$. Table I summarizes the basic consistency equations in the case of two indexing methods for each of the LFSR's.

TABLE I
CONSISTENCY RELATIONS

| LFSR | Basic consistency equation |
|------|--|
| 1 | $z_i = z_{i-8} \otimes z_{i-12} \otimes z_{i-20} \otimes z_{i-25}$ |
| 2 | $z_i = z_{i-12} \otimes z_{i-16} \otimes z_{i-24} \otimes z_{i-31}$ |
| 3 | $z_i = z_{i-4} \otimes z_{i-24} \otimes z_{i-28} \otimes z_{i-33}$ |
| 4 | $z_i = z_{i-4} \otimes z_{i-28} \otimes z_{i-36} \otimes z_{i-39}$ |
| LFSR | Normalized consistency equation |
| 1 | $z_i = z_{i-32} \otimes z_{i-48} \otimes z_{i-80} \otimes z_{i-100}$ |
| 2 | $z_i = z_{i-48} \otimes z_{i-64} \otimes z_{i-96} \otimes z_{i-124}$ |
| 3 | $z_i = z_{i-16} \otimes z_{i-96} \otimes z_{i-112} \otimes z_{i-132}$ |
| 4 | $z_i = z_{i-16} \otimes z_{i-112} \otimes z_{i-144} \otimes z_{i-156}$ |

An OBDD representing an LFSR consistency relation contains 5 variables and 11 nodes.

The second step in the OBDD-attack against E_0 system is construction of the OBDD that represents the compressor unit. This OBDD is built according to the transfer function of the compressor and known keystream bits.

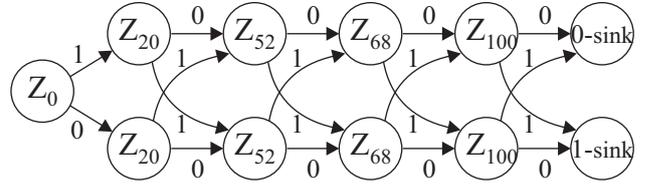


Fig. 2. OBDD representing consistency check for z_{100}

The value of the compressor unit is updated by the sum of the LFSR's output bits. The obtained BDD structure is called basic chain and practically represents the sum of 4 bits. This OBDD structure is illustrated on Fig. 3. The compressor unit consists of 16 identical basic chains, for each of states.

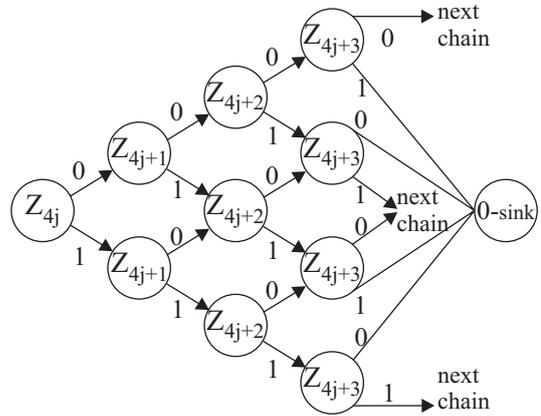


Fig. 3. Basic chain representing sum of 4 bits

The analysis of the algorithm complexity consists of the OBDD space complexity estimation. The size of synthesized OBDD $|P|$ is limited by two bounds. First is the number of satisfying assignments

$$|P| \leq m \cdot |\text{One}(P)| \quad (2)$$

where $\text{One}(P)$ denotes the set of satisfying assignments of the BDD P , and m is the number of variables that BDD contains. Linear bitstream generator introduces 4 new variables each clock tick. There is one constraint because the output bit is known. It can be conclude that the number of satisfying assignments is multiplied by 2^3 per clock tick.

The second bound is synthesis operation bound

$$|P| \leq |Q(m)| \cdot 2^{m-n} \quad (3)$$

where $|Q(m)|$ is size of the OBDD representing the compressor, m is the number of variables, n is size of the given keystream. This bound assumes that the number of introduced OBDD nodes during LFSR consistency check is duplicated by each new variable. The maximal number of the OBDD nodes during attack, $|P| \approx 2^{86}$, is find as intersection point of two bounds.

VI. ZDD BASED CRYPTANALYSIS OF E_0 KEY STREAM GENERATOR

ZDD-attack against E_0 generator [5] is based on the general FBDD-attack but with different data structure. ZDD is a variant of BDD obtained in such a way that one of the reduction rule is changed. Each path from the root to the terminal vertex 1 corresponds to one of the combinations.

The motivation of using ZDD data structure in cryptanalysis is that it is better suited for representation of sets than BDD. This data structure is especially efficient in manipulation with set of combination, that is represented by an binary vector. Each bit in this vector denotes presence or absence of an item in the combination. The set of combination can be shown with Boolean function called characteristic function.

Graph that decided whether a bit stream Z can be produced by linear stream generator, denoted as R_m , is constructed using ZDD data structure instead of OBDD or FBDD.

Data structure that represents the compressor unit and decide whether a bit stream $C(Z)$ produce a prefix of keycipher is denoted as Q_m . Since finite state machine of E_0 generator has 16 state it can be represent with 4 one bit variables q_i^n for $(1 \leq i \leq 3)$. The following function for Q_m should be computed after riding $m+1$ input symbols z_i :

$$F(q_3^{m+1}, q_2^{m+1}, q_1^{m+1}, q_0^{m+1}, z_{4m+3}, z_{4m+2}, z_{4m+1}, \dots, z_0) \quad (4)$$

Algorithm mapped the problem to a combinatorial set problem. The computing of Q_m is reduced to checking of all possible combinations of input bits and final states. Most operations on sets are defined and implemented for ZDD data structure.

The number of variables and constraints during some steps of the algorithms is changeable and can be expressed in function of the length of each shift register, L_i for $0 \leq i \leq 3$. During first $|L_0|$ steps algorithm introduce 4 new variables and one constraint, then the number of assignments is multiplied by 2^3 . After $|L_1|$ steps the output of the first register L_0 is known and represents additional

constraint, then the number of assignments is multiplied by 2^2 . In the same way after $|L_2|$ steps the number of assignments is multiplied by 2^1 . After $|L_3|$ steps, the number of constraints is equal the number of variables, then the number of assignments will be constant.

The overall time complexity of the algorithm is 2^{82} and space complexity of 2^{23} .

VII. CONCLUSION

We have presented a short overview of the BDD-based algorithms in cryptography. The paper provides description of algorithms and their complexity analysis.

The attacks are based on a backtracking approach [6], that build a binary search tree according to the feedback polynomials of LFSR and nonlinear compression function. BDD data structure is successfully performed in cryptography analysis as a compact and canonical presentation of the Binary Decision Tree.

One possible direction of future research is improvement of space consuming of FBDD-attack, which needs a lot of space for all constructed intermediate diagrams. Another open question is check whether FBDD-attack could be combined with other methods of cryptanalysis.

REFERENCES

- [1] Krause, M., „*BDD-Based Cryptanalysis of Keystream Generators*“, In EUROCRYPT, Vol. 2332 of LNCS, 2002, pp. 222–237.
- [2] Bryant, R. E., “*Graph-based algorithms for boolean function manipulation*“, IEEE Transactions on Computers, Vol. C-35, No. 8, Aug., 1986, pp. 677–691.
- [3] Minato, S., “*Zero-suppressed BDDs and their applications*“, International Journal on Software Tools for Technology Transfer (STTT), Vol. 3, No. 2, 2001, pp. 156–170.
- [4] Shaked, Y., Wool, A., “*Cryptanalysis of the Bluetooth E_0 cipher using OBDD's*“, In Proceedings of 9th Information Security Conference, LNCS 4176, 2006, pp. 187–202.
- [5] Ghasemzadeh, M., Meinel, Ch., Shirmohammadi, M., Shazamanian, M. H., “*ZDD-Based Cryptanalysis of E_0 Keystream Generator*“, In Proceedings of 3th International Conference on Mathematical Sciences (ICM 2008), Mar., 2008.
- [6] Zenner, E., Krause, M., Lucks, S., “*Improved cryptanalysis of the self-shrinking generator*“ In V. Varadharajan and Y. Mu, editors, Australasian Conference on Information Security and Privacy ACISP'01, Lecture Notes in Computer Science, Vol. 2119, 2001, pp. 21-35.

Modeling and Simulation of Digital Systems in different Domains

Ivan Paunović, Volker Zerbe

ABSTRACT – A digital system can be specified with the Finite State Machine (FSM) and/or the Statechart approach [1]. A modeled FSM is to be embedded in a Discrete Event (DE) or a Synchronous Data Flow (SDF) domain of a simulator, designtool. It is used MLDesigner, the next generation system design tool [2]. The modelling technique is shown by examples. Simulation results are presented.

KEYWORDS – Finite State Machines, Discrete Event Domain, Synchronous Data Flow Domain, Modeling, Simulation

I. INTRODUCTION

Digital systems are designed on the basis of a functional notation by a structural synthesis or by direct structural notation. The basic concept for specification of digital systems is the finite state machine approach [3], [5]. FSM's also represent a common basis model for both UML (unified modeling language) statecharts and SDL (specification and description language) processes.

In the model based design process a tool chain is used. MLDesigner which is used in the early design phases is a next generation system design tool. The different models are represented graphically in a hierarchy of block diagrams, where the blocks are connected via Input / Output interfaces. The graphical user interface of the MLDesigner application contains a frame set of different editors, including a multi-document editor to create, edit and save the block diagrams. The storage of the block diagrams is based on the Extended Markup Language (XML). An integrated multi-domain simulator facilitates the analysis and simulation of the different systems and supports for example the following domains:

- Finite State Machine (FSM)
- Discrete Event (DE)
- Synchronous Data Flow (SDF)

Furthermore, MLDesigner [4] embraces an extensive library of base modules and provides animation and plot tools to analyse simulation results.

V. Zerbe is with Faculty of Information and Automatics, Ilmenau University of Technology, Germany,
E-mail: volker.zerbe@tu-ilmenau.de

I. Paunović is with the Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, 18000 Nis, Serbia & Montenegro,
E-mail: ivan.paunovic@elfak.rs

In the paper the three domains are shortly described.

Furthermore it is shown the modelling technique by examples.

II. THE FSM DOMAIN

A finite state machine is a conceptual machine with a finite number of states. It can be in only one of the states at any specific time. A state transition is a change in state that is caused by an input event. In response to any input event, the finite state machine might transition to a different state. Alternatively, the event has no effect and the finite state machine remains in the same state. The next depends on the current state as well as on the input event. Optionally, an output action may result from the state transition.

The MLDesigner finite state machine domain includes a graphical editor and an action language for fitting and managing states, transitions and interface elements. It supports the UML Statechart semantic, hierarchical states and special events, as well as key MLDesigner features such as data types and data structures, shared memory, and interaction with other design domains.

The FSM semantic provided by MLDesigner supports synchronous and asynchronous behavior, additional events, variables and parameters for various runs of simulations. The FSM mechanism provided by MLDesigner supports all the basic standard elements of finite state machine. Events are used as triggers to cause a finite state machine to undergo state changes. In doing so, the presence of an event is interpreted as logical true and the absence of an event as logical false. The MLDesigner FSM supports 3 different kinds of events:

Input Ports - The basic events are represented by data parsed into an input port of the FSM model interface.

Special Events - If an FSM model is embedded into a discrete event (DE) environment, MLDesigner Special Event arguments can be used to trigger the FSM model.

Internal Events - In context with the FSM state slave process mechanism, internal events can be set or reset inside an FSM model, without influence of the outer environment of the finite state machine. These internal events are represented by a boolean flag associated with the name of the event. If a state slave model contains an input port with the name of an internal event, this input port gets the current flag value (true or false) of the associated internal event, before the slave model executes. If a state slave model contains an output port with the name of an internal event and this output port contains new data after

execution of the slave model, the associated internal event flag is set to the integer cast (unequal zero = true, equal zero = false) of the new data. If the output port, associated with an internal event, contains no new data after execution of the slave model, the internal event flag is reset to false.

States represent conditions or periods characterized by the concepts of duration and stability. A finite state machine can have an arbitrary number of states but at any time of execution, the FSM must reside in only one state. In the scope of a single FSM, each state of this FSM must have a unique name. This name is centered at the top of the rounded rectangle in the graphical notation of a state.

Hierarchical States - Each state can have an arbitrary number of sub-states and the sub-states can also be hierarchical. All states on the same level of a hierarchy are sibling states. States up the hierarchy are called ancestor states and states down the hierarchy are called descendant states. Leaf states are states without sub-states. In the graphical representation of hierarchical states, the borders of a sub-state must reside completely inside the boundaries of all ancestor states.

Current State - At any time during simulation, a finite state machine must reside in only one state. This state is called the current state. If the current state is a hierarchical state, then only one of its sub-states must be the current sub-state. This rule goes down the hierarchy until a leaf state is the current sub-state of a level of the hierarchy.

State Actions - Together with each state, it is possible to define two sets of operations. The entry action is executed whenever the state is entered and the exit action is a set of operations performed whenever the state is exited. These actions are defined using the C/C++ like FSM action language.

Slave Process - The MLDesigner FSM provides a slave process associated with leaf states. An MLDesigner module or a different FSM model can be used as a slave process. The slave process of the current state executes if the FSM receives new events and no preemptive transition, possessed by the current state, fires.

Transitions - State changes in finite state machines are described via transitions. Each transition specifies a source state and a target state. The graphical notation is a line or multiple line segments between the source and target state with an arrow on one end, pointing to the target state. If the source and target state of a transition is the same state, the transition is called a **self transition**. Each state possesses all transitions for which it is the transition's source state as well as those possessed by its ancestor states. Latter transitions are called **inherited transitions**. Associated with each transition is a boolean property called **preemptive**. All the preemptive transitions, possessed by the current state, are checked for firing, before the slave process of the current state is performed. In that way, the slave process executes only, if no preemptive transition fires.

Entry Type - In context with the FSM state slave process mechanism, every transition contains a property

called Entry Type. If a transition fires, the entry type of this transition specifies in which way the slave process of the next current state will be executed. The entry type can be either Default or History. For example, a leaf state containing another FSM model as slave process. If this state is entered via a Default entry type transition, the slave process FSM model will always be reset to its initial state, before execution. In the case, this state is entered via a History entry type transition, the slave process FSM model continues the execution in the last current state.

Event Expression - Events can be on Input Ports, Special Events, Internal Events... Complex and nested event expressions can be defined using brackets and the C/C++ logical operators ! (NOT), || (OR) and && (AND). The applicability of the different logical operators within transition event expressions depends on the outer domain of the associated FSM model. MLDesigner provides a special Event Expression Dialog for easy event expression composition. If the event expression evaluates to true, while the current state is a transition's source state or one of its sub-states, the transition is triggered and becomes a candidate for firing. A transition without an event expression is immediately triggered after its source state entry action is executed. These transitions are called synchronous transitions.

Guard Condition - Optionally associated with each transition is a guard condition C, specified by an FSM Action expression. If the guard condition of a triggered transition evaluates to true, the transition fires and the finite state machine's next state becomes the transition's target state. A triggered transition without a guard condition fires immediately. Triggering does not automatically cause transitions to fire, it merely enables firing. If a triggered transition cannot fire, because the guard condition evaluates to false, the transition must be triggered again by a satisfied event expression to become a candidate for firing.

Transition Action - Each transition can also have an action A, which is a set of FSM Action statements. Whenever a transition fires, its action is performed before the transition's target state entry action is executed.

Transition Priority - If more than one transition possessed by the current state is triggered, the inherited transitions up the hierarchy have a higher priority to fire.

Transition Conflict - A transition conflict occurs, when two or more transitions with the same priority are triggered and no transition with a higher priority is candidate for firing. In this case of nondeterminism, the FSM scheduler fires the first one, which guard condition evaluates to true.

Default Entrances - The Default entrance is a special state which indicates the point of entry to that level of the state hierarchy. Each level of the state hierarchy, including the FSM top level, has one default entrance, which is depicted by a small solid circle. A default entrance must not have any incoming transitions and must have only one outgoing transition to designate the default sub-state destination. The top level default entrance designates the initial state of the finite state machine. The Top Level

Default Entrance may be linked to a set of FSM Action statements that initialize the state machine (e.g. initialize memories). Upon simulation startup, the entry action of the initial state is not performed.

Histories - Histories are special states, used to resume the last sub-state of a hierarchical state. An FSM can have an arbitrary number of histories, placed at any level of the state hierarchy. A history must have at least one incoming transition and must not have any outgoing transitions. A **static history** memorizes only the previous sub-state of its hierarchy level. If this sub-state is a hierarchical state, then its default entrance destination becomes the current sub-sub-state and so on, until a leaf state becomes the current state. **Recursive histories** apply to all descendant states and refer to the previous current state of their state hierarchy. So the state, memorized by a recursive history, is always a leaf state on the same or lower level of the state hierarchy. If a transition, pointing to a history fires, actions are performed as if the transition is pointing to the state stored in the history. Trying to enter a hierarchical state via an empty history, when the hierarchical state was never visited before, results in an error being displayed and the simulation aborts.

Arguments - In addition to the basic elements, the MLDesigner FSM semantic supports typical MLDesigner arguments associated with finite state machines.

Memory Arguments - Memories can be used to represent variables in finite state machine. The MLDesigner FSM supports memories of both scopes: internal and external, and all types and data structures as is the case elsewhere in an MLDesigner block. FSM action statements have read as well as write access to the value of a memory argument. In addition to events, represented by the inputs of a finite state machine, the MLDesigner FSM supports **special event arguments** of both scopes, internal and external, and of all types and data structures. These events can also be used in the event expression of transitions and can be accessed via FSM action statements to schedule or cancel an event argument. Events of external scope can generate events in other MLDesigner models, i.e. other FSM models. Using MLDesigner special event arguments in context with finite state machines makes sense only if the FSM model is embedded into a discrete event (DE) environment.

Parameter Arguments - Like memories and special events, parameters are fully supported in MLDesigner finite state machines. FSM action statements have only read access to the value of a parameter.

III. THE DISCRETE EVENT DOMAIN

The discrete event (DE) domain in MLDesigner provides a general environment for time-oriented simulations of systems such as queuing networks, communication networks, and high-level models of computer architectures. In this domain, each Particle represents an event that corresponds to a change of the

system state. The DE schedulers process events in chronological order. Since the time interval between events is generally not fixed, each particle has an associated time stamp. Time stamps are generated by the block producing the particle based on the time stamps of the input particles and the latency of the block.

A DE primitive models part of a system response to a change in the system state. The change of state, which is called an event, is signaled by a particle in the DE domain. Each particle is assigned a time stamp indicating when (in simulated time) it is to be processed. Since events are irregularly spaced in time and system responses are generally very dynamic, all scheduling actions are performed at run-time. At run-time, the DE scheduler processes the events in chronological order until simulated time reaches a global "stop time".

Each scheduler maintains a global event queue where particles currently in the system are sorted in accordance with their time stamps; the earliest event in simulated time being at the head of the queue. The difference between the two schedulers is primarily in the management of this event queue. The default DE Scheduler mechanism handles large event queues much more efficiently than the alternative, a more direct DE scheduler, which uses a single sorted list with linear searching. The alternative scheduler can be selected by changing a parameter in the default DE target.

Each scheduler fetches the event at the head of the event queue and sends it to the input ports of its destination block. A DE primitive is executed (fired) whenever there is a new event on any of its input portholes. Before executing the primitive, the scheduler searches the event queue to find out whether there are any simultaneous events at the other input portholes of the same primitive, and fetches those events. Thus, for each firing, a primitive can consume all simultaneous events for its input portholes. After a block is executed it may generate some output events on its output ports. These events are put into the global event queue. Then the scheduler fetches another event and repeats its action until the given stopping condition is met.

It is worth noting that the particle movement is not through Geodesics, as in most other domains, but through the global queue in the DE domain. Since the geodesic is a FIFO queue, we cannot implement the incoming events which do not arrive in chronological order if we put the particles into geodesics. Instead, the particles are managed globally in the event queue.

IV. THE SYNCHRONOUS DATA FLOW DOMAIN

Synchronous data flow (SDF) is a data-driven, statically scheduled domain in MLDesigner. "Data-driven" means that the availability of Particles at the inputs of a primitive enables it. Primitives without any inputs are always enabled (including disconnected Xgraphs.) "Statically scheduled" means that the firing order of the primitives is determined once during the start-up phase. The firing order will be periodic. The SDF domain is one of the most

mature in MLDesigner, having a large library of primitives and demo programs. It is a simulation domain, but the model of computation is the same as that used in most of the code generation domains. A number of different schedulers, including parallel schedulers, have been developed for this model of computation.

SDF is a special case of the dataflow model. In the terminology of the data flow literature, primitives are called actors. An invocation of the go() method of a primitive is called a firing. Particles are called tokens. In a digital signal processing system, a sequence of tokens might represent a sequence of samples of a speech signal or a sequence of frames in a video sequence.

When an actor fires, it consumes a number of tokens from its input arcs, and produces a number of output tokens. In synchronous dataflow, these numbers remain constant throughout the execution of the system. It is for this reason that this model of computation is suitable for synchronous signal processing systems, but not for asynchronous systems. The fact that the firing pattern is determined statically is both a strength and a weakness of this domain. It means that long runs can be very efficient, a fact that is heavily exploited in the code generation domains. But it also means that data-dependent flow of control is not allowed. This would require dynamically changing firing patterns.

V. FSM AND CONCURRANCY DOMAINS

In MLDesigner, a finite state machine is always combined with other MLDesigner Models, since an FSM Model is always embedded into a wormhole of a concurrency domain or different MLDesigner Models can be used as a Slave Process inside an FSM Model [6]. This section describes, how FSM Models interact with the Discret Event domain, the Synchronous Data Flow (SDF) domain and the Finite State Machine (FSM) domain, in the case, an FSM Model is used as a Slave Process Model.

A. FSM AND DE

The MLDesigner DE domain uses an event driven Model of computation. Events occur at a point in time. A time stamp, possessed by every event, indicates the time, at which the associated event occurs.

An FSM Model embedded in a DE domain environment behaves like any other DE Model. New data on an FSM Model Input port represent the presence of a new event to trigger the FSM Model and new data on an FSM Model Output port, generated during the execution, are interpreted as new events for the Discret Event environment, whereas the FSM Model reacts to the outer DE domain as a zero delayed system. In this context, Output events, generated by the FSM Model, get the same time stamp as the Input event, which triggered the execution of the FSM Model. A Discret Event outer

domain is the only case where a FSM Model uses SpecialEvent arguments to cause State changes inside the finite state machine, because SpecialEvent arguments, possessed by an FSM Model, are scheduled by the outer domain and the DE domain is the only one, which supports SpecialEvent arguments. The Event Expression of non-synchronous Transitions should only consist of a single Event Name, if the associated FSM Model is embedded in a Discret Event environment. The FSM Model reacts every time, either an Input event or SpecialEvent occurs and all the events underlie an OR condition.

B. FSM AND SDF

A SDF system consists of a set of modules or primitives interconnected by directed arcs. MLDesigner SDF Models represent computational functions that map Input data into Output data. Unlike the DE domain, the SDF domain is not event driven and there exist always data on each Input and Output port of the SDF Model.

An FSM Model, embedded into an SDF domain environment, behaves like any other SDF Model. To ensure this behavior, the FSM Model needs an approach to differ between the presence and absence of an event, since there exist always data on each Input port of the FSM Model. In this context, the FSM Model determines the presence and absence of an event via the integer cast of the appropriate Input data. If the integer cast returns zero, the associated event is interpreted absent and in the case of a non zero result, the associated event is interpreted present. If the FSM Model execution produces no data for an associated Output port, a zero valued data is placed on this Output port, to ensure that there are always data available on each FSM Model Output port, as required by the semantic of the outer SDF domain.

VI. EXAMPLE

The first example is to modeling the system that simulates the turning on and off a lamp with just one button (toggle). The purpose of this example is introduction to the FSM model and his work with the DE domain and the SDF domain.

It is necessary to model the FSM, which will be universal for both DE and SDF domain. There are one difference, because in DE domain we need to add another entry, which will be put "clock" signal to the system so it could able to function properly. Systems which runs the FSM for different domains are different, but their principle of operation are the same.

Figure 1. shows the FSM. It is known that the lamp can be placed in the 4 possible states. The first situation is when disabled or when the button is not activated (State0). If the button is activated system cross to other state, "State01" which comes to turning on lamp or light. In this state the lamp remains until dismissal button. When this happens the system is going into the third state in which the

lamp lights and the button is not active (State02). Now, the process will go back to initial state, but through condition number four (State03). Of course in this state when it comes to re-activate the button, it comes to the exclusion off light at the lamps. The release button returns the system to the first condition, the initial state.

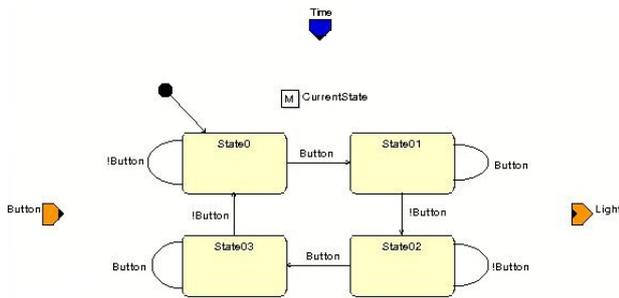


Fig. 1. FSM of Turn ON/OFF light

Framed rectangular shapes symbolize the state in which the FSM is located. Each of the state is necessary to define in the program and set the command Entry Action. In this case we have only to define the output "Light" with:

```
WriteOutput(Light,0); for first and fourth condition (State0 i State03)
WriteOutput(Light,1); for second and third condition (State01 i State02)
```

The lines that are between states represent state change in the FSM and called Transitions. Words that are next to these lines denote the defined entrance to the FSM state in the command Event Expression. Which means that the state make changes if a corresponding signal appears on the input of FSM, or FSM will remain in a given state until the input signal is defined (this goes for the line that starts and ends in the same condition and that line is called Self Transition). Of course if you only state the name of the signal then it expects to be or to appear in its logic unit, and if standing in front of the name "!" it's a logical zero.

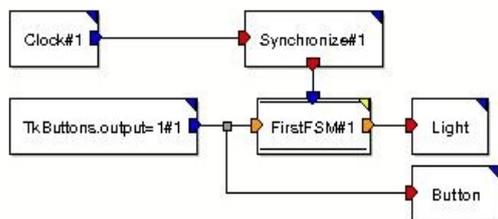


Fig. 2. DE System of Turn ON/OFF light

System for DE domain is shown in Fig. 2. In the FSM model, you need to add models that represent the excitation of system and models that can track signals in the system

and all the models in this case are from the DE library in different sublibrary. Block "Clock" is a model of the system clock signal and is in sublibrary Sources. After this block is a block "Synchronize" which serves to synchronize the clock signal with real time (synchronizes simulation time with the system time) and should drag it from sublibrary Timing. Block "TkButtons" is toggle button model and with it signals are specify (sublibrary TclTk). Besides these there are two models "Button" and "Light", which serve to show the given signal and simulation results respectively (sublibrary Sinks / Xgraph).

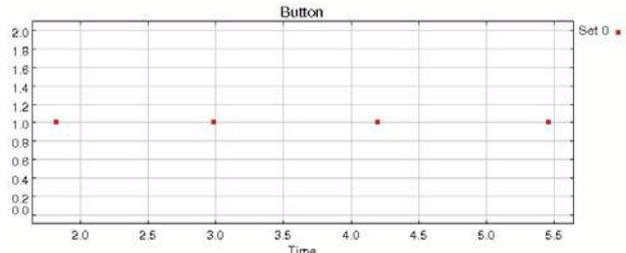


Fig. 3. Input

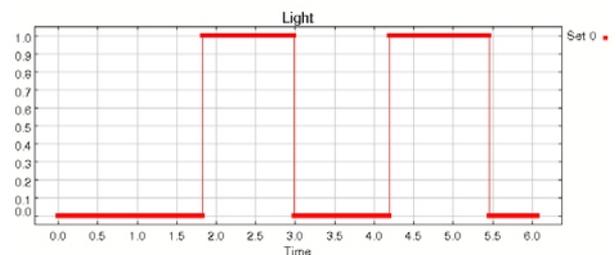


Fig. 4. Output

In Figures 3. and 4. are display the incentive and the simulation results. Comparison of time these two graphics can be seen that the time in which activated the button corresponding to the time of turning on lamps, and after another pressing of the button it comes down (lamp is turning off). This task is fulfilled in the DE domain.

Using exactly the same FSM (except removal TIME entrance), the system were designed in the SDF domain, which is shown in Fig. 5. Models of which is this system made up were used from different sublibrary of SDF library. Models, "Button" and "Light" have the same purpose as in the DE area and were taken from the same sublibrary, but only in the SDF domain (sublibrary Sinks/Xgraph).

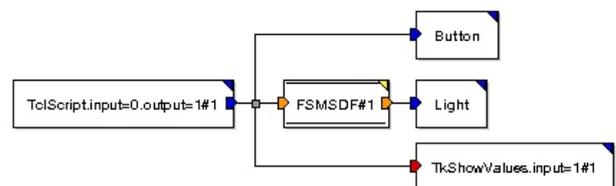


Fig. 5. SDF System of Turn ON/OFF light

For the purposes of setting signals, used models are "TclScript" and "TkShowValues", although we could use

the model of "TkButton". Unlike the previous examples, where we can simply apply "TkButton", with "TclScript" model we need to write a small program to describe the function of button. Code for this program:

```

set s $ptkControlPanel.middle.button_$starID
if {![wininfo exists $s]} {
    button $s -text "Push Me!!!"
    pack append $ptkControlPanel.middle $s {top}
    bind $s <ButtonPress-1> "setOutputs_$starID 1.0"
    bind $s <ButtonRelease-1> "setOutputs_$starID 0.0"
    setOutputs_$starID 0.0
}
unset s
    
```

This code is placed in the file with the extension ".tcl" and then we must connect "TclScript" with him through the function TcL_File. In addition to this model, the model "TkShowValues" was used also, and serves to, when we run simulations and button appears, below it display the value of a button in real time, ie, when the button is not active "0.0", and when is "1.0".

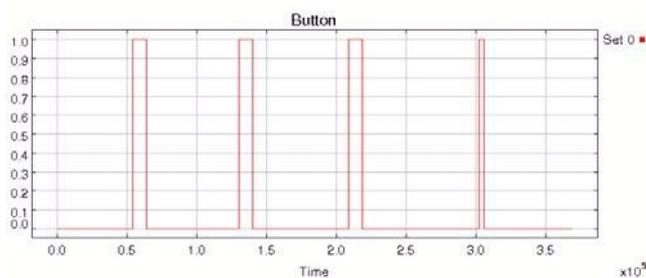


Fig. 6. Input

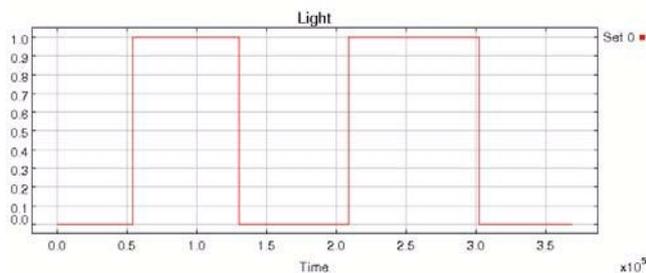


Fig. 7. Output

Figures 6. and 7. represent the incentive system and the simulation results. The first figure shows points when the button is activated and how long was the squeeze, and the other when and how long the lamp was included. Based on this basic insight, is gained some knowledge in SDF domain and in the use of some other models.

VII. CONCLUSION

This paper presents research results after praxis at the Ilmenau University of Technology. First, the different domains are analysed. Modules for a few examples were developed. It is used the FSM-Domain which could be embedded in the DE-Domain and the SDF-Domain. Later the both solutions, modelling techniques, in these different domains were analysed and compared.

REFERENCES

- [1] D. Harel, "Statecharts: A visual Formalism for Complex Systems", Science of Computer Programming, 8/1987, North Holland, pp. 231-274
- [2] <http://www.ml designer.com>
- [3] V. Zerbe, "Mission Level Design of Complex Autonomous Systems", in Proc. XLVII ETRAN Conference, Herceg Novi (Montenegro), 2003, pp. 55-59
- [4] MLDesigner Documentation, version 2.4, 2003.
- [5] V. Zerbe, "Systematischer Entwurf paralleler digitaler Systeme", in Proc. Workshop Boolesche Probleme, Freiberg (Germany), 07. Oct. 1994, pp.73-79
- [6] H. Rath, „Specification of the MLDesigner Finite State Machine Model, Student Research Project, Ilmenau University of Technology 2002

GINISED – Geo-Information System for Support of Evidencing, Maintenance, Management and Analysis of Electric Power Supply Network

Leonid Stoimenov, Aleksandar Stanimirović, Miloš Bogdanović,
Nikola Davidović and Aleksandar Krstić

Abstract - This paper presents GINISED – geo-information system which is realized to support of control and management of electric low-power supply network. The paper presents basic functionalities for network management of GINISED system. Also, its integration with enterprise IT systems is described and a use case of calculation of distribution network losses.

Keywords - Geographic Information System, Electric Power Supply Network, Distribution Network Losses.

I. INTRODUCTION

The functioning of companies engaged in the transmission and distribution of electricity depends on the existence of appropriate electric power supply network geo-data [1]. It is estimated that more than 80% of data used in a variety of processes (network design process, data input and update, maintenance and various analysis) has geographic (spatial) component. Therefore, almost any electric power supply company has a need for the existence of specialized geo-information system that should provide mechanisms for collecting, storing and manipulating spatial data.

Geo-Information Systems (GIS) are being widely used for more than forty years. They have found their purpose in environmental monitoring, transportation management, public safety, facility security, disaster management, etc. GIS enables us capturing, storing, analyzing, and displaying geographically referenced information. It allows us to view, understand, query, interpret, and visualize data in a way that is quickly understood and easily shared. GIS technology can be used for scientific research, resource management, and development planning.

GIS applications enable connecting various types of information in the spatial context and generating new information and conclusions on the basis of these connections. GIS enables fast, accurate and unique presentation of network data. GIS output of electric network can be viewed and easily interpreted compared to any other system output. GIS technology promises benefits not only in increasing operational efficiency but also in improving policy design,

Leonid Stoimenov, Aleksandar Stanimirović, Miloš Bogdanović and Nikola Davidović are with the CG&GIS Laboratory, Department of Computer Science, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: (leonid.stoimenov, aleksandar.stanimirovic, milos.bogdanovic, nikola.davidovic)@elfak.ni.ac.rs.

Aleksandar Krstić is with PD Jugoistok, Niš Zetska 4, Serbia, E-mail: aleksandar.krstic@iugoistok.com.

decision making, communication, and dissemination of information.

The business processes such as network planning, network study, repair operations, maintenance, network changes, connection, and disconnection are based around a network model. Consider a typical power outage scenario; when an outage is reported from SCADA/DMS or any other real time system, an integrated GIS system with the help of its unique prediction engine can identify the most probable part in electrical network (ex. device, feeders etc.) along with the location of problem immediately. It will also report the network that may have been affected downstream to it. A customer service representative will know quickly what the problem is and be able to tell the customer approximately how long it will take to attend. Meanwhile, crews can go immediately to the scene with the proper repair equipment and make repairs in considerably less time.

GIS can help in day-to-day operation and maintenance as it provides the accurate, reliable spatial and non-spatial information to the utility operational staff. It can help engineer in finding most optimum route to take for inspections or maintenance jobs. In addition to added functionality, integrated GIS may be easier and less costly to support. There is no such requirement of having proprietary hardware, software platform, and special skills for GIS implementation. By virtue of openness of GIS, interoperability is easily possible with other systems.

II. GINISED – GIS FOR ELECTRIC UTILITY

For the needs of PD Jugoistok Niš, CG&GIS Lab, Faculty of Electronic Engineering in Niš, with the support Ministry of Science of Republic of Serbia, developed a geo-information system GINISED [1]. GINISED is a specialized geo-information system which, in addition to standard alphanumeric data concerning electrical parameters of electric power supply network, allows recording, processing, analysis and graphic presentation of specialized information about the electric power supply network, such as spatial data, temporal data, image and multimedia.

The main purpose of GINISED project is to apply modern GIS technologies and approaches in order to develop specialized tools for collecting, editing, visualizing and analyzing spatial data of distribution network. This system has three groups of tools [2]:

1. Tools for collecting (digitizing, computer scanning, recognition and vectorization, using GPS and other specialized devices, etc.) and editing spatial data of distribution networks.

2. Tools for visualizing spatial data of distribution networks in selected geographic area.

3. Tools for spatial analysis of distribution networks, potential or real events in distribution network and risk factors in selected geographic area.

GINISED basic components are:

- Centralized geospatial database – It allows thematic and spatial electric power supply network data storing.

- GINISED Editor – Desktop application for recording, searching and editing spatial and geo-electric power supply network data.

- GINISED Web – WebGIS application that allows quick and easy positioning on a specific geographic area, search and selection of parts of electric power supply networks. This application implements information integration functionalities and uses data from centralized geospatial database.

- (Open Geospatial Consortium) WFS and WMS [3] services and other Web services that provide electric power supply network data.

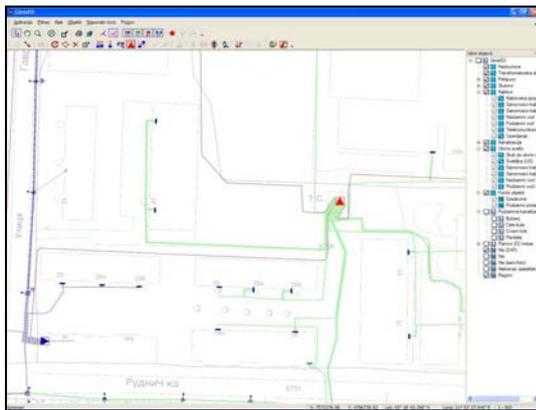


Fig. 1. GINISED Editor

As mentioned before, GINISED platform uses a shared central geo-database that allows integration and dynamic updating of multiple GIS data sources. This database is designed in accordance with broad accepted industry IT standards [4]. It can be extended, rearranged and customized according to customer's requirements. DBMS software can be chosen by the customer from any thirdparty RDBMS vendor (Oracle, SQL server, etc.). This geo-database has to fulfill two contradictory requirements: on one side, it has to be fully flexible, taking into account a wide variety of network elements and configurations, and on the other side, the data management has to be very fast and efficient.

GINISED system consists of two independent applications: Editor and Viewer. GINISED Editor is specialized tool for geographic editing of distribution network (Fig. 1).

It is a desktop application developed in accordance with carefully studied needs and requirements of customers. It is used for creation and editing of geographic schemes of the network, editing parameters of network elements and their connectivity (Fig. 2). It provides multi-user and user-friendly, complete graphical environment for browsing and editing spatial data of distribution network with carefully chosen set of commands similar to popular vector graphic editors, but also with specialization for editing of the distribution networks schemes.

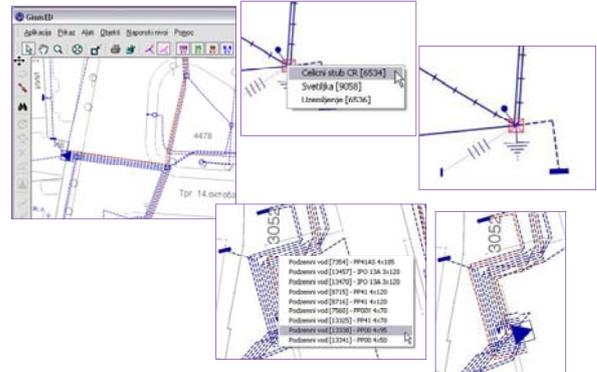


Fig. 2. Editing parameters of network elements

GINISED Viewer is Web GIS application with three-tier software architecture. This application, developed according to modern Web 2.0 standards, delivers feature rich user-interface (Fig. 3).



Fig. 3. GINISED Viewer

III. INFORMATION INTEGRATION AND GINISED

GINISED platform is developed using object oriented ap-approach and modern component technologies. GINISED system was developed using GeoNis platform for the interoperability of GIS applications [5]. GeoNis platform provides the mechanisms and infrastructure for the exchange of information in the environment of local government [6], but can be applied for integration of information on a single company level. This platform is developed for purpose of intelligent integration of information from a number of heterogeneous GIS (geographical and spatial) and non-

spatial data sources [7]. Companies, institutions or their parts that have some information of interest are considered to be data sources. This framework provides means for separating spatial visualization from spatial data sources. This is very important, because, these components make possible development of GIS applications capable to change their data sources and to dynamically build user interface according to user privileges.

Because of its openness, GINISED system is very easy to integrate with other IT systems within a single electric power supply company [7]. A true enterprise GIS for utility company means access to GIS that deals with geographically dispersed assets or customers by every employee. Many utilities consider the GIS system as the “ultimate” source database, acting as a common repository for all enterprise applications. This is done by integrating GIS technology into the mainstream business operations of the company.

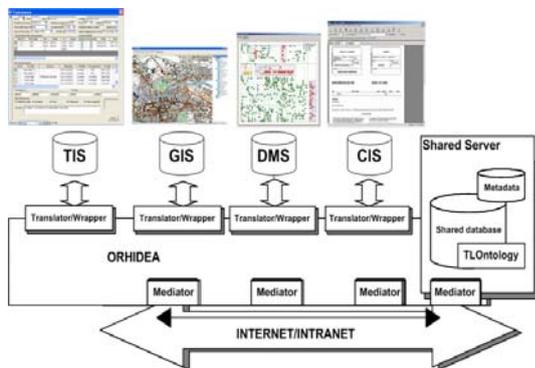


Fig. 4. GeoNis platform for the GIS interoperability

Electric power supply network analysis demands usage of technical information about the electric power supply network elements. In order to implement this analysis, GINISED application uses information from a number of heterogeneous and distributed information sources. The position of GINISED system among other information system is presented in Fig. 4. GeoNis platform is located between GINISED system, which operates as a C3 (Command Control and Communication) module, and relevant data sources. GeoNis environment nodes can be existing applications. For each of those applications, it is necessary to develop translators and domain (local) ontologies. Nodes may also be new applications developed in accordance with the OpenGeoSpatial standards and component software development methodology [3].

Once the Enterprise GIS is implemented it would act as the base system for all the organizational assets, and would cater to the requirements of other departments. The GIS electrical data model is designed keeping this as an important requirement. Enterprise GIS uses a shared central geodatabase that allows integration and dynamic updating of multiple GIS data sources. This considerably reduces the time-consumed for data update, increases the compatibility of data with other systems and also simplifies the transla-

tion issues. Since it is based around industry IT standards and web services the non-GIS applications and systems would be able to easily access GIS functionality, and GIS applications. Every system (for e.g. SCADA, Network Analysis, AMR, etc.) in a utility has a specialized role to play. The GIS system is never a substitute to any of these systems, but once integrated enterprise wide, it would enhance their capabilities, hence increasing the benefits.

IV. APPLICATION FOR LOW VOLTAGE NETWORK LOSSES CALCULATION

On the basis of developed GINISED system, which allows the integration of information from different IT systems in the PD Jugoistok Niš, a prototype application for calculation of losses in the low voltage (LV) network was developed. GIS module holds the central part in the application for the calculation of electricity losses. It is a downscaled GIS application that has retained only the minimum of required GIS functionalities. This application visualizes spatial data of electric power supply network and provides users with a simple interface to GINISED information integration system.

For the purposes of analysis and calculation of losses, data from three different information systems is currently being used. Other systems as information sources will be added with the further development and improvement of the application.

GIS system is used as a source of data related to LV network topology and technical description of LV electric power line sections (section length, section resistance, electric power line type, type of conductor, conductor diameter etc). LV network spatial data was recorded in the field and is being regularly updated. LV network GIS data is related to information about consumers. CIS system contains consumer information. Integration of GIS and CIS allow determination of consumer’s exact position on LV electric power line. It also allows determination of geographical location of connection that the consumer is related to. This enables easy identification of all customers related to particular LV electric power line.

When all consumers related to a particular LV electric power line are identified, their unique consumer codes are used as input data to obtain their daily load characteristic diagrams from AMR system. AMR system uses modern electronic consumption meters. These meters allow storing of load characteristic diagrams for a period of time (load profile). Hence, load characteristic diagram is imposed as one of the basic analytical data for the calculation of energy balance and LV electric power line losses [8]. Figure 5 shows typical load characteristic diagram.

Based on technical information related to LV electric power line (section length, type and diameter of the conductors) and consumer load characteristic diagram, losses calculation module determines losses on a particular LV electric power line.

Losses calculation module is not based on approximate

methods. Instead, it uses recursive method for calculating the electric current that flows through each LV electric power line section [9]. This module uses LV electric power line data topology as graph (from the transformer station to the end consumer). This graph consists of transmission facilities, sub facilities and consumption meters related to company clients. Based on unique customer codes, daily load characteristic diagrams are obtained from AMR system. If daily load characteristic cannot be obtained, particular consumer is related to one of standard load characteristic diagrams.

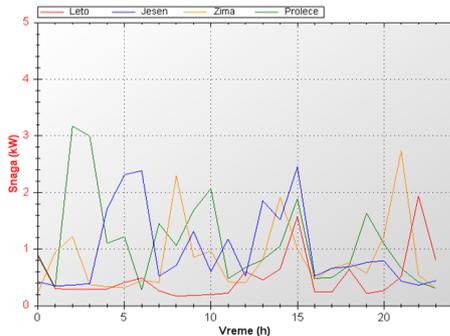


Fig. 5. Typical load characteristic diagram

Figure 6 presents data related to a particular LV electric power line used for calculation of electricity losses. It is possible to alternate LV electric power line section parameters (electric power line length, electric power line type, conductor type, number of cores and conductor diameter) and analyze how these changes affect the percentage of technical losses. It is also possible to define standard load characteristic diagrams for all four seasons.

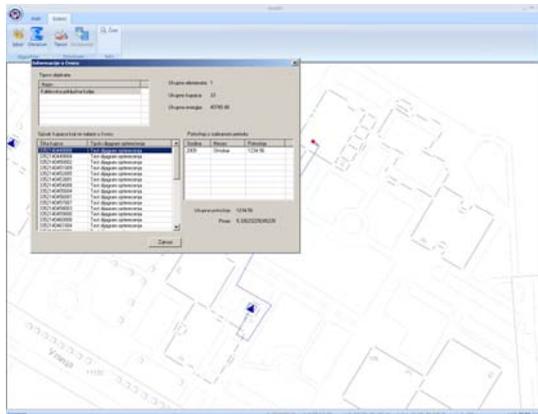


Fig.6. Application for calculation of electricity losses

V. CONCLUSION

Current Electrical Utility business trend is characterized by, change in market conditions, regulatory requirements/policies which require utility to achieve greater competitiveness and effectiveness. Utilities are beginning to see a need for IT business systems to collect, store, and publish various types of data, sharing data among them and to

maintain consistency of data. Number of disparate systems exists to achieve specific tasks. Most of the utility companies implement IT systems like SAP, SCADA, DMS, AMR and GIS for their business operations. But, these IT systems are working in isolation with each other and data is main-tained and accessible to only those who use these systems. The main objective is to efficiently integrate IT systems within the enterprise, and GINISED GIS represents the step toward enterprise application integration within PD "Jugoistok", Niš.

ACKNOWLEDGEMENT

Research presented in this paper were partially funded by the Ministry of Science of the Republic of Serbia and PD Jugoistok Niš, within the project in the field of technological development "Intelligent integration of geo-, business and technical information on the company level," ev. No 13003.

REFERENCES

- [1] Stoimenov, L., Đorđević-Kajan, S., Stojanović, D., Kostić, M., Vukašinović, A., Janjić, A., "Geographic Information System for evidencing, maintenance and analysis of electric power network", YU INFO 2006, Kopaonik, 2006 (in Serbian).
- [2] Stanimirović, A., Stojanović, D., Stoimenov, L., Đorđević-Kajan, S., Kostić, M., Krstić, A., "Geographic Information System for Support of Control and Management of Electric Power Supply Network", IX Triennial International Conference on Systems, Automatic Control and Measurements SAUM, ISBN 86-85195-49-7, Niš, 2007.
- [3] Open Geospatial Consortium, WMS and WFS Specifications, 2002, www.ogc.org
- [4] Faculty of Technical Science Novi Sad – Department of energetics, "Database Model for technical data for management of distribution network", Novi Sad, March 2004
- [5] Stoimenov L., Đorđević-Kajan S., "An Architecture for Interoperable GIS Use in a Local Community Environment", Computers & Geoscience, Elsevier, 2005, Vol. 31, No. 2, March 2005, pp.211-220
- [6] Stoimenov, L., Stanimirović, A., Đorđević-Kajan, S., "Development of GIS Interoperability Infrastructure in Local Community Environment", From Pharaohs to Geoinformatics, FIG Working Week 2005 and GSDI-8 Cairo, Egypt April 16-21, 2005, TS41.2.
- [7] Stanimirović, A., Stoimenov, L., Đorđević-Kajan, S., Kostić, M., Krstić, A., "Company level geodata integration within GINISED application", JUIINFO 2007, Kopaonik, Serbia, CD Edition, ISBN 978-86-85525-02-5, 2007
- [8] Jardini, A., Tahan, C. M. V., Gouvea, M. R., Ahn, S. U., Figueiredo, F. M., "Daily Load Profiles for Residential, Commercial and Industrial Low Voltage Consumers", IEEE Trans. on Power Delivery, Vol.15, No. 1, Jan. 2000
- [9] Tošić, S., Krstić, A., Nikolić, B., "Application for calculation of low voltage losses", CIRED 2008, Vrnjačka Banja, Serbia, 2008 (in Serbian)

3D Simulations for Wireless Ad Hoc Networks in Grid Environment

Sonja Filiposka and Dimitar Trajanov

Abstract - In this paper the usage of grid environment for fast simulation of wireless ad hoc networks in 3D terrains is presented. The possibilities and gains of the grid environment have proven to be very resourceful when simulating wireless ad hoc networks using the NS-2 simulator especially when considering lengthy simulations for performance evaluation with a heavy traffic load involving computing intensive calculations like the estimation of received power in a 3D environment based on the radio wave diffraction.

Keywords – wireless mobile ad hoc networks, network simulator, Durkin's radio propagation model, performance evaluation, grid environment.

I. INTRODUCTION

One of the most popular and vibrant fields of parallel computing is the grid environment [1]. The idea of exploiting the unharnessed computing power of a heterogeneous mass of idle computers via the Internet caught the attention of anyone who has faced an 'impossible' task that without the aid of parallelism will be finished long after the time it is acquired.

Using the grid environment a user can divide his huge task in a number of smaller pieces. These pieces are then distributed to a number of grid computing elements that work in parallel. After the subtasks are finished the user can collect the results which are obtained for a far lesser time than in the standard sequential approach that does not involve parallel execution.

There are a number of applications for this type of computing and we can find a number of different grid environments deployed in various institutions and countries. When considering the scientific community the largest grid community is the EGEE grid (Enabling Grid for E-science) that involves a great number of countries and offers huge amount of CPU resources to the disposal of scientists. The grid resources are mainly used for simulation applications that are very computing intensive and can be performed in parallel.

The process of network simulation has always been a computing intensive task that grows tremendously with the desire to bring the simulation scenarios closer to real life

Sonja Filiposka and Dimitar Trajanov are with the Department of Computer Science, Faculty of Electrical Engineering and Information Technologies, University of Ss. Cyril and Methodius, Karpos 2 bb, 1000 Skopje, R. Macedonia, E-mail: (filipos, mite)@feit.ukim.edu.mk.

network deployment examples. There are many popular network simulators that are used in the scientific community for the purposes of protocol development, performance evaluation, behaviour estimation as well as many other similar objectives.

When simulating wireless mobile ad hoc networks one of the most popular simulators is the NS-2 network simulator [3]. This simulator enables a creation of a wireless scenario for a 802.11 enabled ad hoc network with mobile nodes, a special routing protocol and a fully specified network traffic.

However, the radio propagation models offered in the NS-2 simulator are treating the simulation environment only as a flat rectangular area wherein the nodes are on the same height relative one to another and there are no obstacles between them.

In order to bring our simulations a large step closer to real life scenarios, we developed an extension for the NS-2 simulator that enables us to place and move the ad hoc network nodes in an irregular 3D terrain defined by the means of a DEM [10][11] file that holds the digitalized elevation values for a specified terrain. The radio propagation is then calculated according to the Durkin's propagation model [9] that is based on signal diffraction.

On the other hand, the introduction of 3D terrains and the terrain aware radio propagation had a great impact on the duration of the simulations since the Durkin's calculations have proven to be very computing intensive. Thus, in order to be able to get results from our simulations in a reasonable amount of time, we ported the Durkin's extended NS-2 simulator to the SEEGRID environment [2] which is the South-Eastern Europe part of the EGEE grid.

In this paper, we present the results that show how the grid environments can tremendously speedup the process of obtaining simulation results when using the NS-2 simulator. Via a number of different simulation scenarios we investigated how the traffic load, node speed and terrain complexity influence the duration of the simulations.

The rest of the paper is structured as follows. In Section 2 we present a small introduction to MANETs and our extension of the NS-2 simulator with the Durkin's radio propagation model and 3D terrains. In Section 3 we present the way we ported the simulator to the grid environment. Section 4 presents the results that show the impact of the extension of the NS-2 simulator on the time duration of the simulations. Also a comparison of the influence of different factors on the duration of the simulations is presented. Finally, Section 5 concludes the paper.

II. MANETs and the Durkin's Propagation Model

A. Wireless Mobile Ad Hoc Network

The ability to communicate with people on the move has evolved remarkably during the last decade. The mobile radio communications industry has grown by orders of magnitude and made portable radio equipment smaller, cheaper and more reliable [4]. The large scale deployment of affordable, easy-to-use radio communication networks has created a trend of a demand for even greater freedom in the way people establish and use the wireless communication networks [5].

One of the consequences to this ever present demand is the rising popularity of the ad hoc networks. A mobile wireless ad hoc network (MANET) is an infrastructure-less network that can be established anywhere on the fly [6]. It consists of wireless mobile nodes that communicate directly without the use of any access point or base station. Thus, the nodes are supposed to establish a network environment by the means of self organization in a highly decentralized manner. In order to achieve this goal every node has to support the so-called multihop paths. The multihop path concept is introduced to allow two distant nodes to communicate by the means of the intermediate nodes to graciously forward the packets to the next node that is closer to the destination. This is controlled by a special ad hoc routing protocol [7] that is concerned with discovery, maintenance and proper use of the multihop paths.

The independence of existing infrastructure, as well as the ability to be created instantly, that is, on demand, has made the ad hoc networks a very convenient and irreplaceable tool for many on-the-go situations like: rescue teams on crash sites, vehicle to vehicle networks, lumber activities, portable headquarters, late notice business meetings, military missions, and so on. Of course, every one of these applications demands a certain quality of service from the ad hoc network and usually the most relevant issue are the network performances in terms of end-to-end throughput. However, the trade-off of having no infrastructure and no centralized manner of functioning has influenced the ad hoc networks performances greatly on many aspects.

B. Terrain Aware Radio Propagation Models

As for all wireless mobile communications the mobile radio channel places fundamental limitations on the performances of the ad hoc network. Modelling the radio channel has historically been one of the most difficult parts of mobile radio system design. Propagation models have traditionally focused on predicting the average received signal strength at a given distance from the transmitter, as well as the variability of the signal strength in close spatial proximity to a particular location.

When simulating wireless mobile networks habitually we come to use one of the large-scale propagation models that estimate the radio coverage area of a transmitter for an arbitrary transmitter-receiver separation distance [8]. These practical and fast, yet terrain unaware, frequently used propagation models are the free space propagation model or the ground reflection (Two-ray) propagation model.

The free space propagation model is used to predict received signal strength when the transmitter and receiver (T-R) have a clear, unobstructed line-of-sight path between them. In a mobile radio channel, a single direct path between T-R is seldom the only physical means for propagation, and hence the free space propagation model is in most cases inaccurate when used alone. The two-ray ground reflection model is a useful propagation model that is based on geometric optics, and considers both the direct path and a ground reflected propagation path between T-R. This model has been found to be reasonably accurate for line-of-sight microcell channels. At large distances, the received power falls off with a rate of 40 dB/decade. This is much more rapid path loss than is experienced in free space.

However, radio transmission in a mobile communications system often takes place over irregular terrain. Therefore, the terrain profile of a particular area needs to be taken into account when estimating the path loss since the transmission path between the transmitter and the receiver can vary from simple line-of-sight to one that is severely obstructed by buildings, hillsides or foliage.

In order to bring our observations a large step closer to the real-life ad hoc network deployment, we decided to use a propagation model that incorporates the nature of propagation over irregular terrain and losses caused by obstacles in the radio path. We created an implementation of the Durkin's model [9] as an extension for the NS-2 simulator [3], thus allowing us to conduct more realistic simulation scenarios and analyze the way the terrain profile affects the ad hoc network performances.

The definition of the terrain is given by the USGS DEM standard [10], [11], which is a geospatial file format developed by United States Geological Survey (USGS) for storing raster based digital elevation models (DEM). The raster type of data consists of rows and columns of cells wherein a unique value is stored. Each cell gets a numeric value that can be represented by a unique identifier. The resolution of the raster data is the cell width and length in earth units. Usually the cells are square terrain areas, but other shapes can also be used.

The Durkin's model is based on one of the basic mechanisms of radio propagation, diffraction. Diffraction allows radio signals to propagate around the curved surface of the earth and to propagate behind obstructions. The concept of diffraction loss as a function of the path difference around an obstruction is explained by Fresnel zones. Fresnel zones represent successive regions that have the effect of alternately providing constructive and destructive interference to the total received signal. In

mobile communication systems, diffraction loss occurs from the blockage of secondary waves such that only a portion of the energy is diffracted around an obstacle. That is, an obstruction causes a blockage of energy from some of the Fresnel zones, thus allowing only some of the transmitted energy to reach the receiver. Depending on the geometry of the obstruction, the received energy will be a vector sum of the energy contribution from all unobstructed Fresnel zones.

When shadowing is caused by a single object such as a hill or mountain, the attenuation caused by diffraction can be estimated by treating the obstruction as a diffracting knife edge. This is the simplest of diffraction models, and the diffraction loss in this case can be readily estimated using the classical Fresnel solution for the field behind a knife edge.

The execution of the path loss estimation according to the Durkin's model consists of two parts. The first part addresses a topographic DEM file turned into a topographical database and reconstructs the ground profile information along the path between T-R. The second part of the algorithm calculates the expected path loss along that path.

Specifically, we move along the line that connects T-R in discrete steps. The first step is to decide whether a LOS path exists between T-R. The LOS condition is violated whenever there is an obstacle higher than the T-R line. Second part of the algorithm is checking to see whether first Fresnel zone clearance is achieved. If the Fresnel zone of the radio path is found to be unobstructed, then the resulting loss mechanism is approximately that of free space. The method for determining first Fresnel zone clearance is done by first calculating the Fresnel diffraction parameter v , for each of the ground elements.

If the terrain profile failed the first Fresnel zone test, the algorithm calculates the free space power and the received power using the plane earth propagation equation. The algorithm then selects the smaller of the powers as the appropriate received power for the terrain profile. If the profile is LOS with inadequate first Fresnel zone clearance there is loss that is added (in dB) to the appropriate received power. The loss is evaluated using the highest value of the diffraction parameter v , which means that we settle for the worst case of all existing diffraction knife edges.

Our first implementation of the Durkin's model as presented in [12] has proven to be very slow when trying to simulate mobile networks and has been redesigned and optimized in order to provide a much faster estimation of the received power. The main reason for optimization was the overwhelming number of times the algorithm needed to access the topographical database. This behaviour was due to the large amount of interpolations done for points inside the Fresnel zone.

Also, the other calculations mainly consist of time consuming functions like square roots and logarithms that also add to the time needed for processing one pair of

transmitting receiving nodes. When taking into consideration that in NS-2 this is done in a way that for each transmitting node the receiving power level at every other node that is part of the network has to be calculated in order to decide whether there is interference or no, we came across a performance problem since the simulations were taking a tremendous amount of time.

In order to lessen the computing time, at the beginning we discard the 'impossible' situations, that is we immediately calculate the receiving power according to the free space propagation model and if it is below the interference threshold we do not need to check whether the signal is even less stronger than our first order optimistic approximation. Also, whenever the diffraction parameter reaches its "highest" value, we no longer need to search for another knife edge since the one encountered is "bad enough".

Another performance enhancement is introducing a cache in the case of simulation of static nodes. Since for static simulations the once already calculated received powers for a given transmitter cannot change, we decided to keep these values in a cache in a form of a $N \times N$ matrix where N is the number of nodes in the simulation. At the beginning of the algorithm we first check to see whether we already have the necessary values in the cache. If they are available we simply reuse them, otherwise we calculate and add the new values in the cache. In this way, the speed of completing our simulation sets was brought back to the original when simulating in the usual 2D environment.

As for the cases of mobile nodes, the careful code optimizing by reducing the call of the topographical database to the minimum and by reducing the number of time consuming complex calculations, i.e. the square root whenever calculating the distance between two points, have proven to be satisfactory. The optimized Durkin's propagation model for NS-2 now runs with speed that is comparable to the popular two ray ground model.

III. SIMULATING MANETS IN GRID ENVIRONMENT

A. Using the SEEGRID Infrastructure

In order to make the necessary set of simulations for observing the relationship between the two ray ground and the Durkin's model, as well as for the relationship between the influence of the terrain and the mobility and traffic model on the network performances with nodes that move with different node speed, the number of distinct simulations that need to be run is close to 600 for only one iteration. When striving to obtain average results from multiple runs the number rises up to 6000. Given that the time for execution of one heavy load simulation is 5 to 7 hours, the time needed to make the necessary simulations starts to be measured in terms of months. This was the main motivation for our need to port the simulator to a grid environment that will allow us to perform simulations in

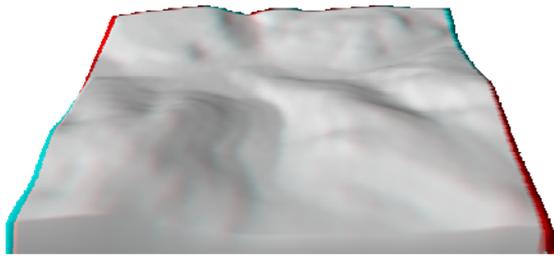


Fig. 1. Real terrain example 1 (1000 m x 1000 m)

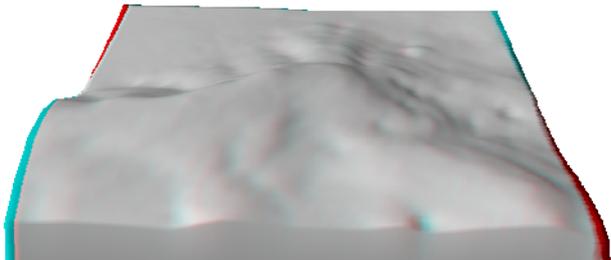


Fig. 2. Real terrain example 2 (1000 m x 1000 m)

parallel.

Thus, we decided to move the NS-2 code to our local branch of the SEEGRID infrastructure [2]. The grid infrastructure offers a great number of free computing elements that can be used for scientific purposes. In our case we wanted to use a homogenous part of the grid and we limited the execution of the simulation scripts only on our local branch that has 12 hyper threading processors that allow for 24 processes to be run at once. The use of homogenous computing elements was needed for the ability to measure the speedup of the simulations when running them in parallel. However, when the objective is to obtain results of the simulations only, then the complete seegrid environment comprised of heterogeneous computing elements can be used.

The computing elements in the grid are given the task that needs to be executed via a so-called job description language that defines the executable and its input and output parameters. The computing environment is run under different unix-like systems, depending on the local branch of the grid.

The possibility of using a so-called parametric job definition has made the execution of the simulations very easy since everything that is needed is the simulator, one general tcl script with tuneable parameters, one text file with all parameter variations needed and a job wrapper that will set the execution environment. In order to use the grid infrastructure we only needed an executable NS-2 that can be obtained with static compilation which is supported by the make file of the NS-2.

Learning the job description language and some basic commands for managing in the grid environment has made our simulation time execution more than 20 times faster. At the same time the easy way of defining all of the simulations at once and letting the grid manager worry about scheduling allows for efficient time fulfillment while waiting for the information that the jobs are done.

The parallel execution of multiple simulation scripts is a way for receiving results much more rapidly and it is our strong belief that one should use this type of simulation execution as much as possible. Today's advances in technology especially in multicore graphics cards allow for a creation of a certain types of "miniature" grids inside the personal computer using the various available processing units. One of our future works is to try to develop a parallel

execution environment using all of the resources available on a given PC.

B. Simulation Scenarios and Parameterization

In order to determine the efficiency of our parallel simulations we created several sets of scenarios. The scenarios were defined as typical scenarios for the purposes of evaluation of the ad hoc network performances. We measured the execution time of each parameterized simulation. Then we were able to compare the parallel versus sequential execution time as well as to pinpoint the simulation characteristics that mostly influence the length of the simulation duration.

The varied parameters include: different radio propagation model – two ray versus Durkin's; different terrain type, flat vs. real terrains; different node mobility, static nodes and nodes moving with speed of 1, 2 and 5 m/s; different traffic load varying from 0.1 to 7 Mbps.

For the simulations we used DEM files with dimensions of 1.000m x 1.000m with a 1:1:0.1 resolution and highest relative point of 200m. We have 100 nodes that are uniformly dispersed in the simulation area. The node transmission range is set to the standard 250m given by the use of the IEEE 802.11b standard wireless equipment. The antenna height is set to 1.5m and it has no relative offset against the wireless node. For route discovery and path set up we are utilizing the AODV protocol [13]. The offered network load is varied from 0.1 to 7 Mbps using UDP data packets with 1 KB size. The simulation time is set to 1.5 hours as to the average battery life of a notebook. During the mobile simulations, the nodes are moving according to the random direction model in the terrain boundaries. The average node speed is varied from 0 (static nodes) to 1, 2 or 5 m/s with deviation of 0.1 m/s.

On Fig. 1 and Fig. 2 the terrain shapes of two real terrains that represent the terrain profile close to Eldorado, USA are shown using the DEM file visualizing software 3Dem [14]. For comparison and verification purposes we use the results obtained for a perfectly flat terrain. In this way we can determine the terrain features impact on the network performances

Using the seegrid user interface we defined each simulation using the parametric job description. All different types of scenarios are defined in the parameters

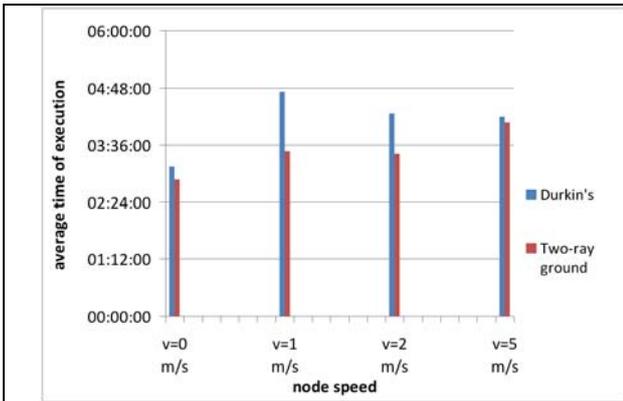


Fig.3. Comparison of the average time of execution for the Two-ray and Durkin's radio propagation models for different node speeds and flat terrain

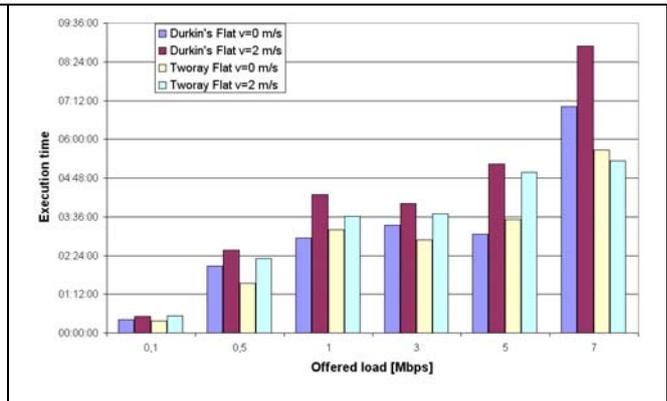


Fig.4. Comparison of the average time of execution for the Two-ray and Durkin's radio propagation models for different offered load in the network

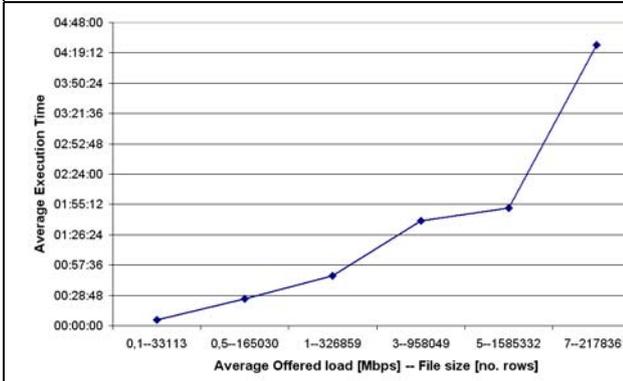


Fig.5. Average time of execution of the MANET 3D simulations depending on the network load (output file size)

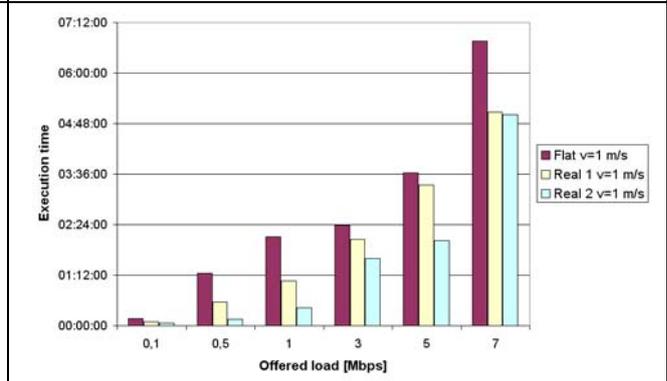


Fig.6. Average time of execution of the MANET 3D simulations depending on the network load (output file size) for different terrain shapes

files, and each parameterized scenario is run independently on one computing elements. We observe the timestamps when the job is starting with execution and before the moment when the results are sent back to the network proxy manager.

IV. RESULTS

The results obtained from our simulation scenarios show some interesting conclusions concerning the time duration of a network simulation.

Our first investigation was to conclude how our Durkin's extension influences the execution time of the simulation in the worst case scenario. The worst case scenario for this radio propagation model is when it is used with a perfectly flat terrain, since in this case there is no diffraction and the inspection of the LOS condition together with the clear Fresnel zone are done for the complete T-R line without any effect.

On Fig. 3 the comparison of the average time of execution is shown when using the traditional Two-ray ground model and the Durkin's model for different node

speeds. It can be seen that when the nodes are mobile they affect the execution time of the simulation but not in a drastic manner. Also it can easily be concluded that the introduction of the Durkin's propagation model does not introduce a great amount of additional time for simulating the network. The average excess time needed when using the Durkin's model (in it worst case scenario) is 10% over the time needed for the two ray ground simulations to end.

Unlike the previous example where it can be seen that the node speed does not greatly impact the execution time of the simulations, Fig. 4 shows the execution time for the simulations depending on the offered load in the network for perfectly flat terrain when using the two ray ground and Durkin's radio propagation model. The figure shows two characteristic behaviors: the increased offered load greatly influences the length of the simulation time and, as the offered load increases, the node speed begins to influence the execution time more noticeably. The second observation is more evident for the Durkin's model. One can also notice that the difference in time of execution between the two ray and the Durkin's model increases with the offered load. This is due to the fact that when the

offered load increases dramatically, the number of times that we need to access the topographical databases increases tremendously and this causes lengthier simulations.

While changing the terrain type, node speed and offered load in the network, we came to the conclusion that the output trace file has the biggest impact on the performances (in terms of execution time) of the network simulator. This behavior is closely related to the fact that the access to the output file is one of the slowest parts of the executing simulation. When considering the fact that the file size is closely related to the offered load in the network since the content of the trace file are the events of sending, receiving and dropping of a packet. On Fig. 5 we present the average execution time of the simulations that run with the Durkin's radio propagation model as a function of the offered load and the average file size that is expected to be encountered for the specified offered load in the network. However, this is not always what is encountered in many simulations since there are many other factors that influence whether a packet is going to be received or dropped. When talking about dropped packets, we must stress that in situations when it is difficult to obtain a route to a destination, the number of dropped packets in the trace file rises rapidly. In this case the execution time of the particular scenario becomes very long.

Fig. 6 presents the influence of the terrain shape on the performances of the network simulator with the Durkin's propagation model. The shape of the terrain has a great influence on the execution time of the simulation scenario especially because of the optimizations in our code that cut down the number of iterations when we encounter 'worst cases' of diffraction. The average simulation time for real terrain DEM1 is 74% of the simulation time of flat terrain. For DEM2 real terrain, the simulation time is only 57% of the flat terrain simulation time.

V. CONCLUSION

In this paper the performances of the network simulation NS-2 extended with the Durkin's radio propagation model where investigated. The performances of the simulator were observed when simulating wireless ad hoc mobile networks in 3D environment defined using a DEM standard file. Because of the overwhelming duration of the simulations, the extended network simulator is ported to the Seegrid environment in order to benefit the possibilities for parallel execution of a number of parameterized simulation scripts.

The performance of the extended simulator was compared to the original version that uses the traditional two ray ground propagation model. When observing the two models under equal scenarios, we conclude that our optimized extension of the Durkin's model needs around

10% more execution time than the estimated value for the two ray ground model, which we feel is a relatively small amount compared to the gained benefits of the more realistic simulation.

The results show that the main impact on the duration of the execution time of the simulator scripts can be found in the offered load in the network. Also, when using terrains, the terrain shape greatly influenced the execution time of the script.

REFERENCES

- [1] Foster I., Kesselman C., *The Grid 2, Blueprint for a New Computing Infrastructure*, Morgan Kaufmann; 2 edition, December, 2003
- [2] South Eastern European Grid Enabled Infrastructure Development, <http://www.see-grid.org/>
- [3] NS-2 network simulator, Available: <http://nslam.isi.edu/nslam/index.php>
- [4] Hekmat R., *Ad-hoc Networks: Fundamental Properties and Network Topologies*, Springer, 2006.
- [5] Brown, J. S., Duguid, P. *Social life of information*, Harvard Business School Press, 2000.
- [6] Ozan, K., Tonguz, G., Ferrari, *Ad Hoc Wireless Networks: A Communication-Theoretic Perspective*, John Wiley & Sons, 2006.
- [7] Kumar A. B. R., Reddy L. C., Hiremath P. S., "Performance Comparison of Wireless Mobile Ad-Hoc Network Routing Protocols", Int. Journal of Computer Science and Network Security, VOL.8 No.6, June 2008
- [8] Rappaport, T. S. *Wireless Communications: Principles and Practice*, Prentice Hall, New York, 2002.
- [9] Edwards, R., Durkin, J., "Computer Prediction of Service Area for VHF Mobile Radio Networks" Proceedings of the IEEE, Vol. 116, No. 9, pp. 1493-1500, 1969.
- [10] U.S. Geological Survey National Mapping Division: Part 1 General, Standards for Digital Elevation Models
- [11] U.S. Geological Survey National Mapping Division: Part 2 Specifications, Standards for Digital Elevation Models
- [12] Filiposka S., Trajanov D., Vuckovic M., "Performances of clustered ad hoc networks on 3D terrains", Second International Conference on Simulation Tools and Techniques SimuTOOLS'09, 2-6 March 2009, Rome, Italy
- [13] Perkins, C., *Ad hoc On-Demand Distance Vector (AODV) Routing*, Internet-Draft Experimental RFC 3561, July 2003.
- [14] 3DEM, Available at: <http://gisremote.blogspot.com/2008/02/3dem-software-ver-203-and-download.html>

Twenty years of ANN research and application in LEDA

Vančo Litovski

Abstract – This is to overview the research results in The Laboratory for Electronic Design Automation (LEDA) at the University of Niš, in the field of application of Artificial Neural Networks in electronic design. In all, 49 papers were published related to ANN application and theory of learning while 21 authors were involved six of them from outside of LEDA.

Keywords – Artificial neural networks, electronic design, modelling, testing, diagnosis, prediction.

I. INTRODUCTION

It all started with the invitation by Prof. Đuro Koruga to the International Summer School and Workshop on Neurocomputing Theory and Application that took place in Dubrovnik on September 1-10, 1990.

TABLE 1
CHRONOLOGY

| Date | Event/person |
|-------------------------------|--|
| November, 1990 | The First International neurocomputing symposium (INS) organized |
| August, 1992 | The MOS transistor model was published (Rađenović, J., Mrčarica, Ž., Milenković, S., and Zografški, Z.) |
| May, 1994 | Implementation to channel routing (Randelović, Z.) |
| September, 1995 | The second order synapse was published (Milenković, S.,) |
| September, 1995 | Implementation to pattern recognition (Milenković S.,) |
| June, 1996 | Simulated annealing learning based on noise signals was published (Milenković, S., Obradović, Z., and Risojević, V.,) |
| February, 1997 | Implementation to automation of the microelectromechanical systems assembly (Rađenović, J., Mrčarica) |
| August, 1997 | Implementation to electro-magneto-mechanical systems modelling and simulation (Mrčarica, Ž., and Ilić, T.,) |
| September, 2000 | Implementation to modelling of two terminal dynamic linear circuits (Zarković, K., Ilić, T.) |
| September, 2002 June, 2003 | Implementation to modelling of two terminal resistive nonlinear circuits – The double hook attractor and the Josephson junction - (Andrejević, M., and Stojilković, S.,) |

| | |
|-----------------|--|
| July, 2002 | Implementation to modelling of two terminal dynamic nonlinear circuits (Andrejević, M.) |
| September, 2003 | Implementation to modelling the A/D and D/A interface (Andrejević, M., Damper, R.I., and Petković, P.) |
| May, 2004 | Implementation to analogue diagnosis (Andrejević, M., and Zwolinski, M.) |
| December, 2004 | Implementation to testing of MEMS (Andrejević, M., and Zwolinski, M.) |
| June, 2006 | Implementation to mixed signal diagnosis (Andrejević, M., and Zwolinski, M.) |
| June, 2007 | Implementation to environmental prediction (Milojković, J.) |
| June, 2009 | Implementation to prediction in microelectronics (Milojković, J.) |
| September, 2009 | Implementation to prediction in power consumption (Milojković, J.) |

After that we organized the First International neurocomputing symposium (INS) that took part at The Faculty of Electronic Engineering on November 1990. It was organized by LEDA. Participants from several countries were present with all papers invited. This event, while modest in scope and from the point of number of participants, became the main milestone in the development of research in the area. Among other, but probably the most important, was the present done by Dr Zlatko Zografski. He delivered to us for free his programs for feed-forward ANN learning (LEARNNET) and running (RUNNET). These programs are successfully used in LEDA all the time. Table 1 gives the chronology of the developments related to ANN research at LEDA.

Most of the implementation of the ANNs at the time were oriented to decision making and pattern recognition and in order for that scientific discipline to survive in LEDA we needed some application in electronic design. One is to mention that we already had implementations of rule based artificial intelligence methods (the competitor of the ANNs at the time) for integrated circuits cell placement compaction.

Vančo Litovski is with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia,
E-mail: vanco.litovski@elfak.ni.ac.rs.

II. HISTORY

Following the claim that ANN can approximate any mapping, what is called generalization property, we came to the idea to model the MOS transistor. At the time the model implemented in SPICE suffered from important problems that were consequence of discontinuous derivatives of the transistor's characteristic at the transition between the linear and the saturation region. SPICE is Newton-Raphson based program and needs continuous derivatives to maintain convergence. The ANN implementation developed did not suffer from that problem. The output characteristics obtained are given in Fig. 1.

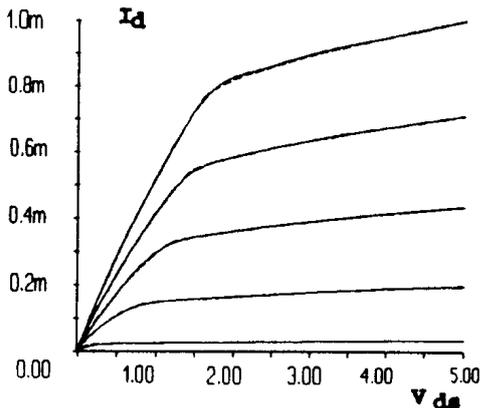


Figure 1. MOS transistor's output characteristics modelled by ANN

Fig. 2 and Fig. 3 represent the output current derivative with respect to the output and input voltages, respectively, for the ANN and SPICE model. The ANN solution solved the problem but not only that. It gave us courage to continue in the subject.

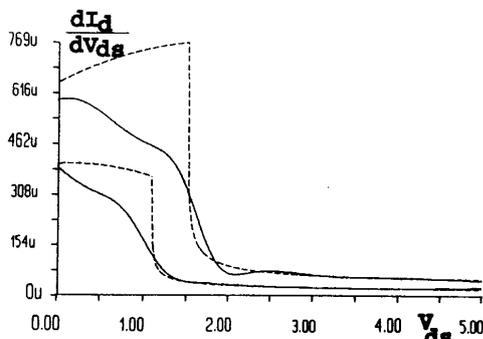


Figure 2. Output conductance of a MOS transistor obtained from the SPICE and ANN model

Theoretical research both in ANN architectures and in learning methods started. As a result we got the second order synapse. There are claims that we were the first to introduce second order synapse as shown in Fig. 4. The implementation results were marvelous. One can see on Fig. 5a the solution of the classification problem when ANNs with linear synapses are used. The task is to classify (separate) dots and circles that are distributed on nested circles. Fig 5b represents the solution to the same problem

with ANNs with second order synapses.

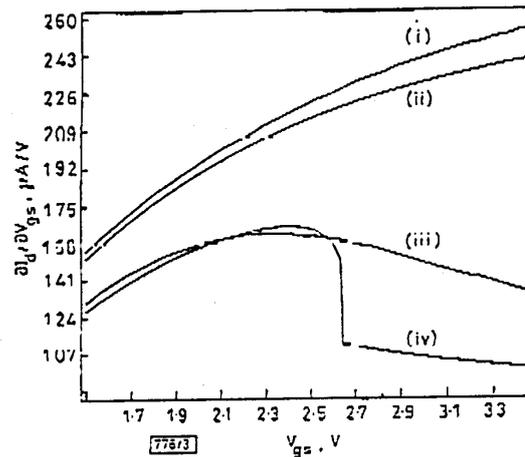


Figure 3. Transconductance of a MOS transistor obtained with SPICE and ANN model

$$f(\mathbf{u}) = \sum_{i=1}^n w_i u_i,$$

$$f(\mathbf{u}) = \sum_{i=1}^n w_i^{(1)} u_i + \sum_{i=1}^n w_i^{(2)} u_i^2$$

$$f(\mathbf{u}) = \sum_{i=1}^n w_i^{(1)} u_i + \sum_{i=1}^n w_i^{(2)} u_i^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^n w_{ij}^{(3)} u_i u_j,$$

Figure 4. S. Milenković's introduction of second order synapse

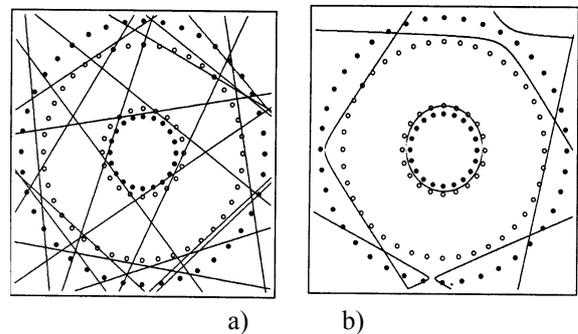


Figure 5. Classification with linear (a) and second order (b) synapses

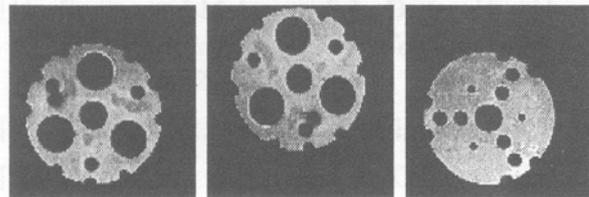


Figure 6. Silicon parts intended to be used in a microelectromechanical system

Based on these results and thanks to the behavioural mixed signal simulator Alecsis developed at LEDA we started international collaboration in the field of microelectromechanical simulation and design with the Technical University of Vienna. As a result a doctorate was granted to S. Rađenović, in Vienna related to implementation of ANNs to automation of the microelectromechanical systems assembly. Parts that were to be classified based on fast comparisons are shown on Fig. 6.

From that moment on, implementation of ANNs was no mystery for LEDA researchers and Ž. Mrčarić, from Vienna, proposed to attack the old problem which we already had solved but in a very complicated manner. It was the dynamics within the magnet with a moving-armature problem. There were two characteristics within the magnet model that were to be captured by ANNs: the magnet's Φ_L-i_L and its $F_{mag}-i_L$ characteristics.

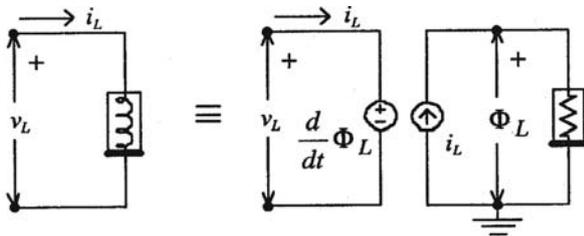


Figure 7. Circuit model of nonlinear inductor

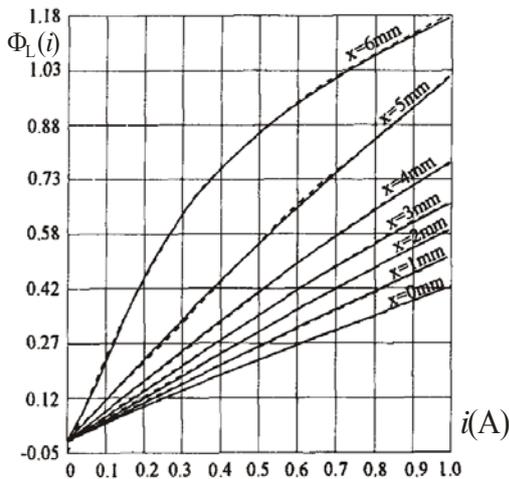


Figure 8. The approximated Φ_L-i_L characteristic of a magnet with moving armature

The task was to get the resistive characteristics and to implement them within the dynamic model of a nonlinear inductor as shown in Fig. 7. After creation of the ANNs (Fig. 8 shows the Φ_L-i_L , while Fig. 9 depicts the $F_{mag}-i_L$) simulation was performed and the simulation results, published in SIMPRA, were awarded the Savastano Award by The European Federation of Simulation Societies in 1998.

Similar concept was implemented in the simulation of the Chua's double hook circuit shown in Fig. 10. There is a resistive nonlinearity depicted in Fig.11. To create a model

of the Chua's cell that is easily implementable in simulation of complex cellular ANNs, we decided to model only the resistive nonlinearity. The rest of the cell is, of course, easy to model. The modelling results are shown in Fig. 11, while Fig. 12 represents simulation results in a form of chaotic diagram.

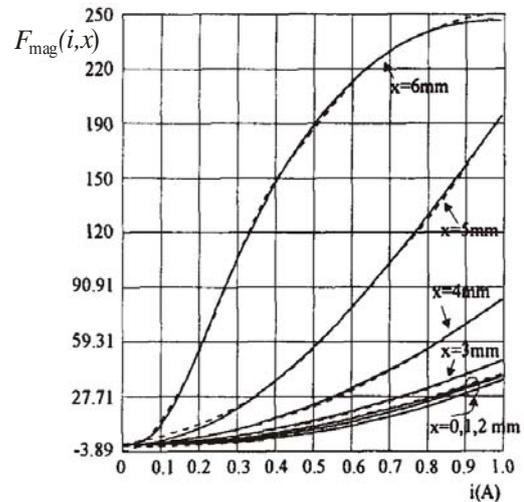


Figure 9. The approximated $F_{mag}-i_L$ characteristic of a magnet with moving armature

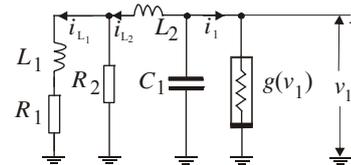


Figure 10. The double hook circuit

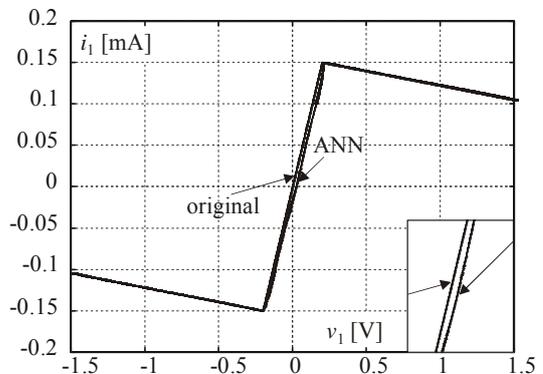


Figure 11. Constitutive relation of the nonlinear conductance and ANN model

The next development was going towards dynamic circuits modelling by a network that models the dynamic behaviour too. Linear circuits were modelled first while culmination of that research came with two results.

The first one is the model of a nonlinear two terminal dynamic network. As a target, the floating mass actuator implemented in hearing-aid systems was chosen. As shown in Fig. 13 it is an iron bar serving as a moving armature of an electromagnet whose axial move is restricted by two rubber balls. The nonlinearity here comes from the

saturation of the magnet and from the friction of the balls. The actuator is dynamic not only because of the inductance but also because of the redistribution of the air between chambers during the vibration.

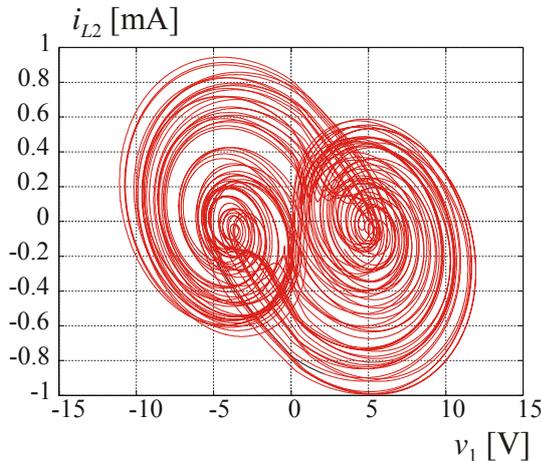


Figure 12. Simulation results for the double hook circuit

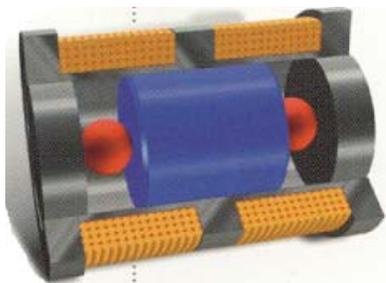


Figure 13. The floating mass actuator

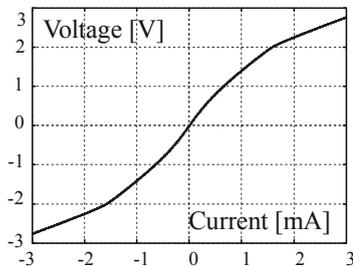


Figure 14. The static characteristic of the NDEC.

Since the proprietor of the IP related to this actuator refused to give us a sample to be measured, starting from its published characteristics we first synthesized an electronic circuit that, in our opinion, mimics the actuator. It was named Nonlinear Dynamic Electronic Circuit i.e. NDEC. The DC $i-v$ characteristic of NDEC is shown to be nonlinear in Fig. 14.

To capture the dynamic properties of the actuator we implemented a chirp signal as shown in Fig. 15. It is a frequency modulated signal that is supposed to cover the whole “pass-band” of the frequency characteristic of the actuator.

As opposed to all previous application where feed-

forward ANN were used, to model the dynamic behaviour of NDEC recurrent topology of the ANN was to be selected. The structure used is depicted in Fig. 16. It was to be trained with all points of the response depicted in Fig. 17. Note that at least ten points per (any) period were to be generated in order to have complete information for training. The responses of NDEC and the ANN synthesized to model it are shown to overlap in Fig. 18.

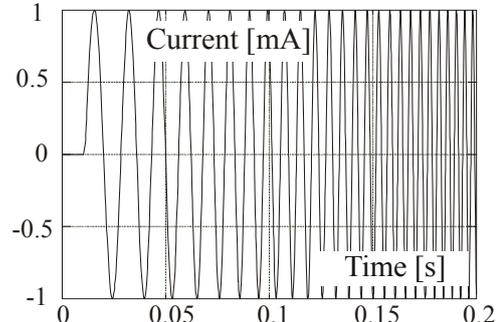


Figure 15. The exciting signal used for modelling.

The model so developed was successfully implemented for simulation of an ensemble consisting of a driving operational amplifier and the floating mass actuator in order to create environment for optimization of the driver’s output behaviour for better accommodation to the actuator.

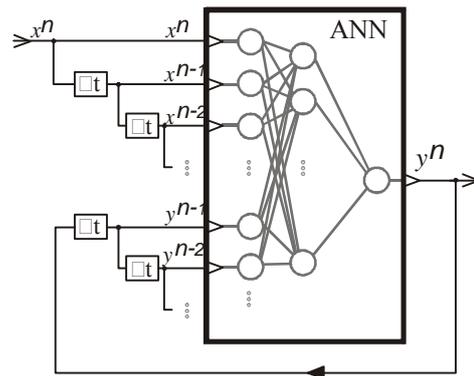


Figure 16. The topology of the proposed ANN.

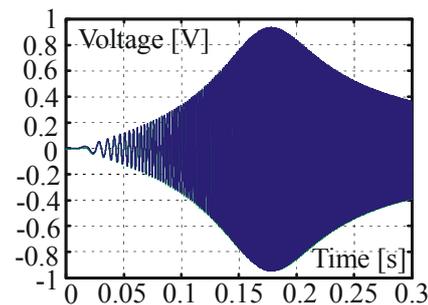


Figure 17. Time domain response of the test NDEC

The second application is related to the simulation of mixed signal circuits where D/A interface may be frequently encountered. The problem is that the analogue load, as depicted in Fig. 19, needs an analogue circuit as a driver.

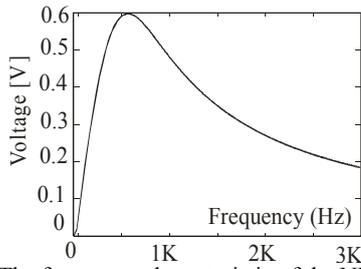


Figure 18. The frequency characteristic of the NDEC and the frequency characteristic of the ANN model overlap

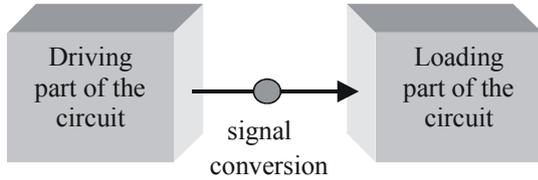


Figure 19. Driver and load interface with conversion of the signal

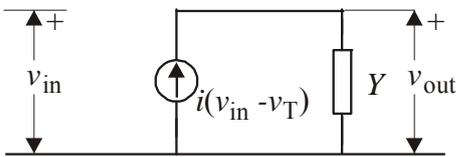


Figure 20. Circuit representation of the model

To create such a driver we needed a circuit model of the output circuitry of the digital part of the D/A interface. In general this solution may be stated as modelling with four terminal dynamic circuits. The circuit of Fig. 20 was proposed.

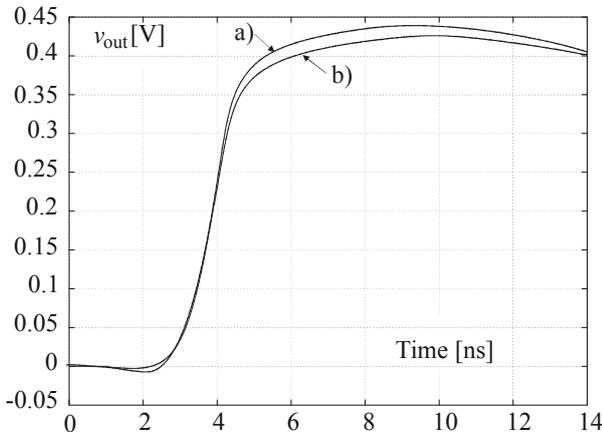


Figure 21. Responses: a) of inverter loaded by a diode and b) of ANN model loaded by a diode

In that circuit the current i is controlled by the effective input voltage $v_{in}-v_T$, and it was chosen to be a tangents hyperbolic function. v_T is a parameter here chosen to be $V_{DD}/2$. On the other side, Y , representing the output admittance of the digital part of the interface, was modelled as a two terminal dynamic nonlinear circuit with the ANN of Fig. 17. The modelling and simulation results are best seen in the example where a CMOS inverter (digital),

represented by the model of Fig. 20, was loaded with a diode (TTL). From the point of view of loading current that is the most difficult condition for the model. Note, while modelling, unloaded circuit was observed. The simulation results are depicted in Fig. 21. Note the value of the diode voltage which is by far lower than the ordinary value of the CMOS supply voltage being $V_{DD}=5$ V.

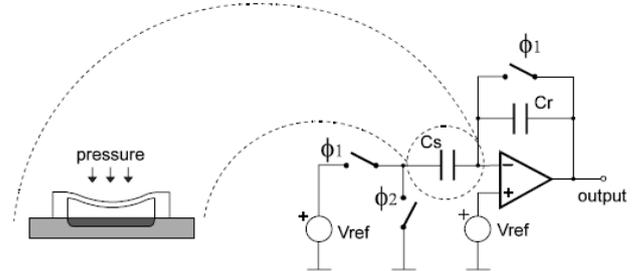


Figure 22. Capacitive pressure sensor and its electronic surroundings

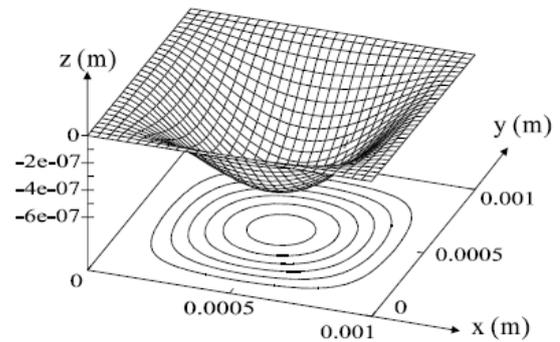


Figure 23. The membrane under pressure. Simulation results.

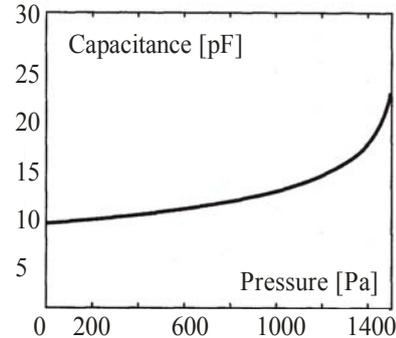


Figure 24. Capacitance versus pressure characteristic

The next development was related to the application of ANN modelling for testing purposes. In fact the ANN was used for acceleration of the verification of the test sequence in MEMS.

Namely, the system of Fig. 22 consists of electronic and mechanical part. While the electronic one is in general defined by ordinary differential equations, the mechanical needs partial differential equations to be described. When solving, the partial equations, after discretization, create a mass of ordinary equations and so, in fact, their solution dominates the simulation time.

On the other side, the number of possible faults in the electronic part of the system is incomparably larger than

those in the mechanical. To verify a test sequence targeting the faults in the electronic part one needs repetitive simulation of the whole system. That becomes a serious problem since one has to solve repeatedly the system of partial equations related to the mechanical subsystem.

To avoid that we proposed the mechanical part to be modelled as a lumped element that will reduce the overall number of differential equations to be solved. The procedure was as follows.

By simulation of the mechanical system alone, the dependence of the capacitance on the pressure was extracted as shown in Fig. 24. That is a characteristic of a linear capacitor whose capacitance is controlled by pressure. This curve was approximated by ANN and the model obtained implemented in a mixed signal simulator. In that way instead of solving 995 ordinary equations (the mesh in the mechanical part, as depicted in Fig. 22, was 30 by 33) for every test signal, only a system with 5 ordinary equations was to be solved.

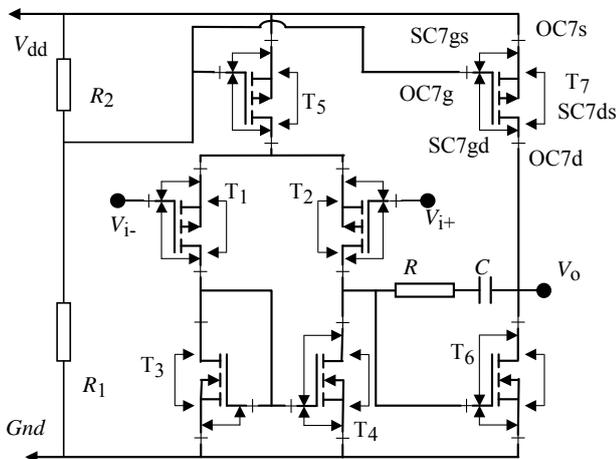


Figure 25. The operational amplifier circuit. SC=short circuit, OC=open circuit

When implementation of ANNs to the problem of diagnosis in analogue circuits is considered, the following was achieved. It was considered, first, that the main responsibility in creating the list of most probable faults in a circuit is on the design engineer. So, the possible fault list and the corresponding fault dictionaries are supposed to be created in design centres in collaboration with foundries. That means no hypotheses are to be created i.e. no faults are to be conceived and searched for, by field engineers when on-site fault effects are observed.

If so, one can memorize the fault dictionary by an ANN and, during exploitation, one may search the ANN in the opposite direction if faulty behaviour of the system is observed. That will lead to a diagnosis. Of course, “opposite direction” means running the ANN with the measured signals at its input what is equivalent to searching the dictionary in order to get the “guilty” fault.

The method was tested on the example of an operational amplifier as depicted in Fig. 25. The so called simulation before test method was applied to create the fault

dictionary meaning that faults were inserted in the circuit repetitively and responses were obtained by simulation. Part of the fault dictionary is given in Table 2. In fact, one test point (the output terminal) was selected while three quantities were measured: the DC output voltage (V_o), the DC gain (A), and the 3 dB cut-off frequency (f_{3dB}). A number (code) was assigned to every fault and was to be learned by the ANN.

TABLE 2
PART OF THE FAULT DICTIONARY FOR THE CIRCUIT OF FIG. 25

| Type | A_m | $f_{3dB} m$ [MHz] | $V_{oDC} m$ [V] | Code (m) |
|-------|--------|-------------------|-----------------|----------|
| FF | 419 | 0.01527 | 0.127 | 0 |
| 1 L+ | 0.0053 | 6.791 | 0.0497 | 37 |
| OC1G | 0.047 | 501.187 | 0.127 | 49 |
| OC3G | 0.049 | 544.042 | 0.093 | 47 |
| SC1DG | 0.042 | 320.440 | 0.0458 | 6 |
| SC2DS | 0.071 | 312.071 | 3.3 | 27 |
| SC5DS | 0.656 | 0.57 | 0.0186 | 55 |
| 6 W- | 5770 | 0.0018 | 0.2146 | 13 |
| OC5D | 0.056 | 507.298 | 3.3 | 25 |
| SC5GS | 0.109 | 0.036 | 0 | 2 |

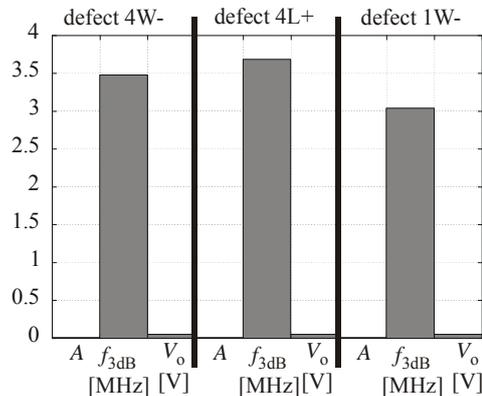


Figure 26. Fault effects of parametric faults 4W-, 4L+ and 1W-

Fig. 26 expresses the difficulties encountered during diagnosis. Namely, fault effects are depicted for three different faults being seen to be very similar. Nevertheless, the ANN developed performed the separation indubitably.

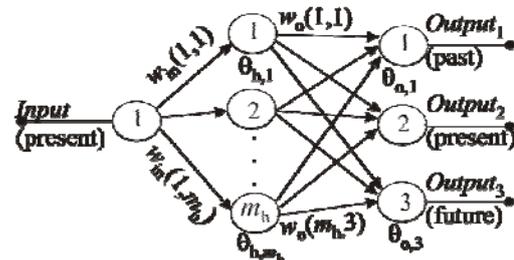


Figure 27. The Feed Forward Accommodated for Prediction (FFAP) structure

The newest efforts in implementation of ANN in electronic design was in prediction. New structure of ANN was proposed named Feed Forward Accommodated for Prediction (FFAP) as depicted in Fig. 27. It learns past

present and future values so, by nature, is accommodated for prediction.

This topology was implemented to several problems starting with prediction of quantities of obsolete computers as depicted in Fig. 28 where the period 1991 to 1999 was covered for this quantity in the USA. The y-axis is expressing millions of cubic feet.

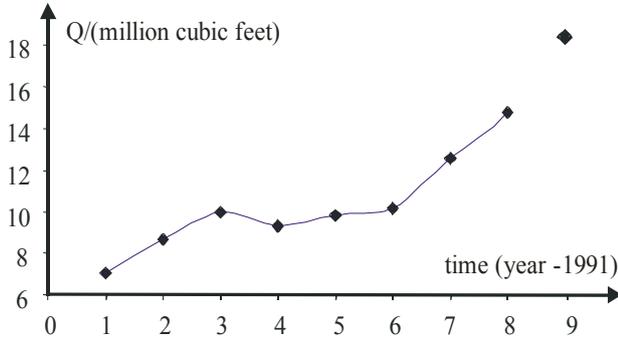


Figure 28. Quantity of obsolete computers

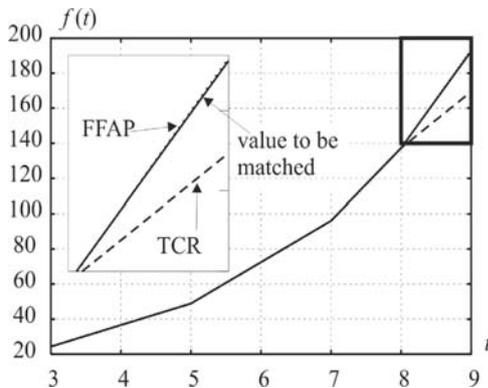


Figure 29. The number of transistor per microprocessor chip in time problem

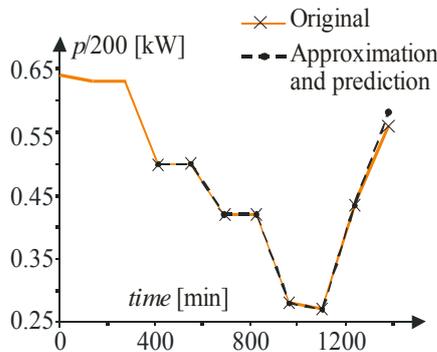


Figure 30. The actual consumption (Solid line), and the approximation (Dashed line) obtained by the EFFAP network.

Using the FFAP structure the value for the 9th year was to be predicted based on the preceding 8. The target value was **18.4** and with a network with ten hidden neurons the value of $f(9) = 18.2274$ was obtained, the difference, expressed in percentage, being **0.94%**. That we consider an excellent result.

The same method was implemented to the prediction of

the number of transistor per microprocessor chip, as a function of time, problem. This was substitution to the application of Moore's law since it may be used for long term prediction only. The error obtained in this case was only 0.33%.

Finally, the FFAP ANN, with an extension, was applied for short term prediction of electricity load on the suburban level. Fig. 30 depicts the actual and the curve obtained by the FFAP method for two-hour-ahead prediction based on one day long information extended with the consumptions in the same hours of the same days in the previous weeks. The last segment of the dashed line finishes with the prediction. It is a miss of the target value by 3.85%.

III. CONCLUSION

A historical overview was given of the efforts, ideas, and results of LEDA researchers in implementation of artificial neural networks for pattern recognition, electronic modelling, simulation, testing, diagnosis, and prediction was given.

ACKNOWLEDGEMENT

We wish to thank Prof. Đuro Koruga from the University of Belgrade for the fact that he first suggested the research in the field of ANNs to us.

REFERENCES

General mixed signal electronic simulation and ANN applications

- [1] Litovski, V., and Zwolinski, M., "VLSI Circuit Simulation And Optimization", Chapman and Hall, London, 1997.
- [2] Litovski, V., "New Methods in Modeling and Simulation of Electronic Circuits and Systems", Scientific Review, No. 29-30, 2001-2002, pp.189-207.
- [3] Andrejević, M., and Litovski, V., "Electronic Modelling Using ANNs For Analogue and Mixed-Mode Behavioural Simulation", Proc. of Neurel 2002, Belgrade, Serbia and Montenegro, Sept. 2002, pp. 113-118.
- [4] Litovski, V., and Pantić-Tanner, Z., "Artificial Neural Network Application In Electronic Circuit Analysis", World Congress On Neural Networks, Portland, Oregon, 11-15 July, 1993.

Implementation of ANNs to two- and four- terminal resistive modeling

- [5] Litovski, V., Rađenović, J., Mrčarica, Ž., Milenković, S., and Zografski, Z., "MOS transistor modelling using neural network" (in Serbian), Proceedings of the XXXVI Conf. of ETAN, Vol. II, Kopaonik, Yugoslavia, 28.9-1.10, 1992, pp. 19-25.
- [6] Glozić, D., Rađenović, J., and Litovski, V., "Implementation of MOS transistor modelled by neural

- network in hybrid simulator Alecsis" (in Serbian), Proceedings of XXXVII ETAN Conf., Beograd, Yugoslavia, Sept., 1993.
- [7] Litovski, V., Rađenović, J., Mrčarica, Ž., and Milenković, S., "MOS Transistor Modelling Using Neural Network", Electronics Letters, Vol. 28, No. 18, August, 1992, pp. 1766-1768.
- [8] Andrejević, M., and Litovski, V., "ANN Application In Modelling of Chua's Circuit", Proc. of Neurel 2002, Belgrade, Serbia and Montenegro, September 2002, pp. 119-122
- [9] Andrejević, M., and Stojilković, S., "Nonlinear conductance modelling in Josephson junction" (in Serbian), Proceedings of XLVII Conf. of ETRAN, Herceg Novi, Serbia and Montenegro, Jun 2003, pp. 172-175.
- Implementation of ANNs to two- and four- terminal dynamic modeling**
- [10] Zarković, K., Ilić, T., Savić, M., and Litovski, V., "ANN application in Modeling of Dynamic Two-Terminal Linear Circuits", Proceedings of ETAI'2000, September 2000, Ohrid, Macedonia, pp. E-27-E-30.
- [11] Ilić, T., Zarković, K., Litovski, V., and Damper, R., "ANN Application in Modelling of Dynamic Linear Circuits", Proceedings of The Small Systems Simulation Symposium, SSSS2000, Niš, Yugoslavia, Sept. 2000, pp. 43-47.
- [12] Andrejević, M., and Litovski, V., "ANN Application to modelling of reactive nonlinear two-port circuits" (in Serbian), Tehnika (Elektrotehnika), Vol. 51, No. 4-5, May 2002, pp. E1-E11.
- [13] Andrejević, M., and Litovski, V., "Non-Linear Dynamic Network Modelling Using Neural Networks", Int. Congress on Computational and Applied Mathematics, Leuven, Belgium, 22. July-26. July, 2002, pp. 16.
- [14] Litovski, V., Mrčarica, Ž., and Ilić, T., "Simulation of Non-Linear Magnetic Circuits Modelled Using Artificial Neural Network", Journal Simulation Practice and Theory, Vol. 5, 1997, pp. 553-570.
- [15] Litovski, V., and Andrejević Stošović, M., "Nonlinear modelling in biomedical applications using ANNs", Proc. of the International Conference on Biomedical Electronics and Devices, Biodevices 2008, 28. - 31. January 2008., Funchal, Madeira, Portugal, Vol. 1, pp.115-116.
- [16] Andrejević, M., and Litovski, V., "Electronic Modelling Using ANNs For Analogue and Mixed-Mode Behavioural Simulation", Journal of Automatic Control, University of Belgrade, Serbia and Montenegro, Vol. 13, No. 1, 2003, pp. 31-37.
- [17] Litovski, V., Andrejević, M., "ANN Application in Modelling of A/D Interfaces for Mixed-Mode Behavioral Simulation", Proc. of the XLVI Conf. of ETRAN, ETRAN 2002, June 2002, Banja Vrućica, Bosnia and Herzegovina, pp. I.51-I.54.
- [18] Litovski V., Andrejević M., and Damper. R., "Modeling the D/A interface for mixed-mode behavioral simulation", EUROCON 2003, Ljubljana, Slovenia, Sept. 2003, pp. A130-A133.
- [19] Litovski, V., Andrejević, M., Petković, P., and Damper, R., "ANN Application to Modelling of the D/A and A/D Interface for Mixed-Mode Behavioural Simulation", Journal of Circuits, Systems and Computers, Vol. 13, No. 1, February 2004, pp. 181-192.
- Use of ANNs for test pattern evaluation of electronic systems**
- [20] Andrejević, M., Litovski, V., and Zwolinski, M., "Black-Box Application in Modeling of Micro-electro-mechanical Systems", Electronics, Vol. 8, No. 2, December 2004, pp. 27-30.
- [21] Litovski, V., Andrejević, M., and Zwolinski, M., "Acceleration of MEMS fault simulation using ANNs", Electronics, Vol. 8, No. 2, December 2004, pp. 49-53.
- [22] Litovski V., Andrejević, M., and Zwolinski M., "Behavioural Modelling, Simulation, Test and Diagnosis of MEMS using ANNs", 2005 IEEE International Symposium on Circuits and Systems, Kobe, Japan, 2005, pp. 5182-5185.
- Diagnosis of mixed signal systems based on ANN implementation**
- [23] Litovski, V., and Andrejević, M., "ANN application in electronic circuits diagnosis", Proc. of the XLVIII ETRAN Conference, Čačak, Serbia and Montenegro, June 6-10, 2004, Vol. 1, pp. 21-24.
- [24] Andrejević, M., and Litovski, V., "ANN application in electronic diagnosis - Preliminary results", Proc. of MIEL 2004, Niš, Serbia and Montenegro, 2004, Vol. 2, pp. 597-600.
- [25] Litovski, V., Andrejević, M., and Zwolinski, M., "ANN based modeling, testing, and diagnosis of MEMS", Proc. of the 7th Seminar on Neural network Applications in Electronic engineering, NEUREL 2004, Sept. 2004, Belgrade, Serbia and Montenegro, pp. 183-188.
- [26] Litovski, V., Andrejević, M., and Zwolinski, M., "Analog electronic circuit diagnosis based on ANNs", Microelectronics Reliability, Vol. 46 (2006), pp. 1382-1391.
- [27] Andrejević, M., Petrović, V., Mirković, D., and Litovski, V. B., "Delay defects diagnosis using ANNs", Proc. of the L Conference of ETRAN, Belgrade, Serbia, June 2006, pp. EL, 1, 27-30.
- [28] Andrejević, M., and Litovski, V., "Fault diagnosis in analog part of mixed-mode circuits", Proc. of the 6th Symposium on Industrial Electronics, INDEL 2006, Banja Luka, Bosnia and Herzegovina, November 2006, pp. 117-120.
- [29] Andrejević, M., Litovski, V., and Zwolinski, M., "Fault Diagnosis in Digital Part of Mixed-Mode Circuits", Proc. of the 25th Int. Conf. On Microelectronics (MIEL2006),

- Belgrade, Serbia and Montenegro, 14-17 May, 2006, pp. 437-440.
- [30] Andrejević, M., and Litovski, V., "Fault diagnosis in digital part of sigma-delta converter", Proceedings of the 8th Seminar on Neural network application in electrical engineering, NEUREL '2006, Belgrade, Serbia and Montenegro, pp. 177-180.
- [31] Andrejević Stošović, M., and Litovski V., "Hierarchical Approach to Diagnosis using ANNs", Proc. of IEEE 25th International Conference on Microelectronics (MIEL2008), Niš, Serbia, May 2008, Vol. 2 pp. 395-398
- Prediction**
- [32] Milojković, J, and Litovski, V.B., „Dynamic Short-Term Forecasting Of Electricity Load Using Feed-Forward ANNs“, International Journal of Engineering Intelligent Systems for Electrical Engineering and Telecommunication, ISSN 1472-8915, accepted.
- [33] Milojković, J. and Litovski, V., "Comparison of Some ANN Based Forecasting Methods Implemented on Short Time Series", 9th Symposium on Neural Network Applications in Electrical Engineering, NEUREL-2008, Belgrade, 2008, pp. 179-179.
- [34] Milojković, J., and Litovski, V.B., "Prediction in Electronics based on limited information", Proceedings of the 8th WSEAS Int. Conf. on Electronics, Hardware, wireless and optical communications, EHAC'09, Cambridge, UK, February 2009, pp. 33-38.
- [35] Milojković, J., and Litovski, V.B., "Short-Term Forecasting Of The Electricity Load At Suburban Level", IX National Conference with International Participation, ETAI 2009, Ohrid, Macedonia, Septembar 2009, prihvaćeno za objavljivanje.
- [36] Milojković, J., Litovski, V., "Short-Term Forecasting Of Electricity Load Using Recurrent Anns", 15th International Symposium on Power Electronics – Ee2009, Novi Sad, Serbia, October, 2009. Paper No. T1-1.7, pp. 1-5.
- [37] Milojković, B., i Litovski, V. B., «New methods of prediction for sustainable development applications », Proc. of the LI Conf. Of ETRAN, Herceg Novi, June, 2007, paper No. EL1.8. (in Serbian)
- [38] Milojković, J. i Litovski, V.B., „Methods of prediction for ecological needs“, Proc. of Ekoist'08, Ecological trouth, June 2008, Soko Banja, pp. 543-548.
- [39] Milojković, J., Litovski, V., "One step ahead prediction in electronics based on limited information", Proc. of the LIII Conf. of ETRAN, June 2009, Vrnjačka Banja, pp. EL 1.7-1-4.
- Miscellaneous**
- [40] Rađenović-Mrčarica, J., Mrčarica, Ž., Detter, H., and Litovski, V., "Neural Network Visual Recognition For Automation Of The Microelectromechanical Systems Assembly", International Journal on Neural Systems, Vol. 8, No. 1, Feb., 1997, pp. 69-79.
- [41] Rađenović-Mrčarica, J., Mrčarica, Ž., Litovski, V., and Detter, H., "Application Of Neural Networks In Microsystem Assembly", 4th Seminar on Neural Network Applications in Electrical Engineering, NEUREL '97, Belgrade, Sept., 1997, pp. 157-160.
- [42] Rađenović-Mrčarica, J., Mrčarica, Ž., Detter, H., Brenner, W., and Litovski, V., "Neural Network Visual Recognition Applied To Microelectromechanical Parts Assembly", Int. Conf. on Engineering Applications of Neural Networks, EANN '96, Kingston upon Thames, 1996, pp. 325-328.
- [43] Ranđelović, Z., and Litovski, V., "An Application Of Hopfield-Tank's Neural Net On Channel Routing", 15-16 International Annual School on Semiconductor and Hybrid Technologies, 1992-93, Sozopol, Bulgaria, May, 1994, pp. 106-111.
- [44] Milenković, S., Litovski, V., and Obradović, Z., "Nondeterminism in Artificial Neural Networks", International Memorial Conference D. S. Mitrinović, Niš, Yugoslavia, 20-22 Jun, 1996.
- [45] Milenković, S., Obradović, Z., and Litovski, V., "Annealing Based Dynamic Learning in Second-Order Neural Networks", International Conference on Neural Networks, ICNN '96, Washington, D.C., USA, 3.-6. June, 1996, pp. 458-463.
- [46] Zarković, K., Litovski, V., and Stojilković, S., "Acceleration of artificial neural networks learning using statistical methods" (in Serbian), Proc. of the XXXVIII Conf. Of ETRAN, Vol. III, Niš, Yugoslavia, June 1994, pp. 205-206.
- [47] Milenković, S., Risojević, V., and Litovski, V., "Noise Based Gradient Descent Learning", Proc. of 4th Seminar on Neural Networks Appl. in Electrical Engineering, NEUREL '97, Belgrade, Sept., 1997, pp. 28-33.
- [48] Milenković, S., Obradović, Z., and Litovski, V., "Dynamic learning of second-order neural networks based on simulated annealing" (in Serbian), 3rd Seminar on Neural Network Applications in Electrotechnics NEUREL-95, Belgrade, Sept., 1995.

Multistep forecasting in electronics based on reduced information

Jelena Milojković and Vančo Litovski

Abstract – New results are reported related to the extension of our prediction methods to the case of multistep forecasting using Artificial Neural Networks (ANNs) based on reduced information. Examples will be given related to prediction of quantities of obsolete computers.

Keywords – Artificial neural networks, prediction, time series, obsolete computers.

I. INTRODUCTION

Prediction of short time series is a topical problem [1]. Cases where the sample length N is too small for generating statistically reliable variants of prediction are encountered every so often. This is a characteristic of many applied problems of prediction in technology development, marketing, politology, investment planning, and other fields. According to statistical analysis, in order to take into account all components, the prediction base period should contain at least several hundreds of units. For periods of several tens of units, satisfactory predictions can be constructed only for the time series representable as the sum of the trend, seasonal, and random components. These models, however, must have a very limited number of parameters. Series made up by the sum of the trend and the random component sometimes may be predicted for even a smaller base period. Finally, as stated in [1] for a prediction base period smaller than some calculated value N_{\min} , a more or less satisfactory prediction on the basis of observations is impossible at all, and additional data are required.

All that is valid for the more difficult problem: multistep ahead prediction. Namely, as the interval in future, for which prediction is made, becomes comparable with the prediction base period, prediction does not make sense no matter how long both series are. Consequently, if the prediction base period is short, the look in future must be limited.

Among the fields not mentioned in [1], dealing with really small set of data or "prediction base period", we will comment here is the environmental impact of electronics which became an important issue nowadays [2]. As a

Jelena Milojković, Vančo Litovski are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: jelena@venus.elfak.ni.ac.rs, and vanco.litovski@elfak.ni.ac.rs.

matter of fact, the eco-design of electrical and electronic products is already a legislative matter [3],[4]). Electronic waste (EW) is considered hazardous while, in the same time, in enormous quantities. Prediction in this area is of paramount importance for planning and installing equipments, plants, and facilities for recycling and end-of-life management of electronic products, while short term data are available only.

In a set of recent studies [5], [6], [7], [8] dealing with the quantities of EW, attempts were made to make prediction based on hunches and rules of thumb. In fact, some presumptions were made and predictions based on them published. Later, having missed the target, the presumptions were corrected, and so on. On the other side, there is a large number of publications dealing with prediction of time series as such, and with prediction related to environmental data based on artificial neural networks (ANNs), to mention only [9].

Having all that in mind we undertook a project of developing an ANN based method that will be convenient for systematic implementation in stationary time series prediction with reduced set of data. Our first results were published in [10], [11], [12]. The main idea implemented was the following. If one wants to create neural network that may be used for forecasting one should enable this property during ANN's training. In addition, the ANN used has to have such a structure to accommodate to the training process for prediction. Following these considerations new forecasting architectures were developed.

The goal of this paper is to present extensions of the methods already published with the goal to implement them in multistep prediction.

The structure of the paper is as follows. After general definitions and statement of the problem we will give a short background related to ANNs application to forecasting. Then we will describe two solutions for possible applications of ANNs aimed to the same forecasting task. Methods of application of these structures and extensions of the ANN structures will be proposed next, aimed to multistep ahead prediction. Finally short discussion of the results and consideration related to future work will be given.

II. PROBLEM FORMULATION AND SOLUTION

A time series is a number of observations that are taken

consecutively in time. A time series that can be predicted precisely is called deterministic. A time series that has future elements which can be partly determined using previous values, while the exact values cannot be predicted, is said to be stochastic [13]. The stochastic models provide the forecast as the expectation of the identified stochastic process. They allow calculations on statistical properties of the forecasting error (which of course rely on the assumptions made on the model). The deterministic models, on the other hand, provide only the forecast values, not a measure for the forecasting error [14].

We are here addressing deterministic type of time series, only. It is our task to find a functional expression that captures the complex interwoven deterministic relationships that exists between the phenomenon under consideration and the independent variables.

Consider a scalar time series denoted by y^i , $i=1,2, \dots, m-1$. It represents a set of observables of an unknown function $Y=F(t)$, taken at equidistant time instants separated by the interval Δt i.e. $t^{i+1} = t^i + \Delta t$. One step ahead forecasting means to find such a function f that will perform the mapping

$$y^m = f(t^m) = Y^m + \varepsilon \quad (1)$$

where Y^m is the desired response, with an acceptable error ε .

The prediction of a time series is synonymous with modelling of the underlying physical process responsible for its generation [15]. This is the reason of the difficulty of the task. There have been many attempts to find solution to the problem. Among the classical deterministic methods we may mention the k -nearest-neighbour [16], in which the data series is searched for situations similar to the current one each time a forecast needs to be made. This method asks for kind of periodicity to function that is not the case in the situation considered in our proceedings.

In the past decades ANNs have emerged as a technology with a great promise for identifying and modelling data patterns that are not easily discernible by traditional methods. A comprehensive review of ANN use in forecasting may be found in [17]. Among the many successful implementations we may mention [18] [19] [20]. A common feature, however, of the existing application is that they ask for a relatively long time series to become effective. Typically it should be not shorter than 50 data points [17]. This is due to the fact that they all look for periodicity within the data what can be easily seen from the typical forecasting competition data [19]. Very short time series were treated [20], [21]. Here additional "nonsample information" was added to the time series in order to get statistical estimation from deterministic data.

That is why we went for a search for topological structures of ANN that promise prediction based on short time series. In the next, we will first briefly introduce the feed-forward neural networks that will be used as a basic structure for prediction throughout this paper.

The network is depicted in (Fig.1). It has only one hidden layer, which has been proven sufficient for this kind of problem [22]. Indices: "in", "h", and "o", in this figure, stand for input, hidden, and output, respectively. For the set of weights, $w(k, l)$, connecting the input and the hidden layer we have: $k=1,2,\dots, m_{in}$, $l=1,2,\dots, m_h$, while for the set connecting the hidden and output layer we have: $k=1,2, \dots, m_h$, $l=1,2,\dots, m_o$. The thresholds are here denoted as $\theta_{x,r}$, $r=1,2, \dots, m_h$ or m_o , with x standing for "h" or "o", depending on the layer. The neurons in the input layer are simply distributing the signals, while those in the hidden layer are activated by a sigmoidal (logistic) function. Finally, the neurons in the output layer are activated by a linear function.

Creation of a feed forward ANN that performs a given task consists of several steps. First, one should decide on the number of inputs and outputs, m_{in} and m_o , respectively. That usually comes with the nature of the problem under consideration.

Next, according to the input-output structure the training data are to be organized. Pairs of input-output vectors are taken from the known data and a list is created intended to be presented to the ANN during training. Generally speaking, part of the input data is kept for validation of the training process but when prediction is considered there is no such data. We simply look to the unknown future. Verification of the prediction may be done only after time passed. Here the importance of the dependability onto the whole prediction algorithm and software comes into fore. The algorithm should be organized in that way to perform automatically and give no chance to mistakes.

The internal structure of the network i.e. the number of hidden neurons (m_h) is of paramount importance for successful prediction. It defines the number of free parameters that are available for optimization (training). Of course, one would prefer as simple the ANN as possible. That not only makes the solution faster to run but also facilitates the training process in: choice of the initial values of the parameters, reaching convergence, and speeding up the training. To get the value of m_h we applied a procedure that is based on proceedings given in [23].

We solve the initial value problem for the weights and thresholds by creating small random numbers with uniform distribution such that $v_k \in (-\alpha, \alpha)$, where v_k stands for the k th parameter (weight or threshold), while α is a properly chosen small number.

When considering the training algorithm one must have in mind that it represents a search of the parameters space for the global minimum of the predefined error function. Any learning algorithm, being it in nature Newthton-Raphson, steepest descent, back propagation, annealing based (Metropolis), or genetic, which, for a given procedure creating initial solution, leads repetitively and systematically to a global minimum, is good enough for application. Of course, some of the algorithms will lead to the solution with different velocity (expressed in number of iterations

and elapsed time) but that is of secondary importance. Among the successful algorithms one chooses the one that is fastest, simplest to program, needs less computer memory, etc. The learning algorithm we used for training in these proceedings is a version of the steepest-descent minimization algorithm [24]. It is our experience (almost twenty years, now) that for the problem under consideration it performs the best.

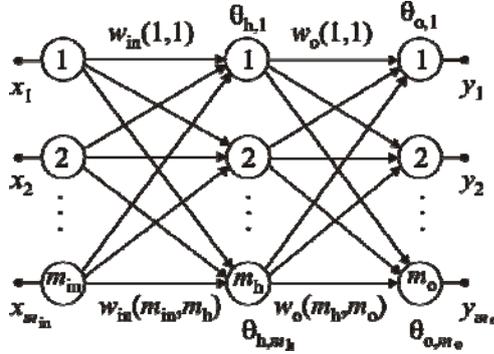


Figure 1. Fully connected feed-forward neural network with one hidden layer and multiple outputs

In prediction of time series, in our case, a set of observables (samples) is given (per year) meaning that only one input signal is available, the discretized time. We are predicting one quantity at a time meaning one output is needed, too. The values of the output are numbers (millions of pieces or weight of obsolete computer units). To make the forecasting problem numerically feasible we performed transformation in both the time variable and the response. The time was reduced by t_0-1 so that

$$t=t^*-(t_0-1). \tag{2}$$

Having in mind that t^* stands for the year, this reduction gives the value of 1 to the year (t_0) related to the first sample. The samples are normalized in the following way

$$y=y^*/M \tag{3}$$

where y^* stands for the current value of the target function, M is a constant which will be chosen according to the problem at hand (for example, $M=10^6$ cubic feet for the volume of obsolete computers).

If the architecture depicted in (Fig. 1) was to be implemented the following series would be learned: $(t^i, f(t^i))$, $i=1, \dots, m-1$. $m-1$ is here the number of samples available i.e. the number of observables.

Starting with the basic architecture of (Fig. 1.), the possible solutions were investigated in [10] [11] and two new architectures were suggested to be the most convenient for the solution of the forecasting problem based on short prediction base period.

The first one, named *time controlled recurrent* (TCR) was inspired by the time delayed recurrent ANN [15]. It is a recurrent and time delayed architecture but, in the same time, insists on the time variable to control the predicted

value as depicted in (Figure 2). Our intention was to benefit from both: the generalization property of the ANNs and the success of the recurrent architecture. Here in fact, the network is learning a set in which the output value is controlled by its own previous instances and the present time. The version of this network, intended to be implemented for one-step-ahead prediction may be analytically expressed as:

$$y^{i+1} = f(t^i, y^i, y^{i-1}, \dots, y^{i-q}), \quad i=q+1, \dots, m-2, \tag{5a}$$

where q stands for the number of previous values of the function used for training. $q>0$ and, obviously, $q+1<m-1$. After training, the predicted value in the first next step is obtained as

$$y^m = f(t^{m-1}, y^m, y^{m-1}, \dots, y^{m-q}). \tag{5b}$$

Note that the learning procedure here was implemented exactly in the same way as in [25].

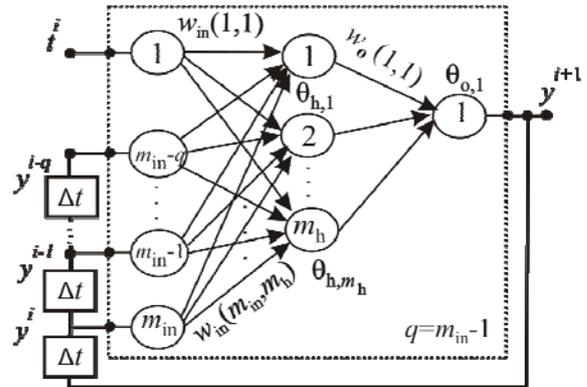


Figure 2. TCR. Time controlled recurrent ANN

The second architecture is named *feed forward accommodated for prediction* (FFAP) and depicted in (Fig. 3). Our idea was here to force the neural network to simultaneously learn the same mapping several times but shifted in time. In that way, we presume, the previous responses of the function will have larger influence on the $f(t)$ mapping.

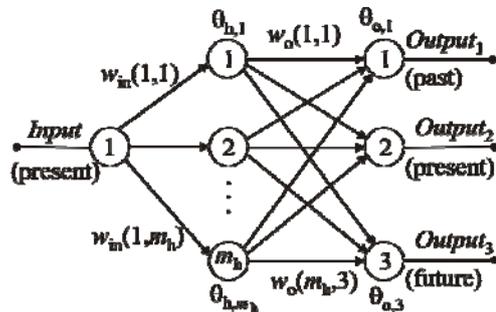


Figure 3. FFAP. Feed forward ANN structure accommodated for prediction

There is one input terminal that, in our case, is t^i . The $Output_3$ terminal, or the future terminal, in our case, is to be forced to approximate y^{i+1} . $Output_2$ should learn the present value i.e. y^i . Finally, $Output_1$ should learn the past value i.e. y^{i-1} . Again, if one wants to control the mapping by a set of previous values, $Output_1$ may be seen as a vector such as $\{y^{i-1}, y^{i-2}, \dots, y^{i-q}\}$. We may express the functionality of the network, for the case of one-step-ahead prediction, as

$$\{y^{i+1}, y^i, y^{i-1}, \dots, y^{i-q}\} = \mathbf{f}(t^i), \quad i=q+1, \dots, m-2, \quad (6a)$$

where $Output_1 = \{y^{i-1}, \dots, y^{i-q}\}$, meaning that: one future, one present and q previous responses are to be learned. After training the predicted and the approximated values of the output are obtained by running the ANN as:

$$\{y^{m+1}, y^m, y^{m-1}, \dots, y^{m-q}\} = \mathbf{f}(t^m). \quad (6b)$$

The presumption of the mutual interrelation between the output responses of the FFAP network comes from the fact that they all depend on the parameters (weights and thresholds) of the hidden neurons. By adjusting the parameters to learn y^i , for example, one simultaneously changes the y^{i+1} response, and vice versa. In that way, during training, the values of the response from previous time instants indirectly control the prediction.

III. IMPLEMENTATION EXAMPLE

An examples will be given here demonstrating the properties of the solutions proposed with $q=2$.

We will consider the prediction of the quantities of obsolete computers in the USA based on data given in [5]. According to [5], putting $t_0=1991$, after normalization, we get (Table 1) as the set of observables representing the quantities of obsolete computers in the USA. Here $M=10^6$ cubic feet. The same data are visualized in (Fig. 4). It may be seen that the function that governs the phenomenon is not monotonic giving rise to the difficulty of prediction. If, for example, periodicity is to be exploited in this example (what would be done if the k -nearest neighbor method was implemented) then $f(9)$ would be less than 14, since after three points of positive increments (as for the interval $\{1,3\}$) comes a negative one. $f(4) < f(3)$ would lead to $f(9) < f(8)$ which is not the case.

The first eight samples will be used as training data while the last one i.e. $t=9$ and $f(t)=18.4$, will be compared with the predictions obtained, in order to validate the method.

In the following, two experiments will be described based ANN architectures emanated from (Figure 2) and (Figure 3).

The results obtained after learning are expressed in (Table 2). It contains information on both the structure of

the networks and the values obtained by prediction.

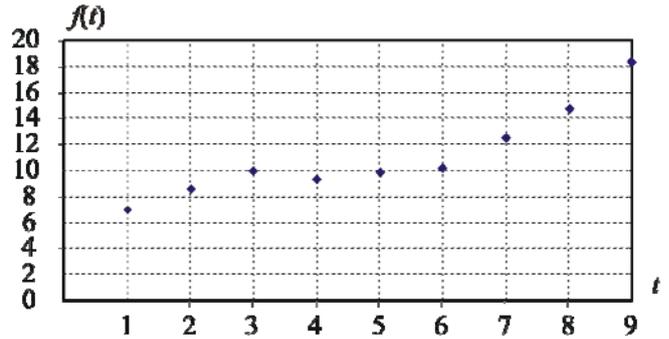


Figure 4. Visualization of Table 1

TABLE 1.
QUANTITIES OF OBSOLETE COMPUTERS IN TIME

| | | | | | | | | | |
|--------|------|------|------|------|------|-------|-------|-------|------|
| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $f(t)$ | 7.03 | 8.67 | 10.0 | 9.33 | 9.85 | 10.18 | 12.54 | 14.76 | 18.4 |

TABLE 2.
PREDICTION OF QUANTITIES OF OBSOLETE COMPUTERS. NOTE:
 $F(9)=18.4$.

| Solution type | No. of hidden neurons | No. of output neurons | $f(9)$ | Error % |
|---------------|-----------------------|-----------------------|---------|---------|
| TCR | 10 | 1 | 17.2114 | 6.46 |
| FFAP | 4 | 4 | 18.2274 | 0.93 |

By examining the results depicted in (Table 2) we may conclude that satisfactory prediction was obtained with both architectures. Nevertheless, it is to be mentioned that the FFAP is considerably nearer to the solution needed. What is not expressed in the table is the fact that the FFAP solution is much more sensitive to the initial solution for the weights and thresholds, making the training process more difficult and uncertain.

It is not shown here, for the sake of simplicity, but it is worth mentioning that both TCR and FFAP approximate excellent. That means that except for the $[9, f(9)]$ point, all previous points on the curve $f(t)$ overlap exactly with the ones depicted in (Figure 4). One should not substitute “approximation” with “prediction”, however. Namely, approximation is achieved *within* a given interval. Here, for this example, that is $t \in \{1,8\}$. The ability of the ANN to successfully calculate the values of the function for any value of the independent variable within that interval is referred to as *generalization*. In our case we are looking for the *extrapolation* i.e. the value of the function outside of the given interval. That is what we consider forecasting or prediction.

There is no recommendation as to which of these solutions is to be accepted or discarded. Namely, prediction is a search in the dark and one always needs some reference for the solution offered. Here, since the FFAP solution offers better results, one should keep the TCR solution as a confirmation that the FFAP is not a complete miss what is, of course, possible since the training of an ANN is iterative process that may be stuck in local minimum.

IV. IMPLEMENTATION TO MULTI-STEP PREDICTION

The main goal of our research was to develop a method for one-step-ahead prediction based on reduced set of data. Implementation to long term prediction was always a temptation while we are aware that it is difficult to believe that one may predict for a period in future as long as the prediction base period is. Instead, here we will give the results of an attempt to apply our method to prediction for a somewhat longer period than one-step-ahead.

There are, in our opinion, two ways of how our method may be applied for longer term prediction. First, one may use the predicted results for the time instant t^{i+1} , namely y^{i+1} , and to concatenate the input set with them. Now, the prediction may start for t^{i+2} as if one has longer prediction base period. This may be repeated as long as wanted. The problem with this idea is related to the fact that the error in prediction contained in y^{i+1} will be accumulated in the next prediction, and so on. At the end, one may have no confidence in the final long term prediction. Example of implementation of this idea to the problem of forecasting quantities of obsolete computers is given in Table 3. Both TCR and FFAP ANNs were implemented to get prediction for $t=9$, based on samples for $t=1,2,3,\dots,7$. The idea is to predict two intervals ahead. The value of y for $t=8$ was predicted first. Then, it was used as if it was part of the input file to predict $y(9)$. We can see that the results are worse than the previous ones with the ones obtained with FFAP ANN being absolutely deteriorated.

TABLE 3. TWO-STEP-AHEAD PREDICTION BY CONCATENATION

| | Actual | Predicted | Error in % |
|------|--------|-----------|------------|
| TCR | 18.4 | 16.8616 | 8.36 |
| FFAP | 18.4 | 26.2071 | -66.3 |

Alternatively, one may predict two (or more) steps ahead directly by skipping the intermediate intervals. In such a case, for the TCR ANN, for instance, one would perform the following

$$(7a) \quad y^{i+k} = f(t^i, y^i, y^{i-1}, \dots, y^{i-q}), \quad i=q+1, \dots, m-1-k,$$

while for the FFAP case we have

$$(7b) \quad \{y^{i+k}, y^i, y^{i-1}, \dots, y^{i-q}\} = \mathbf{f}(t^i), \quad i=q+1, \dots, m-1-k.$$

In this expressions k stands for the number of intervals in future after the prediction base period.

Looking to them we find that for the one-step-ahead prediction ($k=1$) we had $m-1$ samples to be used for training and $b=m-2-q$ "training lessons". On the other side, for multistep prediction, the number of training lessons may be stated as $b=m-1-k-q$, as depicted in Fig. 5 for $q=2$ and $k=2$. If the number of intervals in future, k , rises, b is diminished. It is equivalent to reduction of the prediction base period what should lead to reduction of the quality of the forecast.

| | $q=2$ | | $b=4$ | | | | $k=2$ | | |
|--------|-------|------|-------|------|------|-------|-------|-------|------|
| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $f(t)$ | 7.03 | 8.67 | 10.0 | 9.33 | 9.85 | 10.18 | 12.54 | 14.76 | 18.4 |

Figure 5. The reduction of the number of training lessons in multistep ahead prediction

This method of prediction was checked by an experiment related to prediction of the number of obsolete computers as above. Again both TCR and FFAP ANNs were implemented. The forecasting results are given in Table 4. Comparing these result with the ones depicted in Table 2, we may conclude that the expected deterioration does not becomes apparent at once (the number of training lessons was reduced by 1 only). In the TCR case we got even an improvement. That makes this approach promising in general and especially in cases when a bit larger prediction base period is available.

TABLE 4. TWO-STEP-AHEAD PREDICTION WITH SKIPPING

| | Actual | Predicted | Error in % |
|------|--------|-----------|------------|
| TCR | 18.4 | 18.7458 | -1.879 |
| FFAP | 18.4 | 17.5698 | 4.512 |

Finally, for the FFAP ANN only, in cases where multiple-step prediction is planned $Output_3$ may be seen as a vector. In this situation, referring to Fig. 3, $Output_1$ has q terminals, $Output_2$ has 1 terminal, and $Output_3$ has k terminals. Analytically, this may be expressed as

$$(8) \quad \{y^{i+k}, y^{i+k-1}, \dots, y^i, y^{i-1}, \dots, y^{i-q}\} = \mathbf{f}(t^i), \quad i=q+1, \dots, m-1-k.$$

The appropriate structure of the FFAP ANN is depicted in Fig. 6 for $q=2$ and $k=2$. It is important to notice that no skipping is present now. The network is presented by all the future values of the function.

Applying this method to the case of obsolete computers with two-steps-ahead prediction, with $q=2$ and $k=2$, produced a solution of $f(9)=18.5106$ what is a miss of only 0,6%. It is an excellent result comparing with the results presented here earlier. The number of training lessons was now reduced to $t \in \{3,6\}$.

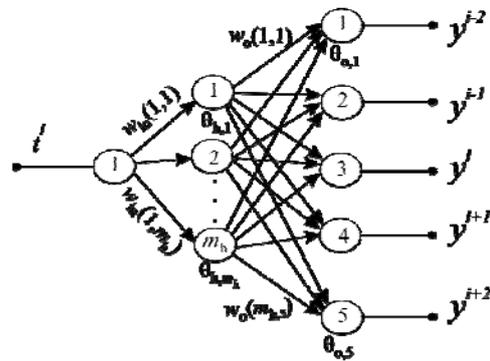


Figure 6. FFAP structure for two step ahead prediction without skipping

TABLE 5.
DATA STRUCTURE FOR THE FFAP NETWORK
PREDICTING TWO-STEPS-AHEAD WITHOUT SKIPPING.

| | i | y^{i-2} | y^{i-1} | y^i | y^{i+1} | y^{i+2} |
|--------------------|-----|-----------|-----------|-------|-----------|-----------|
| Training | 3 | 7.03 | 8.67 | 10.0 | 9.33 | 9.85 |
| | 4 | 8.67 | 10.0 | 9.33 | 9.85 | 10.18 |
| | 5 | 10.0 | 9.33 | 9.85 | 10.18 | 12.54 |
| | 6 | 9.33 | 9.85 | 10.18 | 12.54 | 14.76 |
| 7 Prediction | | 6.447 | 11.78 | 10.59 | 18.40 | 18.51 |
| Expected values | | 9.85 | 10.18 | 12.54 | 14.76 | 18.4 |

The data structure for training and running the FFAP network predicting two-steps-ahead without skipping is given in Table 5. Here, for convenience, in the row $i=7$, the responses of all outputs of the network are presented while only the last one is usable for prediction. The first four outputs are trained to approximate. To go further, in the last row, the expected values for every output are listed. Comparing the last two rows of Table 5 one easily concludes that no output except the one intended to, is predicting successfully. That is in accordance with our previous results discussed in [11]: “No ANN trained for interpolation can predict (extrapolate) successfully”.

Finally, the response of the predicting output (y^5) as a function of time together with the target values is depicted in Fig. 7. One may see that this response not only extrapolates but interpolates excellent as well.

To make the results completely reproducible, Table 6 contains the initial values of the synaptic weights and

thresholds used for the training process of the FFAP network predicting two-steps-ahead without skipping. Table 7 contains the final values obtained after training and used to get the prediction.

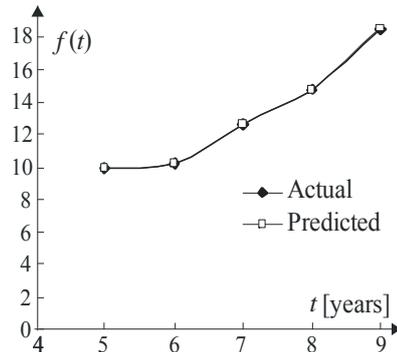


Figure 7. Response of the FFAP structure for two-steps- ahead prediction without skipping

TABLE 6.
INITIAL VALUES OF THE SYNAPTIC WEIGHTS AND THRESHOLDS FOR
THE FFAP NETWORK PREDICTING TWO- STEPS-AHEAD WITHOUT
SKIPPING

| j | $w_{in}(1, j)$ | $\theta_{h,j}$ | $w_o(j,1)$ | $w_o(j,2)$ | $w_o(j,3)$ | $w_o(j,4)$ | $w_o(j,5)$ | $\theta_{o,j}$ |
|-----|----------------|----------------|------------|------------|------------|------------|------------|----------------|
| 1 | -.122 | -.102 | -.132 | -.132 | -.32 | -.32 | -.132 | .135 |
| 2 | .131 | .211 | .1243 | .131 | .124 | .124 | .240 | -.21 |
| 3 | -.212 | -.131 | -.214 | -.14 | -.140 | -.140 | -.124 | .123 |
| 4 | .120 | .121 | .124 | .240 | .24 | .324 | .324 | -.141 |
| 5 | | | | | | | | .121 |

TABLE 7.
VALUES OF THE SYNAPTIC WEIGHTS AND THRESHOLDS FOR THE FFAP NETWORK
PREDICTING TWO- STEPS-AHEAD WITHOUT SKIPPING

| j | $w_{in}(1, j)$ | $\theta_{h,j}$ | $w_o(j,1)$ | $w_o(j,2)$ | $w_o(j,3)$ | $w_o(j,4)$ | $w_o(j,5)$ | $\theta_{o,j}$ |
|-----|----------------|----------------|------------|------------|------------|------------|------------|----------------|
| 1 | -4.45698 | -0.584837 | -5.72051 | 6.86385 | -5.77336 | 1.01748 | -4.08986 | 5.81894 |
| 2 | 6.19043 | -6.12091 | -3.14972 | 4.6372 | 1.23713 | 5.53971 | 2.14366 | -17.4828 |
| 3 | -5.39005 | -6.21706 | -2.41941 | -3.80742 | -4.04059 | -7.0992 | -6.93629 | 15.4904 |
| 4 | 42.7411 | 1.85431 | -4.34902 | 17.3764 | -14.6085 | 2.31862 | -5.84657 | -2.36684 |
| 5 | | | | | | | | 6.52768 |

V CONCLUSION

New results were reported related to multistep ahead prediction using ANNs based on reduced information. Several solutions were proposed and experimental results were given for one implementation. In general, encouraging results were obtained for two-step prediction except for the method using concatenation. These results will be a basis for further research and implementation to different contexts such the ones presented in [10], [12], [26] and [27].

REFERENCES

- [1] Mandel', A.S., “Method of Analogs in Prediction of Short Time Series: An Expert-statistical Approach”, Automation and Remote Control, Vol. 65, No. 4, 2004, pp. 634-641.
- [2] Karn, B., and Matthews, H.S., “Nanotechnology: Emerging Challenges for Electronics and the Environment”, IEEE Spectrum, Vol. 44, No. 9, 2007, pp. 54-58.
- [3] -, Directive 2005/32/ec of the European parliament and of the council of 6 July 2005, EN Official Journal of the European Union L 191/29.
- [4] Bandyopadhyay, A. , “A regulatory approach for e-waste management: a cross-national review of current

- practice and policy with an assessment and policy recommendation for the Indian perspective”, International Journal of Environment and Waste Management, Vol. 2, No. 1-2, 2008, pp. 139–186.
- [5] Matthews, H.S., McMichael, F.C., Hendrickson, C.T., and Hart, D., “Disposition and End-of-Life Options for Personal Computers”, Green Design Initiative Technical Report #97-10, Carnegie Mellon University, 1997.
- [6] Feszty, K., Colin, M., and Baird, J., “Assessment of the quantities of waste electrical and electronic equipment (WEEE) in Scotland”, Waste management & research, Vol.21, No. 3, 2003, pp. 207-217.
- [7] Matthews H.S., and Matthews, D., “Computers and the Environment: Understanding and Managing Their Impacts”, in Computers and the Environment, eds. E. Williams and R. Kuher, Dordrecht, Kluwer Academic Publications, 2004.
- [8] Litovski, V., Milojković, J., Petrović, S., Džipković, D., Šimurina, M., and Krstić, B., “Program of establishment of the recycling system of waste electronic of computers”, Belgrade, Serbian Agency for Recycling 2006, (in Serbian).
- [9] Hawkins, T., Hendrickson, C., Higgins, C., Matthews, H. S., and Suh S., “A mixed-unit input-output model for environmental life-cycle assessment and material flow analysis”, Environmental Science & Technology, Vol. 41, No. 3, 2007, pp. 1024-1031.
- [10] Milojković, J., and Litovski, V. B., “New methods of prediction implemented for sustainable development”, in 51th Conference of ETRAN, EL1.8, 2007, (in Serbian).
- [11] Milojković, J., and Litovski, V. B., “Comparison of Some ANN Based Forecasting Methods Implemented on Short Time Series”, in 9th Symposium on Neural Network Applications in Electrical Engineering, NEUREL-2008, pp. 175-178.
- [12] Milojković, J., and Litovski, V.B., „Dynamic Short-Term Forecasting of Electricity Load Using Feed-Forward ANNs“, Engineering Intelligent Systems for Electrical Engineering and Communication, ISSN 1472-8915, accepted for publication
- [13] Hussain, A., (2002), “Physical time-series prediction using second order pipelined recurrent neural network”, in Proc. of the 2002 IEEE Int. Conf. on Artificial Intelligence Systems, 219-223.
- [14] Murto P., “Neural Network Models for Short-Term Load Forecasting”. MS Thesis, Helsinki University of Technology, Finland, 1998.
- [15] Haykin, S., “Neural Networks, A Comprehensive Foundation”, New York, Macmillan College Publishing Company, 1994.
- [16] Plummer, E.A., “Time series forecasting with feed-forward neural networks: guidelines and limitations”, unpublished M.S. Thesis, University of Wyoming, The Graduate School, 2000.
- [17] Zhang, B. G., Patuwo, E., and Hu, M. Y., “Forecasting with artificial neural networks: The state of the art”, International Journal of Forecasting, Vol. 14, No. 1, 1998, pp. 35-62.
- [18] Connor, J., and Douglas Martin, R., “Recurrent neural networks and robust time series prediction”, IEEE Trans. on Neural Networks, Vol. 5, No. 2, 1994, pp. 240-254.
- [19] -, NN3 Competition, <http://www.neural-forecasting-competition.com/>
- [20] Brännäs, K., and Hellström, J., “Forecasting based on Very Small Samples and Additional Non-Sample Information”, Umeå Economic Studies 472, Umeå University, Sweden, 1998.
- [21] Navone, H. D., and Ceccatto, H. A., “Forecasting chaos from small data sets: a comparison of different nonlinear algorithms”, Journal of Physics A: Mathematical and Theoretical, Vol. 28, No. 12, 1995, pp. 3381-3388.
- [22] Masters, T., “Practical Neural Network Recipes in C++”, San Diego, Academic Press, 1993.
- [23] Baum, E.B., and Haussler, D., “What size net gives valid generalization”, Neural Computing (MIT Press), Vol. 1, No.1, 1989, pp. 151-160.
- [24] Zografski, Z., “A novel machine learning algorithm and its use in modeling and simulation of dynamical systems”, in Proc. of 5th Annual European Computer Conference, COMPEURO '91, 1991, pp. 860-864.
- [25] Bernieri, A., D'Apuzzo, M., Sansone, L., and Savastano, M., “A neural network approach for identification and fault diagnosis on dynamic systems”, IEEE Transactions on Instrumentation and Measurements, Vol. 43, No. 6, 1994, pp. 867-873.
- [26] Milojković, J.B., and Litovski, V.B., “Short-term forecasting of electricity load using recurrent ANNs”, 15th Int. Symposium On Power Electronics – Ee2009, Novi Sad, Serbia, October 2009, Paper No. T1-1.7, pp. 1-5.
- [27] Milojković, J. And Litovski, V.B., „Short-term forecasting of the electricity load at suburban level“, Ninth National Conference with International Participation, ETAI 2009, Ohrid, Macedonia, September 2009, Proc. on disc, Paper No. A2-1.

Low power digital design in Integrated Power Meter IC

Borisav Jovanović, Mark Zwolinski, Milunka Damnjanović

Abstract - This paper considers the low power design aspects of the digital signal processing blocks embedded into three-phase Integrated Power Meter IC. Several optimization techniques were used to implement power efficient design. The techniques mainly rely on clock and data gating.

Keywords – Low-Power Integrated Power Meter

I. INTRODUCTION

Modern power meter devices relies on single chip referred to as integrated power meter (IPM). The designed IPM incorporates all the required functional blocks for three-phase metering, including a precision energy measurement front-end consisting of Sigma Delta AD converters, digital filters, signal processing block, embedded microcontroller, real-time clock, LCD driver and programmable multi-purpose inputs/outputs.

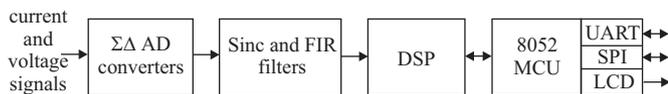


Fig.1 Architecture of the Integrated Power-Meter

The digital filters decimate over-sampled output signals from the on-chip AD converters for both voltage and current signal channels in three phases. The DSP performs the precision computations necessary to measure: active, reactive and apparent energy in four quadrants for all three-phases, instantaneous frequency for each phase, RMS currents and voltages, active, reactive and apparent power and power factor [1].

The microcontroller unit (8052 MCU shown in Fig.1) is compatible with 8052 microprocessors. It includes several communication peripherals: UART, Serial Port Interface (SPI) and LCD driver circuit.

Optimizing power of integrated circuits remains difficult task. This paper considers the low power design aspects of the digital signal processing blocks embedded into three-phase integrated power meter IC.

This paper is organized in five sections and References. The following section gives an overview of power optimization methods applied on DSP block. The third section considers the techniques used for microcontroller's low power optimization. The fourth gives the achieved

Borisav Jovanović and Milunka Damnjanović are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail:(borisav.jovanovic,milunka.damnjanovic)@elfak.ni.ac.rs.

Mark Zwolinski is with School of Electronics and Computer Science, University of Southampton, UK, mz@ecs.soton.ac.uk

power consumption for all digital blocks on chip.

II. LOW POWER TECHNIQUES APPLIED ON DSP BLOCK

A. DSP's operation

DSP block receives from filters (through its 16-bit inputs) the digital samples for voltage, current and phase-shifted voltage, and calculates following results: root mean square values for voltage and current, mean values for active and reactive power, apparent power, active and reactive energy, power factor and frequency, [1,2] The measurement results are obtained for all three power line phases. DSP provides three result sets, one set for each power phase. The measurement range for current signal is from 10 mA RMS to 100 A RMS, and up to 300V RMS for voltage. The values are represented by 24-bit numbers.

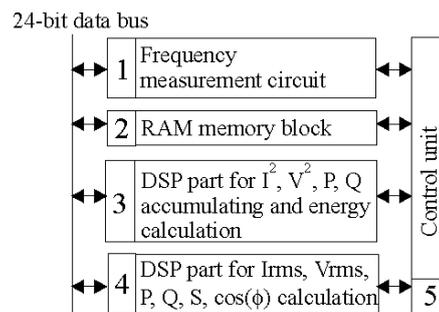


Fig. 2. DSP's block diagram

DSP utilizes controller/datapath architecture and consists of blocks which can be divided into several main groups (Fig. 2):

1. Frequency measurement circuit
2. RAM memory block
3. Part for I^2 , V^2 , P, Q accumulating and energy calculation
4. Part for current and voltage RMS, active, reactive and apparent power and power factor calculation);
5. Control unit that controls all other parts of DSP.

One of power-line parameters provided by DSP, root-mean-square current - I_{rms} , is calculated once per second. Current samples, obtained from digital filters, are multiplied and the current square values are accumulated over the constant time period of one second. After, derived sum is divided by number of samples, and the root-mean-square current is found after square rooting (according to exp.(1)).

$$I_{rms} = \sqrt{\frac{\sum_{n=1}^N i(nT)^2}{N}} \quad (1)$$

The sequence of arithmetical operations for current square summing, performed by Block 3, part of DSP, is shown in Fig.3. The sequence is performed 4096 times per second.

At the sequence beginning, DC offset is removed from instantaneous current values. It is done either by subtracting the constant offset determined during the calibration procedure or by passing the signal through the digital high pass filter. The second doesn't require calibration procedure. After, AC part of instantaneous current is squared in multiplication unit. The value I^2 is passed through the single pole Low Pass Filter (LPF), and after that, it is accumulated into register $AccI^2$ (Fig. 3). All these operations are done by digital circuitry within Block 3. Input (current samples), output (the sum of I^2) and intermediate results (the HPF and LPF registers) are stored outside the Block 3, in one of the three SRAM 64x24 memory blocks. The operations are governed by Control unit - Block 5 in Fig.2.

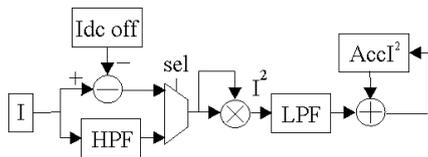


Fig.3. Data processing for current-square accumulation

The same procedure is performed and the same hardware is used for V^2 accumulating. Also, active and reactive power accumulation is done through the same procedure. The only difference is in multiplication process: voltage and current sample-values are used for active power calculation, and current value is multiplied with phase-shifted voltage value to obtain reactive power.

The architecture of Block 3 consists of data registers, arithmetical units for addition and multiplication, and a multiplexer circuit.

After, to generate current root mean square, the intermediate results are passed to Block 4, where, accumulated sum is divided with the constant number 4096 (number of samples). Then, square rooting operation is performed and the result is multiplied with gain correction value, determined during calibration procedure. The same procedure stands for root mean square voltage. The calculation of mean active and reactive power is similar, except there is no rooting. Apparent power is obtained by multiplying root mean square of current and voltage values, and power factor is obtained by dividing active and apparent power values.

Block 4 (Fig. 2) consists of two registers and arithmetical units that implement square rooting, subtraction, multiplication and division. It performs

calculations once after every second in the time period which lasts only 1/4096 seconds. The operation time of Block 3, which performs intensive calculations during the one second period, is 4096 times greater than the one of Block4. The power consumption of Block 4 is, therefore, much lower than the power consumed by Block 3.

The chip is implemented in AMI CMOS 0.35um standard cell technology. This technology does not allow low power optimizations at technology and circuit level. CMOS transistors have only single threshold voltage and cells operate at constant 3.3V power supply. The leakage currents can be neglected comparing to dynamic consumption. The power reduction can be achieved at gate and architectural level through the reducing the clock and data switching activity.

The power dissipation of DSP block can be divided into three main areas. The first area is the power cost associated with accesses to the three data memories (represented by Block 2 in Fig.2). The memories power consists of the power consumed within the RAM units themselves, and the power required to transmit the data across the large capacitance of the 24-bit data bus.

Three 64x24 bit memories supplied by technology manufacturer are located near the functional units to minimize the capacitance of the associated wiring. The number of memory accesses of $8 \cdot 10^5$ gives the power consumption of $150 \mu W$.

The second main area of power consumption comes from the energy dissipated in performing the actual operations on the data. This is made of the energy dissipated by transitions within the datapath and clock tree circuitry. In the DSP block, the most of dissipated power comes from Block 3.

The third area is power consumed by control unit block (Block 5 in Fig.2). The control unit is implemented as finite state machine that controls the operations executed within Blocks 3 and 4. It has more than 500 states and occupies significant part of DSP's area.

Comparing to other blocks, Block 3 is active most of time, performs most of calculations, and, communicates with SRAM memories most frequently. It is extracted from design and examined in detail. The sequence of states in Block 5 which controls the operation within Block 3 is also extracted into new design. The used low power techniques that reduce the switching activity are: clock gating, operand isolation, FSM state decomposition and Gray encoding of FSM states. The application of techniques and obtained results are presented onwards in the paper.

B. Clock gating techniques applied to DSP

Clock power is one of the dominant components of total power consumption. The clock signal is fed to most of the circuit blocks and switches every cycle. The clock tree has large capacitances comparing to other nets and reducing the switching activities of clock signal is important.

Clock gating is the technique for dynamic power reduction [5]. It is based on fact that power is saved by

disabling the clock signal to unused circuits. By AND-ing the clock signal with the some gate control signal, clock gating disables the clock to a circuit, avoiding the unnecessary charging and discharging of net capacitances.

The datapath of DSP incorporates several sequential circuits which are not all the time active. For example, arithmetical units for multiplying, dividing and square rooting in DSP are realized as sequential circuits and they have large inactive periods. The unit for multiplication in Block 3 multiplies two operands in 18 clock periods. It is used during chip normal operation four times inside the interval of 256 clock cycles. Therefore, the multiplication unit is inactive during 70% of chip operation time. In Block 4 similar arithmetical units exists: for multiplication, square rooting and dividing. Since those arithmetical blocks are not used all the time, their clock trees can be gated. Only when arithmetical units are active, their clock signals are enabled.

To avoid glitches in clock signal, 2-input AND cell with D latch is used as a gate. The level sensitive D latch holds the input enable signal from the rising edge until the falling edge of the clock. Since the latch captures the state of the enable signal and holds it until the complete clock pulse has been generated, the enable signal needs to be stable around the rising edge of the clock. The signal at the AND cell output is free of glitches and is used as a clock signal of subsequent sequential circuits.

The architecture of Block 3 consists of two 48-bit data registers, arithmetical units for addition and multiplication and a multiplexer circuit. The control unit generates signals for starting the multiplication, selection one of multiplexer's inputs, and, both memory and register data transfer operations. Considering the non-optimized design, the total clock power is a substantial 32% of the circuit's power. The power of non-optimized design is 1104 μ W and the power consumed by clock tree is 354 μ W. To reduce the clock power, first, the multiplication unit was gated. The design was further power optimized in the way that gating signals are used to write data into registers and memory blocks. The power consumption and area of non-optimized and power optimized design of Block 3 are given in Table I. The power dissipation is improved for 27%. The occupied area remained almost the same as before optimization.

TABLE I

| Block | Original design | | Optimized by clock gating | |
|----------------------|-----------------|------------------|---------------------------|------------------|
| | Area [gates] | Power [μ W] | Area [gates] | Power [μ W] |
| Clock tree | 8 | 354 | 2 | 89 |
| Registers | 456 | 40 | 456 | 32 |
| Three-state circuits | 615 | 106 | 615 | 112 |
| Adder | 320 | 143 | 320 | 129 |
| Multiplexer | 307 | 48 | 307 | 44 |
| Multiplier | 663 | 105 | 663 | 96 |
| FSM circuit | 990 | 308 | 990 | 303 |
| Total | 3359 | 1104 | 3353 | 805 |

C. Operand isolation low power technique applied to DSP

Operand isolation or data gating reduces power consumption by selectively blocking the unused switching activity caused by redundant propagation of data signals through combinatorial circuits. Data gating is added to high-fanout paths - data buses in the datapath. The bus implementation is usually made of three-state cells. Else, the gating in the datapath main sub-blocks consists of AND gates that stop the propagation of signal to the inputs of unused adders and subtraction circuits.

The multiplexer circuit in Block 3 incorporates multiple parallel data paths. By adding the gating at the multiplexer inputs, the power can be saved. Finally, three-state buffers were used instead the multiplexer. The 3-8 decoder circuit provides individual enable signals for three-state buffer array. The transparent latch placed in front of decoder is clocked only if its select output is going to change.

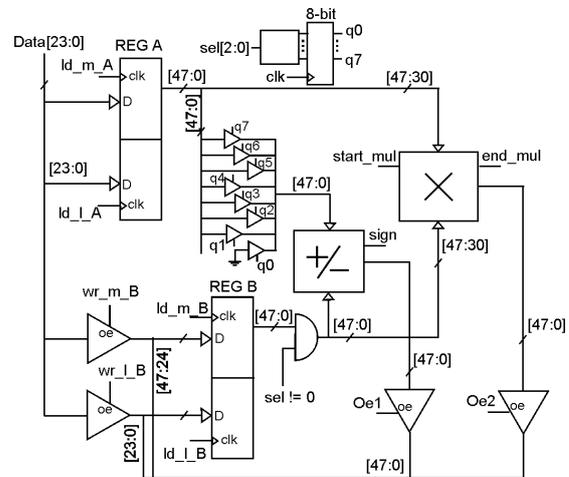


Figure 4 Part for I^2 , V^2 , P, Q accumulating and energy calculation optimized for low-power by operand isolation and gating

The outputs of three state buffers and register B are connected to the inputs of arithmetical circuit (Fig.4). When control signal which represents the input of 3-8 decoder is in range "001" to "111", the corresponding output enable signal is active and new data pass through three state buffers to the adder input. When "000", the write operation into latch is disabled, and thus, the input of the arithmetical operator is not changed. To isolate the second operand of arithmetical circuit, register B output is gated by AND gating cells. When control signal is in range "001" to "111", data propagation through AND cells is enabled.

The results of optimizations are given in Table II. The modifications in multiplexer circuit didn't give the expected results. The obtained power is increased because of large net capacitances at the three-state circuit outputs.

D. FSM state gray encoding and decomposition

State encoding or state assignment techniques is crucial step in the synthesis of the low-power controller circuitry

[6,7]. The techniques augment the state transition graph with state probabilities, and also, transition probabilities between the states and use these probabilities to guide the state assignment. Adjacent Gray binary encodings are assigned to the states connected with a high probability transition. This minimizes the number of state transitions, thus attempting to minimize switching activity in next state logic and output logic of synthesized FSM.

To consider the impact of state assignment in the consumed power of the combinational part, a number of heuristics are introduced. The key idea of those heuristics is that a combinational circuit optimized in terms of area is also characterized by low-power consumption. Therefore, beside transition probabilities, algorithms take into account the occupied area of the circuit.

The finite state machine that drives the Block 3 controls the operations for removing the DC components from instantaneous values of current and voltage signals, else, the generation of current and voltage square, active and reactive power signals and their accumulation over the time, and, generation of pulses necessary for energy measurement. The sequence of states is simply encoded in Gray binary code during the synthesis process. This was considered as good idea for power reduction because the fact that the most states appear only once in 256-clock cycle lasting sequence. Beside, the state transition is regular in a way that for some state the next state is known in advance or there exist a small number of possible next states.

Decomposition of finite state machines has also been used to reduce the power.[8,9] The basic idea is to decompose the state transition graph of a finite state machine into two or more graphs that jointly produce the equivalent input-output behavior as the original machine. The states are partitioned by searching for a subset of states with high probability of transitions among these states and a low probability of transitions to and from other states. This subset of states then constitute a small sub-FSM which is active most of the time. When the small sub-FSM is active, the other larger sub-FSM can be disabled. Power is saved because, except for transitions between the two sub-FSMs, only one of the sub-FSMs needs to be clocked: the sub-FSM which is active at the moment. The other sub-FSMs which are not producing useful data are shut down by disabling the clock signal.

The non-optimized version of FSM (Block 5) has 4 input signals (beside clock and reset), 25 output signals, 229 states, and executes a state sequence which is periodically repeated with a period equal to 256 clock cycles. The FSM is decomposed into two sub FSMs: FSM1 and FSM2. The FSM1 controls the process of I^2 , V^2 , P and Q generation and accumulation, while, FSM2 is responsible for high-pass filtering and energy pulses generating. There exists only one transition from FSM1 to FSM2 during the main, 256 clock lasting period. The subsequences of states last 151 and 105 clock cycles for FSM1 and FSM2, respectively.

The additional control block determines which of two sub FSM is active at the moment. Each sub FSM generates a signal for ending of state sequence which is fed to the control block. The control block produces the two enable signals Enable1 and Enable2. When Enable1 signal is on for FSM1, it is off for FSM2. Conversely, the Enable2 signal is always off for FSM1 while it is on for FSM2.

Beside clock gating, the operand isolation technique is applied on finite state machines. To stop data propagation in the combinatorial logic block in inactive subFSM, the sequence of two-input AND cells are used in front of it. One of the inputs of AND cells is the FSM's input signal which is gated and the other one is the enable signal from control block.

The benefit in power reduction achieved by disabling a part of finite state machine is slightly degraded by new circuits introduced by decomposition. The new hardware consists of multiplexer circuits at the outputs of sub FSMs and adds extra switching activities.

The design in which the FSM is Gray encoded, and also incorporates clock gating, gives the power reduction of 35%. The final design where FSM is divided into two clock-gated sub-FSMs gives the minimal power consumption. Achieved consumption is 648 μ W and represents 42% reduction of consumption for non-optimized circuit

TABLE II

| Block | FSM gray encoding | | Decomposition with grey encoding | |
|----------------------|-------------------|------------------|----------------------------------|------------------|
| | Area [gates] | Power [μ W] | Area [gate] | Power [μ W] |
| Clock tree | 2 | 87 | 2 | 89 |
| Registers | 456 | 32 | 456 | 28 |
| Three-state circuits | 615 | 96 | 615 | 108 |
| Adder | 320 | 129 | 320 | 108 |
| Multiplexer | 307 | 41 | 307 | 30 |
| Multiplier | 663 | 94 | 663 | 94 |
| FSM circuit | 1081 | 258 | 1246 | 191 |
| Total | 3444 | 737 | 3609 | 648 |

III. OPTIMIZATION OF EMBEDDED 8052 MICROCONTROLLER BLOCK

A. MCU's structure

The instruction set of 8052 microcontroller (MCU) contains 255 instructions, which have variable length in range from one to three bytes. The opcode of an instruction is encoded in its first byte. The optional second and third bytes represent the operands. The instruction set can be considered as a complex, and, the 8052 microcontroller is classified as CISC (Complex Instruction Set Computer) [10,11]. The instructions can be divided into 5 main classes: arithmetical, logical, data transfer, boolean and jump instructions.

The complex and irregular instruction set increase the energy cost of fetching and decoding of instructions. Although the microcontroller does not represent the best choice for energy efficiency, the choice is justified by the fact that it is one of the most popular microcontrollers, which is often found in applications where the energy efficiency is important.

The global structure of microcontroller block embedded into Integrated Power Meter Chip consists of MCU core, memory blocks, the block for programming and initialization and peripheral units.

The MCU core performs fetching, decoding and executing of instructions and consists of Control logic block, Arithmetical-logical unit (ALU) and Special Function Registers I/O control logic.

The on-chip peripherals are comprised of: three digital input/output parallel ports (Port0 and Port1 are 8-bit and Port2 6-bit wide); LCD driver control circuit (driving up to 168 pixels LCD display) and several communication modules - two asynchronous universal receiver/transmitter blocks (USART0 and USART1) and one I2C-like serial interface. Also, three standard 8052 timer/counter circuits are present (TC0, TC1 and TC2).

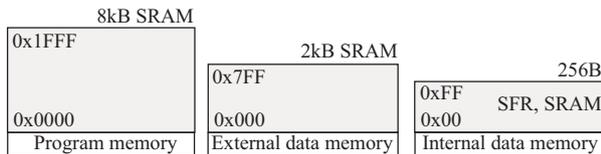


Fig.5. Microcontroller memories

The memory organization is similar to that of the industry standard 8052. Three main memory areas associated with the microcontroller are physically located on the Integrated Power Meter IC. They are illustrated in Fig. 5: Program memory (on-chip 8kB SRAM block), external data memory (physically consisting of XRAM - on-chip 2kB SRAM block, I/O RAM made of standard cells), and internal data memory (Internal RAM comprising of 256 Internal Dual port RAM and Special Function Registers).

The MCU doesn't have internal non-volatile memory for program storing. Instead, MCU utilize on-chip SRAM memory and external EEPROM chip. After the reset state, the program memory is automatically loaded from external EEPROM chip into 8kB SRAM block. The block for programming and initialization is responsible for this operation.

B. Optimization for low power

Optimization of microcontroller's power consumption is difficult task. Digital designers need to undertake a considerable amount of work to realize the most power efficient design.

The first implementation of MCU was made with two goals: to fulfil primary requirements concerning correct

functioning, and the second, to use a minimal number of clock periods for an instruction execution. Since instruction set is complex (has 255 instructions), and 6 different addressing modes exist, the design of microcontroller demanded huge effort. The first implementation of microcontroller was made to be fully synchronous. The architecture has three pipeline stages that execute one-byte instructions in a single cycle. The activities within MCU core are localized as much as possible. Special function registers (SP, PSW, DTPR, A, B) are made to have their own data busses and function units (instead of using shared busses and units) for time saving.

When functionality was achieved, the power reduction became important issue. Clock gating schemes had been extensively used in the further MCU design. For low power consumption of increment logic a ripple carry adders were used. The result of power optimization of clock gating is given in Table III.

After clock gating, great effort was taken to minimize the switching activity: no register and execution unit receives control unless it processes data for a given instruction. Interrupts and pins only cause switching when accessed by software or when an input pin changes. Also, address and data lines for all memories have been made to change only when new data is to be read or written. The memories power save issue is particularly important because memories represent huge power consumers. In modern chips 30% of power is spent on read and write operations.

The total power reduction is 70% comparing the first implementation which met only functionality requirements.

C. MCU's Power saving modes

The implementation of power saving modes provides simple control of power consumption of microcontroller so the most appropriate operation mode can be chosen for any application. The MCU, beside Active operating mode, offers following low-power modes: Power Save, Standby and Power Down mode.

Power saving in Active operation mode should be explained first. One of the solutions to reduce power consumption in this mode is to reduce the clock frequency. Current consumption increases directly with the system clock frequency so keeping the system clock as low as possible is critical to keeping the power consumption down. In Active operation mode, few different clock frequencies are at disposal. The chip uses 32 kHz clock on-chip oscillator. Internal 4.1494 MHz clock signal is generated using on-chip PLL frequency multiplier and microcontroller has the option to use one of the outputs of clock divider circuit as the input clock signal. The nominal frequency of 4.194 MHz can be divided by one of the numbers 1, 2, 4, 8, 16, 32, 64 and 128. The user can select an optimal clock frequency instead of having highly power consuming microcontroller in a much slower system.

TABLE III

| | Block | Non-optimized | | | Optimized by clock gating | | | Optimized by minimization of switching activity | | |
|-----|--------------------------------------|---------------|------------------|-----------|---------------------------|------------------|-----------|---|------------------|-----------|
| | | Area [gates] | Power [μ w] | Clk sinks | Area [gates] | Power [μ w] | Clk sinks | Area [gates] | Power [μ w] | Clk sinks |
| 1. | Clock tree | 0 | 5770 | 0 | 0 | 1420 | 0 | 0 | 642 | 0 |
| 2. | I/O RAM | 2235 | 10 | 244 | 2174 | 21 | 23 | 2179 | 3 | 23 |
| 3. | DSP's interface | 156 | 0 | 7 | 156 | 0 | 7 | 156 | 0 | 7 |
| 4. | Port0 circuit | 328 | 26 | 16 | 328 | 14 | 16 | 314 | 27 | 16 |
| 5. | Port1 circuit | 346 | 21 | 16 | 346 | 12 | 16 | 334 | 26 | 16 |
| 6. | Port2 circuit | 188 | 20 | 12 | 188 | 13 | 12 | 186 | 27 | 12 |
| 7. | TC0 and TC1 | 968 | 58 | 57 | 754 | 43 | 21 | 755 | 55 | 21 |
| 8. | TC2 | 640 | 54 | 52 | 659 | 46 | 23 | 658 | 54 | 23 |
| 9. | UART 0 | 693 | 46 | 73 | 725 | 44 | 22 | 726 | 32 | 22 |
| 10. | UART 1 | 880 | 45 | 86 | 902 | 32 | 50 | 906 | 38 | 50 |
| 11. | I2C | 597 | 41 | 21 | 597 | 33 | 21 | 593 | 44 | 21 |
| 12. | 8052 core | 4468 | 2010 | 266 | 3953 | 1990 | 104 | 4233 | 1286 | 104 |
| 13. | ALU | 2331 | 452 | 120 | 2331 | 276 | 120 | 1965 | 200 | 120 |
| 14. | SFR read/write logic | 992 | 97 | 84 | 941 | 82 | 38 | 986 | 140 | 38 |
| 15. | Programming and initialization logic | 4675 | 163 | 428 | 4809 | 90 | 59 | | 93 | 59 |
| 16. | LCD driver control | 1081 | 33 | 1 | 1081 | 33 | 1 | 1091 | 28 | 1 |
| | | | | | | | | | | |
| | | 20578 | 8846 | 1483 | 19944 | 4149 | 533 | 15082 | 2695 | 533 |

The other power saving method used in Active operation mode is to gate the clock input of the microcontroller parts that are not used. The following peripheral units can be gated: Ports 0, 1, 2; Timer/Counters 0, 1, 2, UARTs 0, 1, and I2C communication controller.

The Power Save is very useful in applications in which microcontroller is often latently waiting the information from some sensor or other microcontroller. When the information is acquired, fast data processing is expected. In this mode, only clock input signal of the microcontroller is blocked out, peripheral units continue its normal operation. Like the Active operation mode, the selected peripheral units of the microcontroller can be gated. Disabling the peripheral modules results in 5-10% reduction of the total power consumption in Active mode, and 10-20% in Power Save mode.

The device can be turned back from Power Save mode to the Active operation mode by two different events: the system reset and interrupt. In the case of interrupt request, the MCU continues with the execution of the next program command and after that starts processing the interrupt and jumping to the interrupt processing routine. The MCU's wake up by reset restarts the program execution. Since the clock generator is active in this mode the wake-up time is short.

In Standby mode, the clock generator producing main clock is operative but clock inputs of microcontroller and peripheral units are gated. In Power Down mode, everything is shut down including the main clock source.

The clock controller module is the part of microcontroller block responsible for power saving modes. The module produces two main clock signals, one dedicated to microcontroller and the other one clocking the

peripheral units. During the Active operation both signals behave equally. In low-power operation modes one or both of the clock signals are stopped. Low-power modes are simply invoked by writing to one of the Special Function registers dedicated for power management.

IV. THE OPTIMIZATION FOR LOW-POWER OF IPM'S DIGITAL BLOCKS

TABLE IV

| Block | Area [gates] | Power [mW] |
|---------------------------|--------------|------------|
| Sinc- Current | 4623 | 0.238 |
| Sinc- Voltage | 7077 | 0.275 |
| FIR - Current | 6491 | 0.472 |
| FIR- Voltage | 6607 | 0.489 |
| Hilbert filter | 8820 | 0.323 |
| DSP | 21425 | 1.150 |
| RTC | 1437 | 0.002 |
| XRAM - 2kB | 18884 | 0.010 |
| Int. Dual Port RAM - 256B | 7796 | 0.310 |
| Program memory SRAM 8kB | 50030 | 2.238 |
| MCU | 15082 | 2.695 |
| Total: | 148272 | 8.202 |

The power optimization results for digital blocks are obtained after Verilog simulations during which complete switching activity was recorded.

The chip was implemented in AMI CMOS 0.35 μ m standard cell technology. Design was first described in VHDL, and after, synthesized by Cadence's Build Gates tool. The digital signal blocks of Integrated Power Meter

are carefully designed to prevent synchronization errors between them. Also, the blocks are power optimized using techniques described above. The layout was generated by Cadence's tool First Encounter. Signal delays were obtained considering parasitic capacitances of nets in the layout. The Verilog netlists, extracted from layout, were simulated by NCSim logical verification tool. Switching activity file, which was obtained after Verilog simulation, was imported into First Encounter for estimation of average power consumption.

The power consumption of blocks is given in the Table I. Two blocks that consume the most of power are MCU and DSP block. The total power consumption of digital part of a chip is 8.202mW.

IV. CONCLUSION

In this paper, a low power Integrated Power Meter IC is presented. The chip incorporates several digital data processing blocks: filters, digital signal processor dedicated to power metering and embedded microcontroller.

Two blocks identified as blocks with the highest power consumption are DSP block and embedded microcontroller. The applied low-power techniques are mainly based on clock and data gating. Clock gating incorporated into the DSP induced the significant power saving - reducing the overall power by 27%. After DSP's state machine had been Gray encoded, the power reduction gain became 35%. The total power reduction of 42% is achieved by FSM's state decomposition used along with the other two techniques.

Great effort was taken to minimize the switching activity of embedded MCU: no register and execution unit receives control unless it processes data for a given instruction. The microcontroller's control logic was built in a way that address and data lines for the memories change only when new data is to be read or written. The clock gating was used in the design wherever it was possible. The total power reduction is 70% comparing the first implementation which met only functionality requirements.

The main objective, which was to realize power efficient design, was fully reached. Measurement on the chip, which will be in manufacture, has to be carried out, to confirm those results.

REFERENCES

- [1] Jovanović, B., Damjanović, M., Petković, P. "Digital Signal Processing for an Integrated Power Meter ", Conference Proceedings of 49. Internationales Wissenschaftliches Kolloquium Technische Universität Ilmenau 27-30 September 2004, Vol. 2, pp. 190-195
- [2] Damjanović, M., Jovanović, B., "Energy Calculation in Power Meter IC", Zbornik radova sa V simpozijuma industrijske elektronike INDEL 2004, pp. 126-131.
- [3] Sokolović, M., Jovanović, B., Damjanović, M., "Decimation Filter Design", Proc of 24. Int. Conf. on Microelectronics MIEL 2004, pp. 601-604
- [4] Chandrakasan, A., Sheng, S., Brodersen, R., "Low-Power CMOS Digital design", IEEE Journal Of Solid-State Circuits., Vol 27, No 4., April 1992, pp. 473-484
- [5] Wu, Q., Pedram, M. Wu, X., "Clock-Gating and Its Application to Low Power Design of Sequential Circuits", IEEE Proc. of CICC, Santa Clara, 1997, May, pp.479-482
- [6] Benini, L.; De Micheli, G. /"State assignment for low power dissipation", Solid-State Circuits, IEEE Journal of Volume 30, Issue 3, Mar 1995 pp.:258 – 268
- [7] Wu, X.; Pedram, M.; Wang, L.; "Multi-code state assignment for low power design", Circuits, Devices and Systems, IEE Proceedings -Volume 147, Issue 5, Oct. 2000 pp. 271 - 275
- [8] Chow, S.H., Yi-Cheng Ho, Y.C., Hwang, T., "Low power realization of finite state machines - a decomposition approach", ACM Transactions on Design Automation of Electronic Systems (TODAES) Volume 1, Issue 3 (July 1996) pp.: 315 – 340, ISSN:1084-4309
- [9] Lee, W.K., Chi-Ying Tsui, C.Y., "Finite state machine partitioning for low power", Circuits and Systems, 1999. ISCAS'99., Proceedings of the 1999 IEEE International Symposium, Volume 1, June 1999, pp. 306 – 309
- [10] Martin, A.J.; Nystrom, M.; Papadantonakis, K.; Penzes, P.I.; Prakash, P.; Wong, C.G.; Chang, J.; Ko, K.S.; Lee, B.; Ou, E.; Pugh, J.; Talvala, E.-V.; Tong, J.T.; Tura, A., "The Lutonium: a sub-nanojoule asynchronous 8051 microcontroller", Asynchronous Circuits and Systems, 2003. Proceedings. Ninth International Symposium on 12-15 May 2003 pp. 14 – 23
- [11] Manet, P., Bol, D., Ambroise, R., Legat, J.D., "Low Power Techniques Applied to a 80C51 Microcontroller for High Temperature Applications", Journal of LowPower Electronics, Volume 2, Number 1, April 2006 , pp. 95-104
- [12] Lim, K.M., Jeong, S.W., Kim, Y.C., Jeong, S.J., Kim, H.K., Kim, Y.H., Chung, B.Y., Roh, H.L., Yang, H.S. "CalmRISCTM: A Low Power Microcontroller with Efficient Coprocessor Interface", Computer Design, 1999. (ICCD '99) International Conference on 10-13 Oct. 1999 pp. 299 – 302
- [13] Van Gageldonk, H.; Van Berkel, K.; Peeters, A.; Baumann, D.; Gloor, D.; Stegmann, G. "An Asynchronous Low-Power 80C51 microcontroller" , Advanced Research in Asynchronous Circuits and Systems, Proceedings, Fourth International Symposium, 1998, pp. 96-107
- [14] Yu Zhou; Hui Guo, "Application Specific Low Power ALU Design", Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on Volume 1, 17-20 Dec. 2008 pp. 214-220

Analysis of Real-Time Systems Timing Constrains

Sandra Đošić and Milun Jevtić

Abstract - In this paper we analyze timing constrains of one fault tolerant hard real-time system with time redundancy. Our goal is to analyze possibility for overcoming transient faults, which are detected during tasks executions, using technique of executing task again or executing some alternative task. We created and presented in the paper program for estimation of possibility to overcome transient failure in one real-time system. On the basis of timing characteristics of real-time tasks and the value of redundant time we can find the value for minimum time between two consecutive faults which real-time system can tolerate.

Keywords - Real-time systems, Response time analysis, Fault tolerance.

I. INTRODUCTION

A system is said to be real-time if the total correctness of an operation depends not only upon its logical correctness, but also upon the time in which it is performed, [1]. The classical conception is that in a hard real-time system, the completion of an operation after its deadline is considered useless - ultimately, this may cause a critical failure of the complete system. A soft real-time system on the other hand will tolerate such lateness, and may respond with decreased service quality (e.g., omitting frames while displaying a video).

One of the goals during real-time systems designing process is to create predictable real-time systems. Analysis of real-time systems timing constrains is fundamental for design such systems. Designing predictable real-time systems is easier with the assumption that there is no fault during system execution. However, this fault-free assumption is, in fact, not realistic because "non-faulty systems hardly exist, there are only systems which may have not yet failed", [2]. So, if a fault occurs during real-time tasks execution then it is necessary to overcome that fault and satisfied all real-time tasks timing constraints.

Focus of our research is fault tolerant hard real-time systems and in this paper we will analyze timing constrains for such systems, [3]. We also wrote program for that analyses. Input data for program are timing characteristics of real-time tasks and the result is minimum time between two consecutive faults which real-time system can tolerate. Due to result of analysis we can conclude how much is one real-time system fault tolerant.

Sandra Đošić and Milun Jevtić are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: (milun.jevtic, sandra.djosic)@elfak.ni.ac.rs.

II. SOFTWARE REALIZATION OF ALGORITHM FOR ANALYSIS RTS TIMING CHARACTERISTICS

A. Response time analysis

One of the goals of our research is to the design predictable hard real-time systems. Response time analysis is one approach that has successfully been used to achieve this goal. The basis of response time analysis is Eq. (1) and more about that analysis can be found in [3].

$$R_i(T_E) = C_i + \sum_{\tau_j \in hp(i)} \left\lceil \frac{R_i(T_E)}{T_j} \right\rceil C_j + \left\lceil \frac{R_i(T_E)}{T_E} \right\rceil \max_{\tau_k \in hpe(i)} \overline{C}_k \quad (1)$$

We use response time analysis for set $\Gamma = \{\tau_1, \dots, \tau_n\}$ of n real-time tasks, called primary tasks, that must be scheduled by the system in the absence of faults. Any primary task τ_i , in a set Γ , has a period T_i , a deadline D_i ($D_i \leq T_i$), and a worst-case execution time, C_i . Each primary task τ_i can have some alternative tasks $\overline{\tau}_i$ associated with it, [4]. Each alternative task represents some extra processing that is necessary to recover a task from a given faulty state caused by a fault. Any alternative task has a worst-case execution time, called worst-case recovery time, \overline{C}_i .

We also consider n different priority levels (1, 2, ..., n), where 1 is the lowest priority level. We denote the priority of primary task τ_i and alternative tasks $\overline{\tau}_i$ as p_i and \overline{p}_i , respectively. We also assume in the analysis that there is a minimum time between two consecutive fault occurrences, T_E .

The input parameters of this analysis are: the task attributes (T_i , D_i , C_i and \overline{C}_i), the primary task priorities (p_i) and the assumed value of T_E . The priorities of alternative tasks are assumed to be the same as their primary tasks $\overline{p}_i = p_i$.

If there is no faults in the system then the worst-case response time of task τ_i is the time necessary to execute τ_i and all tasks τ_j such that $p_j > p_i$. When faults are considered in the system, we have to include in the calculation of the worst-case response time of τ_i the time necessary to recover the faulty task. We use time redundancy for systems recovering, [5].

Since R_i appears on both sides of the Eq. (1), the solution can be obtained iteratively by forming a recurrence relation with $R_i^0 = C_i$. This iterative procedure finishes either when $R_i^{m+1} = R_i^m$ (the worst-case response time of τ_i

is found) or when $R_i^{m+1} > D_i$ (τ_i is considered unschedulable).

Fig. 1 illustrates possible scenarios of real-time tasks scheduling with different assumed value of T_E .

The first scenario, Fig. 1(a) presents scheduling of two periodic real-time tasks τ_1 and τ_2 when there is no fault in the system. System of these two tasks are schedulable i.e. both tasks execute before their deadlines, D_1 and D_2 .

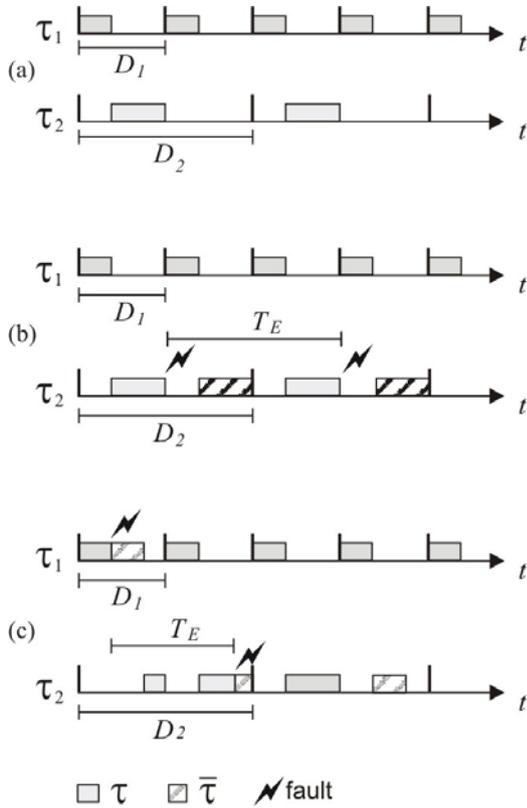


Fig. 1. Illustration of possible real-time tasks schedule when: (a) there is no fault; (b) value for T_E is long enough and real-time system is fault tolerant; (c) value for T_E is not long enough that real-time system stays fault tolerant

Fig. 1(b) presents scheduling of the same real-time tasks τ_1 and τ_2 when two faults occur in the system. Time between two consecutive faults T_E is long enough and real-time system can tolerate these faults. First fault occurs just a little bit before the end of tasks τ_2 execution. Real-time system overcomes this fault by executing task τ_2 again or executing alternative tasks with less or equal execution time as task τ_2 . Second fault occurs again just a little bit before the end of tasks τ_2 execution. Time redundancy is enough to tolerate this fault too. Like before, when the first fault occurs, system overcomes fault by executing task τ_2 again or executing some alternative tasks.

Fig. 1(c) presents scheduling of the same real-time tasks τ_1 and τ_2 when two faults occur in the system. Now, time between two consecutive faults T_E is not long enough and real-time system cannot tolerate these faults. First fault

occurs just a little bit before the end of tasks τ_1 execution. Real-time system can overcome this fault by executing task τ_1 again or executing alternative tasks with less or equal execution time as task τ_1 . In this case, second fault occurs just a little bit before the end of tasks τ_2 execution. Now time redundancy is not enough to tolerate this fault. System starts procedure for overcoming fault by executing task τ_2 again but timing characteristics if tasks τ_2 cannot be satisfied and τ_2 missing its deadline. This is not acceptable in one hard real-time system, so in this case real-time system is not fault tolerant.

B. The Algorithm

Based on the Eq. (1) we realized algorithm for analysis real-time systems timing constraints shown in Fig. 2. Input data are number of real-time tasks n , task period T_i , worst-case execution time C_i , worst-case recovery time \bar{C}_i , task deadline d_i and task priority p_i . For these parameters algorithm have to check if the real-time system is fault tolerant. We considered that fault can occur during tasks execution and that is necessary to execute some recovery tasks for faults overcome. The goal of algorithm is to find minimum time between two consecutive faults which real-time system can tolerate.

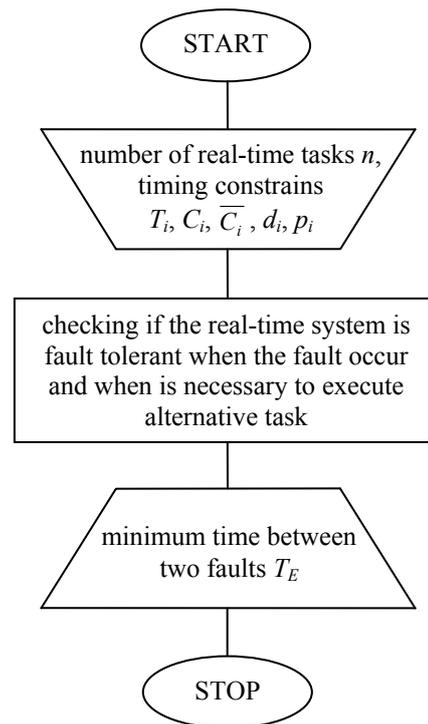


Fig. 2. Algorithm for analysis RTS timing constraints

Fig. 3 shows more detailed algorithm for analysis real-time systems constraints. Input data for shown algorithm are number of real-time tasks n , task period T_i , worst-case

execution time C_i , worst-case recovery time \overline{C}_i , task deadline d_i and task priority p_i , step (1) on Fig. 3. In the beginning we assume in the analysis that minimum time between two consecutive fault occurrence is $T_E = 1$ step (2) on Fig. 3.

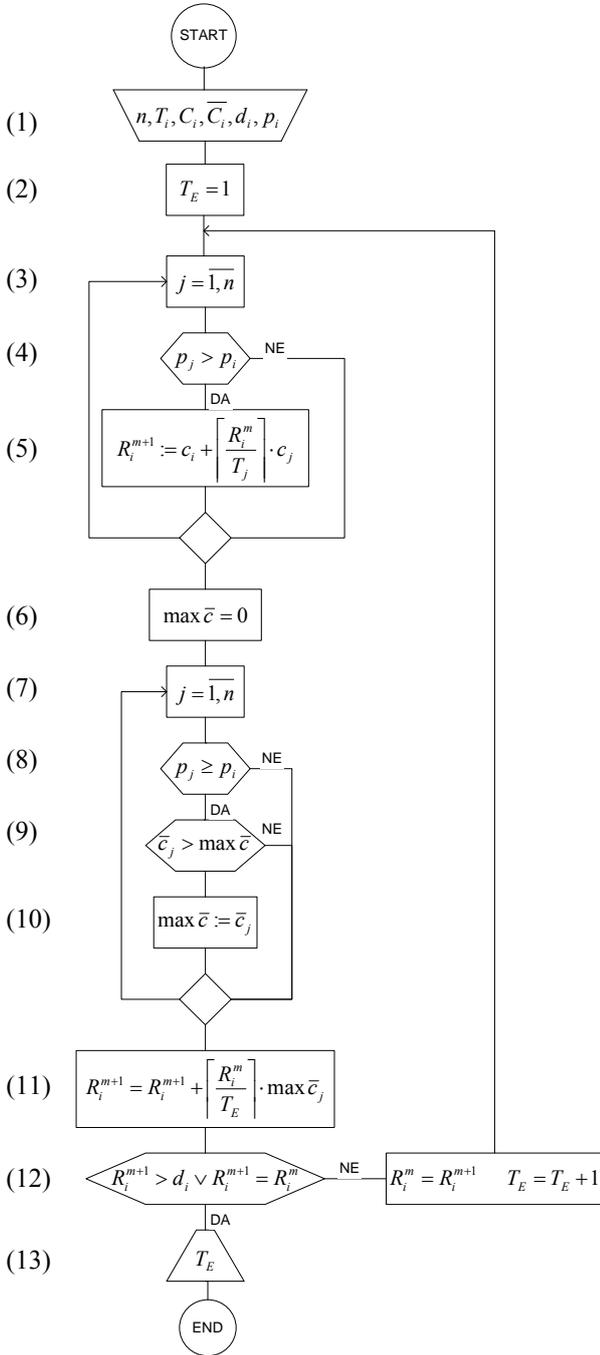


Fig. 3. More detailed algorithm for analysis RTS timing constrains

In the first algorithm loop, step (3), step (4) and step (5) on Fig. 3, we calculate the first and the second addend of

Eq. (1). In this loop only task with higher priority then priority of task τ_i are important for us.

The second loop in algorithm, step (6) to (10) on Fig. 3, describe process of finding maximum worst-case recovery time from the tasks with equal or higher priority then priority of task τ_i .

Step (11) on Fig. 3 calculates the worst-case response time R_i for task τ_i . According to Eq. (1) this process is iterative and it finishes either when $R_i^{m+1} > d_i$ (τ_i is considered unschedulable) or when $R_i^{m+1} = R_i^m$ (the worst-case response time of τ_i is found), step (12) on Fig. 3. If the condition step (12) is true then we have output result T_E step (13) on Fig. 3. If the condition step (12) is false then we must increase T_E and continue iterative process until it is necessary.

Using algorithm shown on Fig. 3 we wrote code and generated .exe file “AlgFix.exe” which could be started from command line with command:

ALGFIX [<input file>] [<output file>].

As you can see from the above command, optionally the name of the input and output file could be written. If you don't write name for the input and output file then their standard name “AlgFix Input.txt” and “AlgFix Output.txt” are considered.

Input file is .txt format with parameters separate with space. In the first line, the number of real-time tasks n should be written. After that in the next n line we have to specify timing characteristics of n real-time tasks: period T_i , worst-case execution time C_i , worst-case recovery time \overline{C}_i , deadline d_i and task priority p_i .

Output file is also .txt format with parameters separate with space. The first line is required result T_E - minimum time between two consecutive faults which real-time system can tolerate. In the next n line are parameters R_i^m and R_i^{m+1} for each of n real-tasks.

C. Results of Software Realization

In order to prove the correctness of the realized algorithm and the whole program, we do a number of tests and two of them are shown on Fig. 4.

Fig. 4(a) presents input and output file for case I of three real-time tasks scheduling according with rate monotonic algorithm, [6], [7].

Timing characteristics for these three tasks are shown in Table I and inputs file “AlgFix Input” on Fig. 4(a). For these parameters, we started our program for analyses. Program considers that faults can occur in real-time system during task execution and that system recovers executing task again. Therefore, for this case worst-case recovery time is equal as worst-case execution time, $C_i = \overline{C}_i$.

The output file “AlgFix Output.txt” shows results of timing analyses. From that file, we can see that real-time systems can tolerate minimum time between two

consecutive fault occurrences of 11 time units. For $T_E = 11$ parameters are $R_1(11) = 4$, $R_2(11) = 8$ and $R_3(11) = 22$ and this is also shown in output file. For all three tasks we got that $R_i(11) < d_i$, for $i = 1, 2$ and 3 , what means that all tasks finished before their deadlines. It can be concluded that system is schedulable.

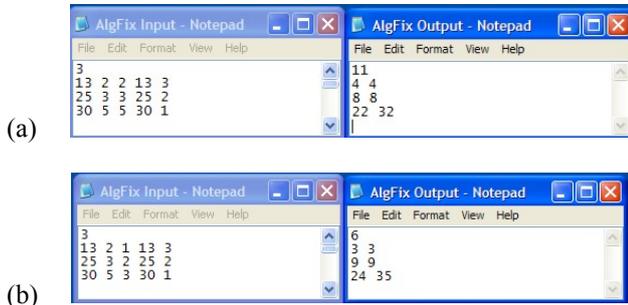


Fig. 4. Input and output file of realized software for
 (a) case I - system recovers executing task again
 (b) case II - system recovers executing alternative task

For $T_E = 10$ parameters are $R_1(10) = 4$, $R_2(10) = 8$ and $R_3(10) = 32$ and they are also shown in output file “AlgFix Output.txt”. For task τ_3 we got that $R_3(10) > d_3$ what means that this task overflows its deadline, so the whole system is not schedulable.

TABLE I
 REAL-TIME TASKS TIMING CHARACTERISTICS - CASE I

| Task | Task characteristics | | | | |
|----------|----------------------|-------|-------------|-------|-------|
| | T_i | C_i | \bar{C}_i | d_i | p_i |
| τ_1 | 13 | 2 | 2 | 13 | 3 |
| τ_2 | 25 | 3 | 3 | 25 | 2 |
| τ_3 | 30 | 5 | 5 | 30 | 1 |

Table II presents manually obtained results for the same input parameters. If we compare the output file “AlgFix Output.txt” and Table II, it can be conclude that we got the same results, manually and software obtained.

TABLE II
 RESULTS OF TIMING ANALYSES FOR CASE I

| Task | $R_i(11)$ | $R_i(10)$ |
|----------|-----------|-----------|
| τ_1 | 4 | 4 |
| τ_2 | 8 | 8 |
| τ_3 | 22 | 32 |

The second case presents real-time system that recovers from the fault executing some alternative tasks. Usually those tasks have less worst-case execution time then primary tasks, i.e. $\bar{C}_i < C_i$. This case is shown on Fig. 4(b).

Fig. 4(b) presents input and output file for case II of three real-time tasks scheduling also according with rate

monotonic algorithm. Timing characteristics for these three tasks are shown in Table III and inputs file “AlgFix Input” on Fig. 4(b). For these parameters, we started our program for analyses. Program considers that faults can occur in real-time system during task execution and that system recovers executing alternative tasks whose worst-case recovery time is less then as worst-case execution time of primary task, $\bar{C}_i < C_i$.

TABLE III
 REAL-TIME TASKS TIMING CHARACTERISTICS – CASE II

| Task | Task characteristics | | | | |
|----------|----------------------|-------|-------------|-------|-------|
| | T_i | C_i | \bar{C}_i | d_i | p_i |
| τ_1 | 13 | 2 | 1 | 13 | 3 |
| τ_2 | 25 | 3 | 2 | 25 | 2 |
| τ_3 | 30 | 5 | 3 | 30 | 1 |

The output file “AlgFix Output.txt” shows results of timing analyses. From that file, we can see that real-time systems can tolerate minimum time between two consecutive fault occurrences of 6 time units. For $T_E = 6$ parameters are $R_1(6) = 3$, $R_2(6) = 9$ and $R_3(6) = 24$ and this is also shown in output file. For all three tasks we got that $R_i(6) < d_i$, for $i = 1, 2$ and 3 , what means that all tasks finished before their deadlines. It can be concluded that system is schedulable.

For $T_E = 5$ parameters are $R_1(5) = 3$, $R_2(5) = 9$ and $R_3(5) = 35$ and they are also shown in output file “AlgFix Output.txt”. For task τ_3 we got that $R_3(5) > d_3$ what means that this task overflows its deadline, so the whole system is not schedulable.

TABLE IV
 RESULTS OF TIMING ANALYSES FOR CASE II

| Task | $R_i(6)$ | $R_i(5)$ |
|----------|----------|-----------|
| τ_1 | 3 | 3 |
| τ_2 | 9 | 9 |
| τ_3 | 24 | 35 |

Table IV presents manually obtained results for the same input parameters. If we compare the output file “AlgFix Output.txt” and Table IV, it can be conclude that we got the same results, manually and software obtained.

III. CONCLUSION

In this paper, we presented program for analyzing timing constraints of real-time tasks in one real-time system. We considered that these tasks are schedule according with rate monotonic algorithm and that faults can occur during tasks execution. We also considered that real-time system recovers from faults executing task again (case I) or executing some alternative tasks (case II). In both

cases, we use time redundancy for systems recovery after faults. For these two cases, we do a number of tests and prove the correctness of the realized algorithm and the whole program.

We specially presented two cases of tree real-time tasks whose input parameters are almost the same, the only difference is value for worst-case recovery time. Case I presents real-time system who recovers from faults executing task again, so $C_i = \overline{C}_i$. Case II presents real-time system who recovers from faults executing some alternative tasks whose worst-case recovery time are less then tasks worst-case recovery time, i.e. $\overline{C}_i < C_i$. If we compare output results for case I and case II we can conclude that if worst-case recovery time is less than minimum time between two consecutive fault occurrences which systems can tolerate is also less. This reduction of parameter T_E indicates increasing real-time fault tolerance, what is good.

Realized program offers the possibility to analyze timing constraints of multiple real-time tasks very fast, much faster than manually obtained. From the output program result, we also got information about minimum time between two consecutive fault occurrences that systems can tolerate. This is important information from which we can conclude how much is one real-time system fault

tolerant.

REFERENCES

- [1] Nisanke, N., "Realtime Systems", Prentice Hall, 1997.
- [2] Laprie, J.C., "Dependability: Basic Concepts and Terminology", Springer-Verlag, 1992.
- [3] Lima, G., Burns, A., "An Optimal Fixed-Priority Assignment Algorithm for Supporting Fault-Tolerant Hard Real-Time Systems", IEEE Transaction on Computers, Vol. 52, No. 10, October, 2003, pp. 1332-1346.
- [4] Johnson, B., "Design Analysis of Fault-Tolerant Digital Systems", Addison-Wesley Publishing Company, 1988.
- [5] Đošić, S., Jevtić, M., "Planiranje zadataka u sistemu za rad u realnom vremenu sa redundansom u vremenu za prevazilaženje otkaza", Zbornik radova V simpozijuma industrijske elektronike, INDEL 2004, Banja Luka, novembar 2004, pp. 146-149.
- [6] Cottet, F., Delacroix, J., Mammeri, Z., "Scheduling in Real-Time Systems", John Wiley & Sons, 2002.
- [7] Juvva, K., "Real-Time Systems", Carnegie Mellon University, 18-849b Dependable Embedded Systems, or http://www.ece.cmu.edu/~koopman/des_s99/real_time/index.html

An approach to Digital Low-Pass IIR Filter Design

Bojan Jovanović, and Milun Jevtić

Abstract – The paper describes the design process of discrete network – digital low-pass filter with Infinite Impulse Response (IIR filter). Based on given parameters that network should meet, and with the use of HDL hardware description language and MATLAB software package, hardware implementation of the filter transfer function was designed. Simulations confirmed the validity of the implementation design in an FPGA chip.

Keywords – Digital low-pass IIR filter, VHDL, MATLAB.

I. INTRODUCTION

Digital processing of continuous signal is based on numerical processing of the data which are used to present continuous signals. It has its beginning in the first numerical methods for solving mathematical problems such as numerical solution of integral-differential equations, numerical solution of integrals, interpolation etc. With the development of the computing machines already existing algorithms for digital signal processing are applied. New algorithms are also developed. Fig. 1 shows a simplified block diagram of digital processing of continuous signals [1]. Continuous signal that is processed is marked with $x(t)$. After analog-to-digital conversion in A/D block we get the digital signal $\{x(n)\}$ which is presented in the form of an array of numbers. This signal is then processed using some processing device (DSP, FPGA, CPLD etc.). We will adopt that $x(n)$, which represents the output from A/D converter, is 8-bit binary number represented in second complement. Signal from the output of processing device is also an array of numbers $\{y(n)\}$. This array is converted into continuous signal $y(t)$ using D/A block for digital-to-analog conversion. Each member of the array $\{y(n)\}$ is also 8-bit binary number represented in the second complement.

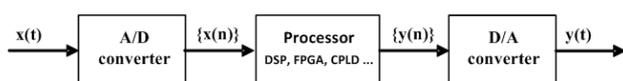


Fig. 1. Digital processing of continuous signals

Digital signal processing has a very broad application areas, ranging from simulation of analog (continuous) networks on digital computers to the development of new digital systems that will completely replace analogue. There are many advantages of digital over analog signal processing. Digital signal processing can be realized with higher degree of accuracy. Then, one computer can

Bojan Jovanović and Milun Jevtić are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia,
E-mail: bojan@elfak.ni.ac.rs. milun.jevtic@elfak.ni.ac.rs

simultaneously perform multiple digital signal processing's. Digital networks are very flexible, easily can be changed by changing the program parameters. Significant advantage is also in the fact that the computer can execute a series of transformations that can not be achieved using analogue elements, either because of the complexity of the transformations, either due to lack of analogue elements required for transformation. The changes of network parameters due to aging of components does not exist in digital networks. Characteristics of digital networks are stable, and reliability is great. We should not neglect the fact that the price of digital components is significantly lower than the cost of analog components. In terms of physical dimensions of the networks the advantage is also on the digital processing side.

The limitation of digital networks is that the signal processing takes some time. Also, the use of digital networks for processing high frequency digital signals is limited. Besides the limitations related to the speed of digital signal processing digital networks have more drawbacks, such as the existence of the noise and low dynamic of the signal. Noise can not be avoided because it occurs as a result of signal quantisations, rounding of the product and rounding of the sum in the numerical calculations. Noise reducing and increasing of the signal dynamic can be achieved by increasing the length of digital words which are used to represent numbers. However, the speed of filter reduces with digital word length increasing. It is therefore necessary to find compromise between the filter speed and the size of the noise generated by digital network [2].

II. PROCEDURE OF DESIGNING RECURSIVE IIR FILTER

A. Sampling Theorem

According to sampling theorem, frequency limited continuous signal $x(t)$ can be reconstructed from its discrete samples if the sampling frequency is at least $F_s=2f_c$, where f_c is the cutoff frequency in continuous signal frequency spectrum. Analog signals (audio, video etc.) can therefore be processed digitally and the processing result can be converted to analog domain. One of the key requirements for correct processing of analog signal $x(t)$ is the frequency of its sampling F_s . Sampling frequency must be at least twice the highest frequency from the frequency spectrum of signal $x(t)$. To properly reproduce 20kHz audio signal it must be sampled at least 40.001 times per second ($20.000 \times 2 + 1$).

B. Designing task

Design digital low-pass filter with Infinite Impulse Response and the following characteristics:

- Sampling frequency: $F_s=44kHz$;
 - Maximal passband frequency: $f_p=5kHz$;
 - Minimal stopband frequency: $f_a=8kHz$;
 - Maximal signal attenuation in passband $a_{max}=1dB$;
 - Minimal signal attenuation in stopband $a_{min}=50dB$;
 - For referent filter adopt Chebyshev approximation.
- Filter attenuation characteristics are given in Fig. 2.

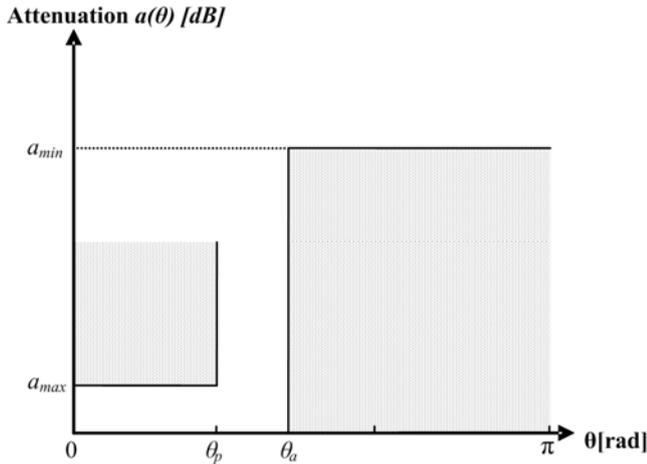


Fig. 2. Attenuation characteristic of low-pass filter

C. Designing procedure

Fig. 2 shows filter attenuation characteristic in respect to discrete frequency θ which is in following relation with analog frequency f :

$$\theta = \frac{2\pi f}{F_s} \quad (1)$$

where F_s is sampling frequency.

For discrete frequency $\theta=\pi$ analogue frequency is $f=F_s/2$. According to sampling theorem this is the maximal frequency in frequency spectrum of signal $x(t)$. Therefore we consider attenuation characteristic for discrete frequency range $[0, \pi]$. Passband frequency range is $[0, \theta_p]$ and maximal signal attenuation in this range is $a_{max}=1dB$. Stopband frequency range is $[\theta_a, \infty]$ and minimal signal attenuation in this range is $a_{min}=50dB$. Frequency range $[\theta_a, \theta_p]$ represents transition zone. There are not defined tolerances of attenuation characteristic.

Based on given filter parameters, and using MATLAB software package [3] we calculate the filter's order and the polynomial coefficients of filter transfer function in z-domain – $H(z)$ (Fig. 3).

$$H(z) = \frac{B(z)}{A(z)} \quad (2)$$

```

Rp=1; % amax [dB]
Rs=50; % amin [dB]
Fs=44; % sampling frequency [kHz]
fp=5; % maximal passband frequency [kHz]
fs=8; % minimal stopband frequency [kHz]
wp=2*fp/Fs; % normalized max passband freq.: wp [0,1]
ws=2*fs/Fs; % normalized min stopband freq.: ws [0,1]
[N,wn]=cheb1ord(wp,ws,Rp,Rs); % calculating the order of Chebyshev filter
[b,a]=cheby1(N,Rp,wn); % calculation of the polynomial coefficients
    
```

Fig. 3. MATLAB function for filter order and polynomial coefficients calculation

After calculation in MATLAB, for filter order is obtained $N=7$, and filter transfer function coefficients are:
 $b = 10^{-3} * [0.0176 \ 0.1232 \ 0.3696 \ 0.6160 \ 0.6160 \ 0.3696 \ 0.1232 \ 0.0176]$;
 $a = [1.0000 \ -5.5152 \ 13.7614 \ -20.0229 \ 18.2902 \ -10.4726 \ 3.4791 \ -0.5178]$.

For filter transfer function we have the following equation:

$$H(z) = \frac{b_0 z^7 + b_1 z^6 + \dots + b_6 z^1 + b_7}{a_0 z^7 + a_1 z^6 + \dots + a_6 z^1 + a_7} \quad (3)$$

Stability condition for discrete networks is that all the poles of network transfer function lie within unity circle in z-plane. MATLAB function `zplane(b,a)` calculates and displays in z-plane zeros and poles of filter transfer function (Fig. 4).

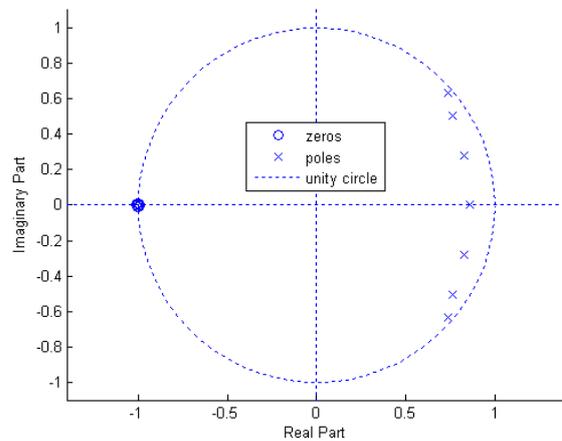


Fig. 4. Zeros and poles of filter transfer function

From Fig. 4 can be seen that discrete network meets stability requirement.

The next step is quantization of coefficients of the polynomial transfer function. Coefficients quantization means their presentation with finite number of bits.

Considering that some coefficients are negative and the maximal absolute value of coefficients is 20.0229, to represent the integer part we need a minimum of 6 bits. It is now needed to determine the minimum number of bits to represent fractional part so that filter attenuation characteristics $a(\theta)[dB]$ do not exceed beyond given tolerances. Filter attenuation is defined as the amplitude characteristic of the filter transfer function $H(z)$ on the unit circle and it is expressed in decibels:

$$a(\theta) = -20 \log |H(e^{j\theta})| \quad (4)$$

Fig. 5 shows comparative characteristics of the filter attenuation in case of quantized and unquantized coefficients of polynomial of transfer function. Quantized coefficients are represented with 17 bits, in the format [17 11], 6 bits for integer, and 11 bits for fractional part.

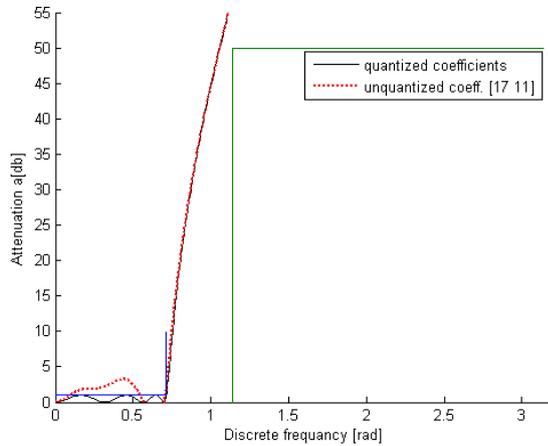


Fig. 5. Attenuation characteristic for quantized [17 11] and unquantized coefficients

A set of MATLAB commands used for calculate and plot comparative characteristics is shown in Fig. 6.

```

q=quantizer('fixed','round','saturate',[17 11]);
Bbin1=num2bin(q,b);% conversion of B(z) coeff. into [17 11] binary format
Bdek1=bin2num(q,Bbin1);% conv. of binary format [17 11] into decimal number
Abin1=num2bin(q,a);% conversion of A(z) coeff. into [17 11] binary format
Adek1=bin2num(q,Abin1);% conv. of binary format [17 11] into decimal number
[h,w]=freqz(b,a,1000);% h - filter transfer funct. with unquantized coeff.
[h1,w]=freqz(Bdek1,Adek1,1000);% h1 - filter trans func. with quant coeff.
plot(w,-20*log10(abs(h)), 'k',w,-20*log10(abs(h1)), 'r',[0 10*pi/44 10*pi/44],[1 1 10],[4*pi/11 4*pi/11 pi],[0 50 50])
legend('quantized coefficients','unquantized coeff. [17 11]');
xlabel('Discrete frequency [rad]');
ylabel('Attenuation a[db]');
    
```

Fig. 6. MATLAB code for calculating and plot filter attenuation characteristics

From Fig. 5 can be seen that the attenuation characteristic with quantized coefficients exceeds outside the given tolerances for passband maximal attenuation, ie. that

attenuation in passband is greater than 1dB. Therefore, for the presentation of fractional part one more bit is added. Quantized coefficients are now in the format [18 12]. Fig. 7 shows comparative characteristics of the filter attenuation in case of quantized (in the format [18 12]) and unquantized coefficients.

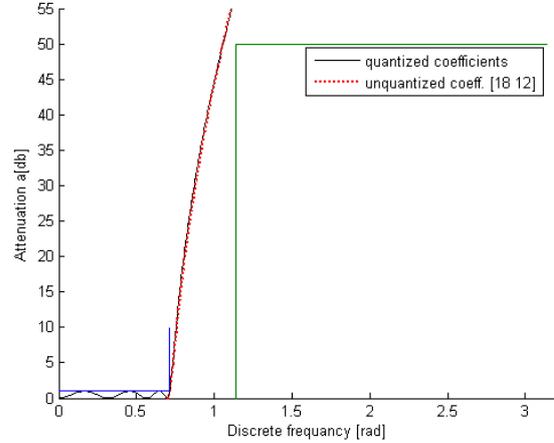


Fig. 7. Attenuation characteristic for quantized [18 12] and unquantized coefficients

Fig. 7 shows that with quantization in the format [18 12] we obtain filter attenuation characteristic that meets the given tolerances both in passband and in stopband.

In addition to the direct realization of the transfer function, there are cascade, parallel, scale and grid implementations.

Cascade realization is based on factorization of digital filter transfer function into the product of transmission functions of the first and second order:

$$H(z) = G \prod_{i=1}^K H_i(z) \quad (5)$$

where $H_i(z)$ are the transfer functions of the first:

$$H_i(z) = \frac{1 + b_{1i}z^{-1}}{1 + a_{1i}z^{-1}} \quad (6)$$

or second order:

$$H_i(z) = \frac{1 + b_{1i}z^{-1} + b_{2i}z^{-2}}{1 + a_{1i}z^{-1} + a_{2i}z^{-2}} \quad (7)$$

K is the number of sections and G is the constant.

Given that our obtained filter transfer function is of the seventh row, it breaks on three second order and one first order transfer functions.

MATLAB code $[sos,g]=tf2sos(b,a)$; is used to get the polynomial coefficients of first and second order filter transfer functions. These coefficients are:

- b1 = [1.0000 1.0101]
- a1 = [1.0000 -0.8577]
- b2 = [1.0000 2.0126 1.0127]
- a2 = [1.0000 -1.6544 0.7640]
- b3 = [1.0000 1.9955 0.9956]
- a3 = [1.0000 -1.5325 0.8390]
- b4 = [1.0000 1.9818 0.9819]
- a4 = [1.0000 -1.4706 0.9418]

Constant G has the value of $G=1.76 \times 10^{-5}$. It can be noted that the coefficients of these polynomials are considerably smaller than polynomial coefficients of transfer function in the case of direct implementation. Fig. 8 shows the block diagram of digital filter cascade realization.

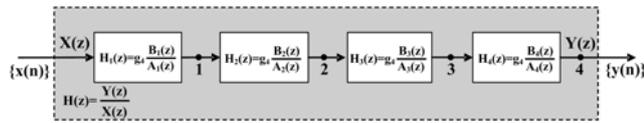


Fig. 8. Block diagram of cascade realization of digital filter transfer function

Filter structure is the cascade connection of one first order and three second order sections. Constant G is evenly distributed in all sections:

$$g_4 = \sqrt[4]{G} = 0.0648 \quad (8)$$

In cascade realization there are problems of constant G distribution among the sections as well as the sequence of the sections. We will not consider them now.

In terms of used components direct and cascade realization are practically equivalent. However, in the case that digital filter need to be hardware realized, cascade realization is more convenient, having in mind its modularity.

Based on the transfer functions $H_i(z)$, $i=1,2,3,4$ we get differential equations in time domain, which are used during the hardware implementation of each module.

Differential equation for first order transfer function is:

$$y(n) = 0.8577y(n-1) + 0.0648x(n) + 0.0655x(n-1) \quad (9)$$

From the equation we can see that for the calculation of the output sequence $\{y(n)\}$ n-th member we need to know prior calculated $y(n-1)$ member and two members of the input sequence $\{x(n)\}$. Constants that multiplies these elements are represented in the second complement, in the format [18 12]. Shown in Fig. 9 is the block diagram of the first order section hardware implementation. Section consists of data path and control unit [4], [5].

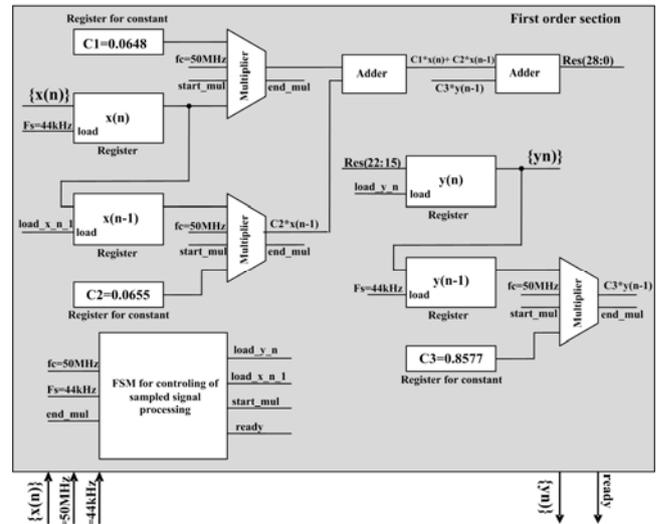


Fig. 9. First order section block diagram

The input to the first order section is sampled signal $\{x(n)\}$, which, as it is said, is a sequence of 8-bit numbers represented in the second complement. Signal $\{x(n)\}$ is the result of A/D conversion of analog signal $x(t)$ which we want to filter. $x(n)$ and $x(n-1)$ elements of the input sequence $\{x(n)\}$ are stored in 8-bit registers as well as elements of the output signal $\{y(n)\}$, $y(n)$ and $y(n-1)$. Constants C_1 , C_2 and C_3 are stored in 18-bit registers. Data path also includes adders and multipliers. Entire process of processing input signal elements is up to control unit – Finite State Machine (FSM). The adder is of the *ripple-carry* type, based on *full adders*. Multiplier has *start_mul* input signal that initiates the process of multiplication and the output signal *end_mul* which signals that the process of multiplication is over. Input signals to the first order section are also two clock signals: *clock1* with frequency of $f_c=50MHz$ which is used to synchronize the processing of sampled signal and *clock2* with frequency of $f_s=44kHz$ (sampling frequency) which synchronizes the storage in registers of elements $x(n)$ and $y(n-1)$. From Fig. 9 can also be seen that register $y(n)$ stores only 8 bits of result $Res(28:0)$. The first 15 bits of the result signal, $Res(14:0)$, are the fractional part, because the constant C_1 is in the format [18 12] and therefore the result of multiplication has 15 bits in fractional part. 6 MSB bits of the result, $Res(28:23)$, are a sign extension of 8-bit number that is stored in register $y(n)$. These 6 bits also aren't of importance.

Setting the output signal *ready* FSM indicate the end of sample $x(n)$ processing. Now, 8-bit output $y(n)$ has the valid data.

Differential equations for second order sections are:

$$y(n) = 1.6544y(n-1) - 0.764y(n-2) + 0.0648x(n) + 0.1304x(n-1) + 0.0656x(n-2) \quad (10)$$

$$y(n) = 1.5325y(n-1) - 0.839y(n-2) + 0.0648x(n) + 0.1292x(n-1) + 0.0645x(n-2) \quad (11)$$

$$y(n) = 1.4706y(n-1) - 0.9418y(n-2) + 0.0648x(n) + 0.1284x(n-1) + 0.0636x(n-2) \quad (12)$$

From the equations can be seen that for calculating the n-th member of the output sequence $\{y(n)\}$ we need to know prior calculated $y(n-1)$ and $y(n-2)$ members and three members of input sequence $\{x(n)\}$. Constants that multiplies these elements are also represented in the second complement, in the format [18 12]. Shown in Fig. 10 is the block diagram of the second order section hardware implementation.

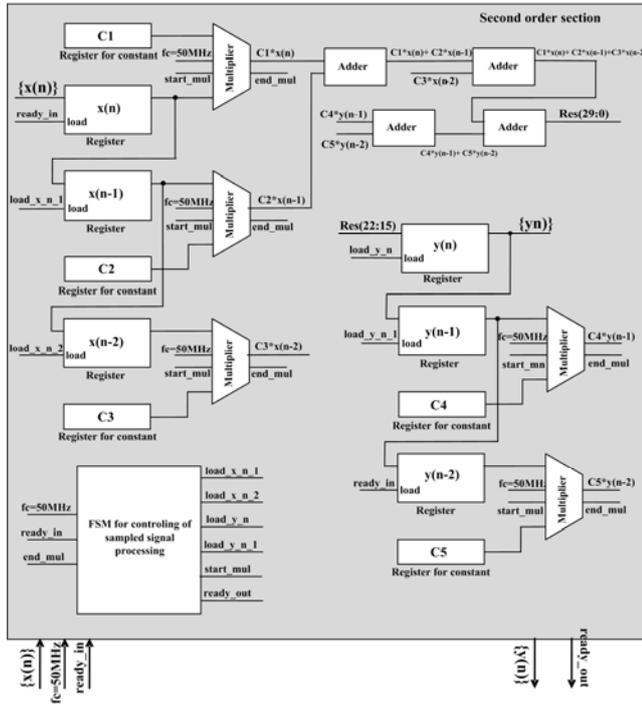


Fig. 10. Second order section block diagram

Second order section is organized as well as first order section. It also consists of data path and control unit. Unlike the first order filter section, this section has two more multipliers and two more adders as well as additional 8-bit and 18-bit registers for storing coefficients and $x(n-2)$ and $y(n-2)$ elements. For the same reasons as in first order section, register $y(n)$ stores only 8 bits of the result $Res(29:0)$.

Finally, Fig. 11 shows the block diagram of low-pass IIR digital filter cascade realization.

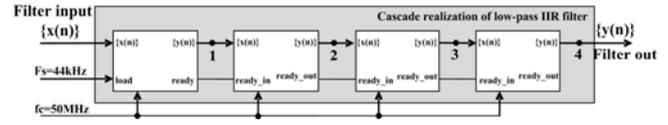


Fig. 11. Cascade realization of digital low-pass IIR filter

Each section of digital filter hardware realisation is described and simulated in Active-HDL software package, using schematic and text editor and VHDL language. Validation of filter is made for input signal:

$$x(t) = 2 \sin(2\pi f_1 t) + 2 \sin(2\pi f_2 t) \quad (13)$$

The input signal is sum of two sinusoids with frequencies $f_1=2kHz$ and $f_2=9kHz$. Amplitude of both sinusoids are $A=2$. Considering that the filter bandwidth is $f_p=5kHz$ we expect that sinusoid with frequency f_2 should be suppressed in the filtered output signal. Filtered signal therefore need to be sinusoid with frequency f_1 and amplitude $A=2$.

Numbers 1, 2, 3 and 4 in Fig. 11 indicate the outputs from the first, second, third and fourth filter section, respectively. Filter input $\{x(n)\}$ is the sequence of 8-bit numbers represented in the second complement. This sequence is obtained on the output of A/D converter which converts analog voltages within a range of $[-5, 5V]$ to 8-bit digital number presented in the second complement. A/D converter sampling frequency is $F_s=44kHz$.

Filter simulation is also performed in MATLAB. MATLAB code for this simulation is shown in Fig. 12.

```

t=0:1/44000:0.0015; % time vector [0, 1.5ms], delta(t)=1/Fs
f1=2000; % passband frequency component of y signal [Hz]
f2=9000; % stopband frequency component of y signal [Hz]
y=2*sin(2*pi*f1*t)+2*sin(2*pi*f2*t); % signal for filtering
[sos,g]=tf2sos(b,a); % cascade realization of filter transfer function
g4=g^0.25; % constant G is evenly distributed by sections
yy1=filter(a1*g4,b1,y); % filtering with first section
yy2=filter(a2*g4,b2,yy1); % filtering with second section
yy3=filter(a3*g4,b3,yy2); % filtering with third section
yy=filter(a4*g4,b4,yy3); % filtering with fourth section
    
```

Fig. 12. MATLAB code for signal filtering

At least, Figs. 13 and 14 show comparative results of filter simulation. Fig. 13 shows simulation results using Active-HDL and Fig. 14 simulation results using MATLAB software package.

As can be seen, there is some degree of mutual match of simulation result. Mutual match would be even better if we reduce digital noise ie. if elements of input y sequence $\{x(n)\}$ were presented with more than 8 bits. That means the use of better A/D converter for analog-to-digital signal conversion. Also, mutual match would be better if the constants were presented in a larger than [18 12] format.

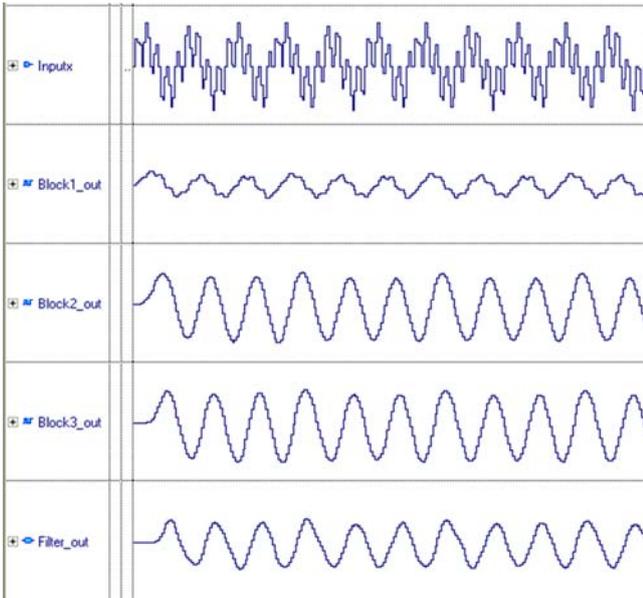


Fig. 13. Filter simulation in Active-HDL software package

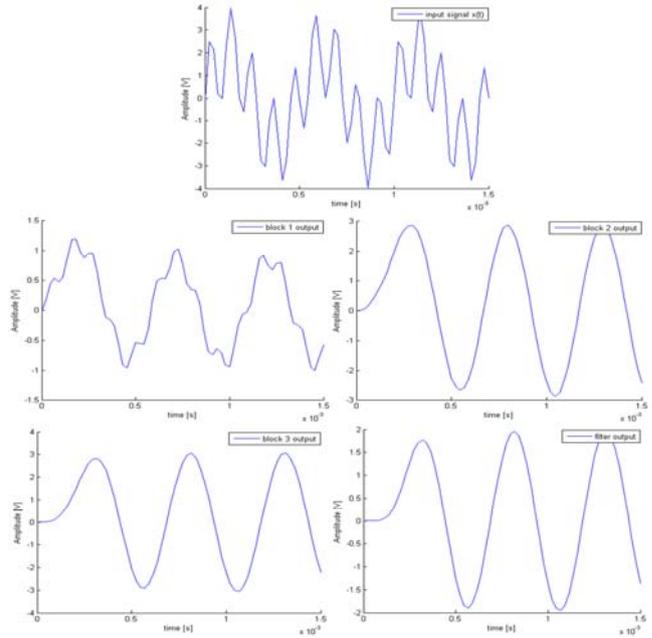


Fig. 14. Filter simulation in MATLAB software package

IV. CONCLUSION

HDL description of digital low-pass IIR filter can be implemented in any programmable logic component (CPLD or FPGA) that has the appropriate number of programmable logic blocks. Since each section consists of data path and control unit it is possible to perform section optimization by the criteria of size, speed, power consumption, etc.

REFERENCES

- [1] Stojanović, V., “*Diskretne mreže i procesiranje signala*”, Elektronski fakultet, Niš, 2004.
- [2] Winder, S., “*Analog and Digital Filter Design*”, Supertex Inc., Ipswich, 2002.
- [3] Ingle, V., Proakis, J., “*Digital signal processing using MATLAB, 2e*”, Thomson, Boston, 2007.
- [4] Pedroni, V., “*Circuit Design with VHDL*”, MIT Press, London, 2004.
- [5] Ronald, T., Neal, W., “*Digital Systems-Principles and applications*”, Prentice Hall, New Jersey, 2007.

Galvanotechnical Manufacture of Parts of Electrical Components using Pulse-reversed current

Zoran Stević, Mirjana Rajčić-Vujasinović, and Dragan Topisirović

Abstract- Electrochemical deposition of gold, silver and similar precious and non-ferrous metals is of great importance for quality manufacture of fine electrical components, for example of integrated circuit pins. This paper presents procedure of modeling of such galvanic plating processes. Successful modeling enables automation of layer forming process and its control using PC-based system and appropriate software. Also, the advantage of pulse-reversed regime which is related to forming a uniform thickness of layer, radiance, adhesion and substrate sharp edges tracking is shown.

Keywords - Pulse current, Pulse-reverse current, Plating, Au, Ag, Cu, ORCAD, Simulation, Plating bath

I. INTRODUCTION

Galvanic plating of electrical components often uses cyanide electrolytes because they provide the best current distribution on the surface of an object on which the layer is deposited to, and so in that way its uniform thickness [1], [2], [3]. From the quality aspect of working and living environment those electrolytes are, however, very harmful. It is proved that there is a possibility that quality layers on complex form objects are obtained also from non-complex electrolytes if pulse and reverse currents are applied. Pulse and reverse current regimes are used for obtaining layers with better characteristics in aspects of radiance, adhesion, edges tracking and uniform deposit distribution on complex form objects, in opposite to layers which are obtained using constant current [4], [5], [6]. Those are the reasons why in contemporary industrial plants, especially galvanic ones, very often current sources with fast and simple changes of current direction and intensity are needed, or even with a desirable current shape in time.

The development of electrical and computer equipment opened the possibilities of obtaining an arbitrary intensity or current shape in time, with complete process automatization and introduction of necessary feedbacks for system breakdowns elimination.

Here, the application of one such system is described – bipolar current source with a possibility of assigning

Zoran Stević and Mirjana Rajčić-Vujasinović are with the Technical Faculty of Bor, University of Belgrade, Vojske Jugoslavije 12, 19210 Bor, Serbia, E-mail: zstevic@tf.bor.ac.rs, mrajcic@tf.bor.ac.rs.

Dragan Topisirović is from Regional centre for talents Niš, Serbia, 18 000 Nis, 9. Brigade 10, E-mail:center@medianis.net.

time intervals and current intensities in one or in the other direction, for electrochemical deposition of silver and gold on electronic components of small dimensions. Silver is applied as a layer on electrical contacts and conductors, especially in high-frequency electronics, or as a protection from corrosion of devices in chemical industry because of its high electrical conducting characteristics, mirror radiance and chemical stability both in alkali as well as in solutions of majority of organic acids.

II. EXPERIMENTAL SECTION

A. Experimental technique

The experiments are performed using a pulse-reverse current source up to 50 A, so in pulse period so in a reverse period, as showed in Figure 1. Complete system is based on Pentium IV personal computer and LabVIEW software platform [7], [8]. Interface, current amplifier and application software are the result of authors' own development [9], [11]. Device for generating of pulse-reversed current, which is completely computerized, tracks galvanic cell response by registering changes of cell voltage in time during the whole process and records them on assigned address under a name which is defined by user. Then, data can be processed and showed in an arbitrary way.



Fig.1. Computer controlled pulse-reversed current source

B. Experimental results

Preliminary experiments of galvanic deposition of silver and gold are performed with pulse and pulse-reversed current [11], [13]. These experiments showed that the best results are achieved when using pulse current when radiance of obtained layer is set as a criterion. Further experiments

have confirmed that longer pauses also provide better results, so the regime of pulse and pause ratio 5s : 1s is adopted as optimal.

Diagram on Figure 2 shows first 10 periods of galvanization cell voltage response, which correspond to the experiment with current density of 100 A/m^2 with a duration of pulse 5s and pause 1s. The plate thickness of $10 \text{ }\mu\text{m}$ is obtained by such regime in a classic electrolytic cell. This layer cross section look is showed in Figure 3. It can be seen that the layer is of very uniform thickness and very compact.

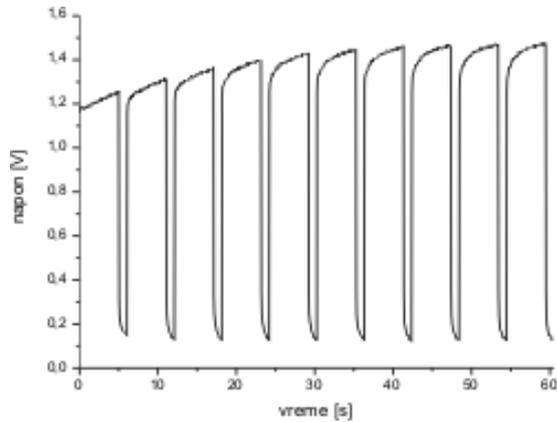


Fig.2. Cell response during deposition of silver by current of density 100 A/m^2 with pulse duration 5s and pause 1s

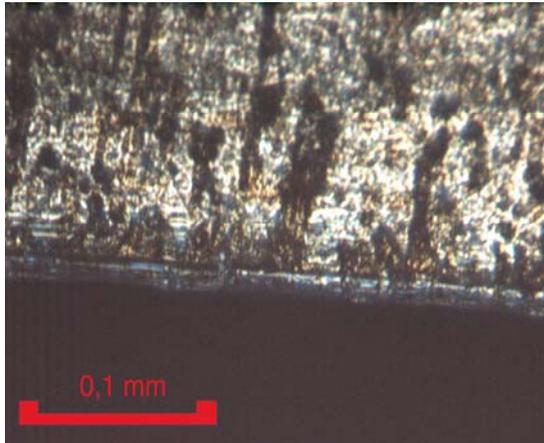


Fig.3. Cross section of a silver layer obtained by a pulse current; current pulse density = 100 A/m^2 , $t_p = 5\text{s}$, $t_0 = 1\text{s}$

Figure 4. gives galvanic bath voltage change in time during the gilding process using pulse current density of 120 A/m^2 with a pulse duration 7s and a pause duration 1s. For clearness, here are given only four pulse-pause cycles, while all data are stored in suitable file and they can be used for presenting in arbitrary way.

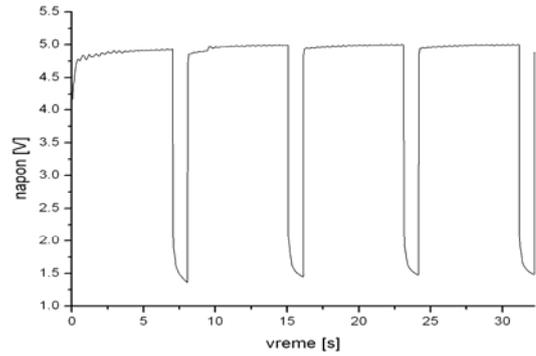
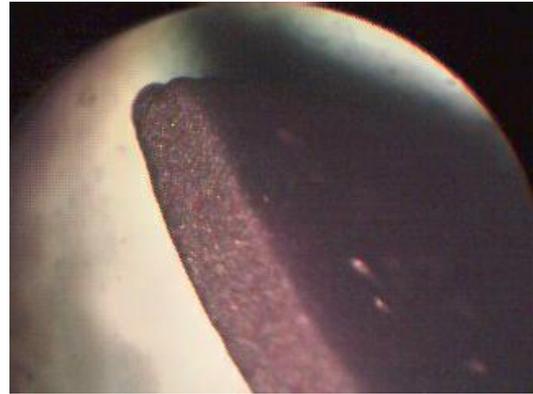
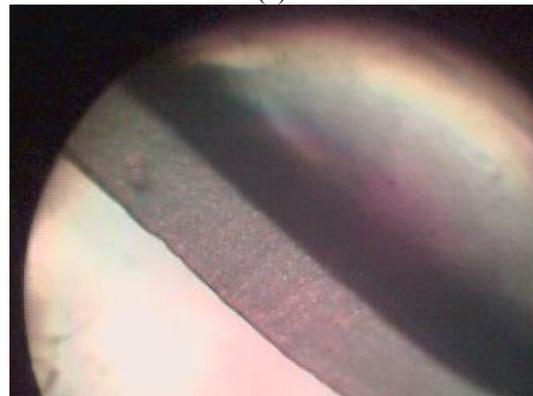


Fig.4. Voltage change during deposition of gold in Hull cell using current density of 120 A/m^2 with a pulse duration 7s and a pause duration 1s

The advantages of well-led pulse-reversed regime are also shown in the case of copper layer deposition on a steel substrate. Figure 5. gives a shot of a sample whose sharp edges are impossible to track by a direct galvanization. Much better result is accomplished using pulse current.



(a)



(b)

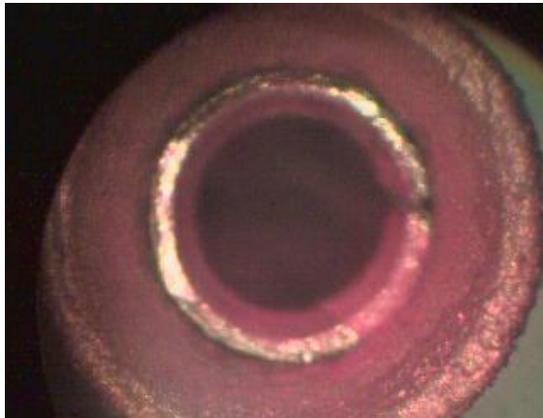
Fig. 5. Edge effect: (a) using direct current; (b) using reversing current.

Need for forming quality layer represents very important demand in manufacturing process and especially testing VLSI circuits. The most important element when testing a component is binding a component which testing, so-called DUT, to a testing device. DUT configuration,

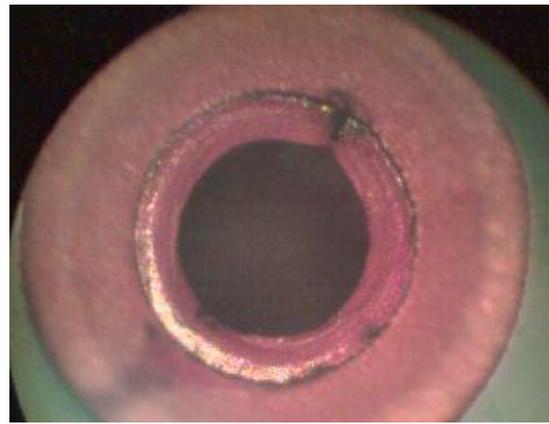
architecture and the ability of testing device and testing role determine an optimal use approach [14]. Therefore, it is worked on projecting and manufacture of persistent adapters, so-called pin cards, which are being projected for testing of several different elements and represent a complex totality, which in a phase of projecting and manufacture in fabric could be very hard work. Characteristics of every reliable adapter are: transparency, simplicity, persistency and safety. These features are normally also related to the purpose of the adapter.

The saturation of DUT (device under test) and a signal distribution towards input and output pins are guided through interface adapter. Required power for testing of VLSI circuits can be either in miliwatts [mW] or for testing of PC cards in wats [W], and the signals on pins reach frequencies of order of several GHz. Functional testing requires such grounds and connectors which provide from a few dozens to several hundreds of contacts towards component which testing, that is for saturating of DUT. Pins and connectors must have their own duration period of time and reliability during their exploitation. Therefore, the forming procedure of quality layers in manufacturing process is of great importance, in concrete case in forming pins of integrated circuits.

A copper-plated pin of a ground of an integrated circuit is shown in Figure 6. The layer obtained using pulse-reversed current is of much better quality (example on Figure 6^b) in comparison to the layer obtained using direct current of same density and the same electrolyte (Figure 6^a).



(a)



(b)

Fig. 6. Copper deposit obtained on relief surface: (a) using direct current; (b) using reversing current

C. Process modeling

Electrochemical processes which are observed belong to class of reactions which are successfully modeled by an equivalent electrical circuit shown in Figure 7. In this paper, the parameters of model obtained for reaction of gold deposition using pulse current density of 120 A/m^2 with pulse duration 7s and pause duration 1s are determined. In this case, voltage response of galvanic bath is shown in Figure 4. Values of model parameters which are used for successful simulation of experimentally obtained diagram in Figure 4., are: $R_0 = 0,1 \ \Omega$, $R_1 = 11 \ \Omega$, $R_2 = 12 \ \Omega$, $R_3 = 5,15 \ \Omega$, $C_1 = 6 \text{ mF}$, $C_2 = 245 \text{ mF}$.

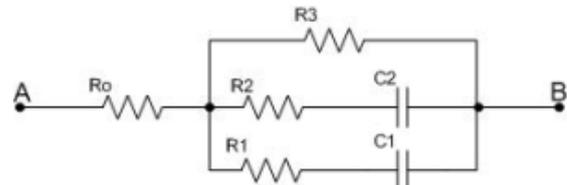


Fig.7.Equivalent electrical circuit for observed class of electrochemical reactions

Simulation is performed in program package ORCAD, and the simulation result is shown in Figure 8. Shown diagram gives very good agreement to a real curve from Figure 4., which means that chosen model very good describes observed process in quasi-stationary regime. Qualitative analysis provided in that way allows determining optimal length of pulse and pause duration, without performing a real experiment, but on the basis of assigned criterions, which could in this case be stationary state reaching in a pause period. Using appropriate software, a system could be realized which would automatically regulate pulse and pause duration for every real system, following earlier defined criterions.

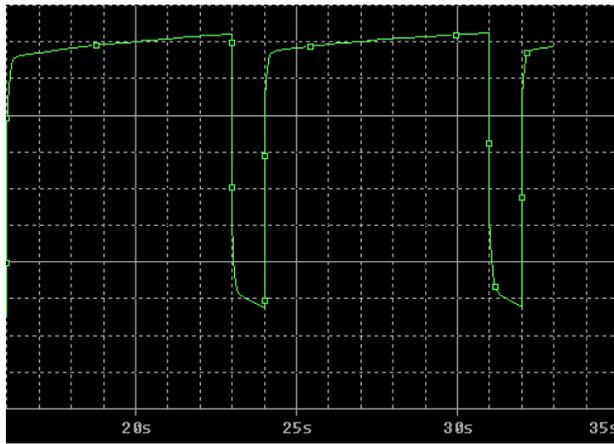


Fig.8. The simulation result is performed in programme package ORCAD

III. CONCLUSION

Quality layers of gold, silver and other precious and non-ferrous metals which are of great importance in industry of fine electrical components, can be obtained by using pulse current with optimal parameters. It is possible to reach and regulate those parameters using computerized systems, if an adequate model of process which takes place in galvanic bath is assumed. This paper describes modeling of these processes with calculated parameters and also shows successfully performed simulation of galvanic cell voltage response. The advantage of pulse-reversed regimes is experimentally proved when they give uniform layer thickness, radiance, adhesion and substrate sharp edges tracking using examples of silver, gold and copper layers.

REFERENCES

- [1] Đorđević, S., „Metalne prevlake”, Beograd, 1970.
- [2] Кудрявцев, Н.Т., Электролитические покрытия металлами, „Химия”, Москва, 1979
- [3] Нечаев, Е.А., Бек, Р.Ю., Электрохимия, 2,1, 1966, 150-154
- [4] Chernenko, V.I., Litovchenko, K.I., Papanova, I.I., "Progressivnie impulsnie i peremenotokovie rezhimi elektroliza", 1988, Naukova dumka, Kiev
- [5] Maksimović, M. D., "Pulsirajuća i reversna struja u

- galvanotehnici", YUCORR 2000, Knjiga radova, 131
- [6] Popov, K.I, Maksimović., M.D, "Theory of the Effect of Electrodeposition at a Periodically Changing Rate on the Morphology of Metal Deposits" From: Modern Aspects of Electrochemistry, 10, Ed. by B.E. Conway, J. O'M Bockris and R.E. White, 1989, Plenum Publishing Corporation, 193
- [7] National Instruments LabVIEW; *Analysis concepts*; NI Corporation, 2007.
- [8] Elliot, C.; Vijayakumar, V.; Zink, W.; Hansen, R. National Instruments LabVIEW: "A Programming Environment for Laboratory Automation and Measurement", *J. Assoc. Lab. Automat.* 2007, 12, 17-24.
- [9] Dragan Milivojević, Zoran Stević and Mirjana Rajčić-Vujasinović, "Hardware and Software of a Bipolar Current Source Controlled by PC", *Sensors* 8 2008, 1977- 1983
- [10] Zoran Stević, Mirjana Rajčić-Vujasinović, Vesna Fajnišević, Ljubiša Stamenković, "Pulsno-reversni izvor napajanja za poluindustrijsko galvansko postrojenje", IT konferencija: Saradnja istraživača različitih struka na području korozije i zaštite materijala, Tara, 2005, Knjiga radova, 271-275
- [11] Mirjana Rajčić-Vujasinović, Zoran Stević, Aleksandra Milkovski, "Hardver i softver inteligentnog izvora napajanja za primenu u galvanotehnici", INFOTEH JAHORINA 2005, Istočno Sarajevo (2005)
- [12] M. Rajčić-Vujasinović, Z. Stević, V. Fajnišević, J. Kocev, V. Rankić, B. Milenković, Lj. Stamenković, "Prevlake srebra dobijene pulsним strujama", VIII YUCORR Korozija i zaštita materijala u industriji i građevinarstvu, Tara 2006, Knjiga radova, pp.318-321
- [13] M. Rajčić-Vujasinović, Z. Stević, J. Kocev, V. Rankić, B. Milenković, Lj. Stamenković, "Prevlake zlata dobijene pulsним strujama", VIII YUCORR Korozija i zaštita materijala u industriji i građevinarstvu, Tara 2006, Knjiga radova, pp.322-325
- [14] D. Topisirović, "Testiranje integriranih kola-problem sprege sistema i komponente", SYM-OP-IS 1998, Zbornik radova, Herceg Novi, 21-24 septembar 1998, pp.929-932.

Parallelizing Electronic Circuit Simulation on Multicore Computer Cluster

Bojan Anđelković, Marko Dimitrijević,
Miona Andrejević Stošović and Vančo Litovski

Abstract - This paper presents an algorithm for parallelization of transistor-level analog circuit simulation. Basic information regarding modern simulation, leading to the need for parallel simulation is presented. Implementation of a new algorithm based on parallel equation formulation in a mixed-mode simulator is explained. Simulation performances are considered for two computer cluster configurations: one based on nodes with single core processors and the other based on nodes with multicore processors.

Keywords – Parallel circuit simulation, Distributed simulation.

I. INTRODUCTION

The simulation process of modern complex integrated electronic circuits may be characterized as memory and computationally intensive, and algorithmically complex. These properties pertain to the fact that a large number of ordinary (and, potentially, partial) nonlinear differential equations have to be solved for long running excitations. In addition, a huge amount of data may be created in a single simulation run, which needs to be processed and interpreted. Because of all these reasons simulation runtimes are very long. Having in mind that every design needs many simulation runs of the same design in order to get optimal solutions and satisfy the design requirements, it is obvious that long simulation runtimes lead to a slow design process. One possibility to reduce these runtimes is to parallelize the simulation algorithm and use computer clusters or multicore processors to execute simulations. In this approach complex calculations necessary during the simulation process can be distributed over different workstations/processors and performed simultaneously.

Over last decade, as personal computers performance has increased and prices have fallen steeply, both for the PCs themselves and the network hardware necessary to connect them, dedicated clusters of PC workstations have provided significant computing power at low cost.

Bojan Anđelković is with Fujitsu Microelectronics Europe, Germany. E-mail: abojan@gmail.com

Marko Dimitrijević, Miona Andrejević Stošović and Vančo Litovski are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia. E-mail: (marko.dimitrijevic, miona.andrejevic, vanco.litovski)@elfak.ni.ac.rs.

There are several parallel simulators of electronic circuits that have been developed recently, such as Xyce [1], Titan [2] and SEAMS [3]. Titan and Xyce are parallel transistor level simulators that use SPICE as modeling language. Both simulators implement complex partitioning algorithms to split the circuit description and distribute generated partitions to different workstations/processors. These partitions are then simulated in parallel. Appropriate synchronization protocols should be applied to exchange necessary simulation data between the circuit partitions. The main goal of these parallel algorithms is to minimize communication between the workstations/processors and achieve their equal load. SEAMS is a VHDL-AMS simulator that implements parallel digital simulation, while parallel mixed-signal simulation is under the development. A broad survey of various parallel simulator implementations and algorithms can be found in [4].

This paper presents the concept of development a parallel transistor level simulator of electronic circuits that executes on a computer cluster using MPI. Parallelization of equation formulation for nonlinear analog circuit elements is implemented in order to reduce simulation time.

II. PARALLEL EQUATION FORMULATION

In order to simulate complex mixed-signal electronic circuits at transistor level, they have to be modeled using algebraic equations and nonlinear Ordinary Differential Equations (ODE). ODEs are discretized in order to create sets of nonlinear algebraic equations. This generates a potentially large system of equations to be solved at a large number of time instants depending on the properties of the system under simulation and the stimulus signals. The system of nonlinear equations is solved iteratively with the help of linearization i.e. by application of Newton methods.

The algorithm for simulation of nonlinear dynamic electronic circuits in time domain is shown in listing 1 [5]. As it can be seen, at each iteration and at every time instant the matrix entries of the system of linear equations have to be recalculated. These entries are derivatives of the nonlinear equations and are computed within separate subroutines. Having in mind the number of matrix entries, the number of iterations and the number of time instants, it is necessary to provide an immense computational effort. It has been shown that even for small systems, equation

formulation takes more computational time than equation solution. Therefore, the calculation of matrix entries and equation formulation for non-linear circuit elements can be parallelized. The part of the simulation algorithm that can perform in parallel is highlighted by a rectangle in listing 1.

If we consider the circuit matrix as a sum of several matrices the number of which is equal to the number of processors implemented, we may create the whole matrix by creating its parts and then by summing them. That is illustrated in Fig. 1. There is no specific criterion for allocation of the circuit elements to specific processor (or submatrix). Simply the total number of non-linear circuit elements is divided by the number of processors and the list of elements is divided to equal parts and partitions are created. Such parallelization of the simulation algorithm is a new approach different from already developed solutions. It requires neither a sophisticated circuit and task partitioning algorithm nor synchronization protocols between these partitions, so it is easy to implement on a computer cluster using MPI routines.

```

generate node voltages and specified branch
currents  $x^0 = [(v^0)^T (i^0)^T]^T$ ;
choose  $h$ ;
 $n = 0$ ;

while( $t < T$ ) { /* time loop */
     $m = 0$ ;
    predict  $x^{n+1,0}$ ;
    until convergence { /* iterative loop */
        generate discretized models;
        generate linearized models;
        formulate system of linear equations;
        solve the system and find  $x^{n+1,m+1}$ ;
         $m++$ ;
    }
     $t = t + h$ ;
     $n++$ ;
}
    
```

Listing 1. The simulation algorithm for nonlinear dynamic circuits in time domain

The generation of matrix entries for constant and linear dynamic elements is not performed in parallel, since these calculations may be performed outside of the iterative loop. Moreover, the matrix contributions for constant elements are calculated only once outside time and iterative loops, while entries for linear time dependent elements are calculated at every time instant outside iterative loop. All this reduces the overall time necessary for equation formulation. When parallel generation of matrix entries for various nonlinear elements is finished, the complete circuit matrix is formed (Fig. 1). Then system of linear equations is solved. That task can also be parallelized which should lead to further reduction in simulation time.

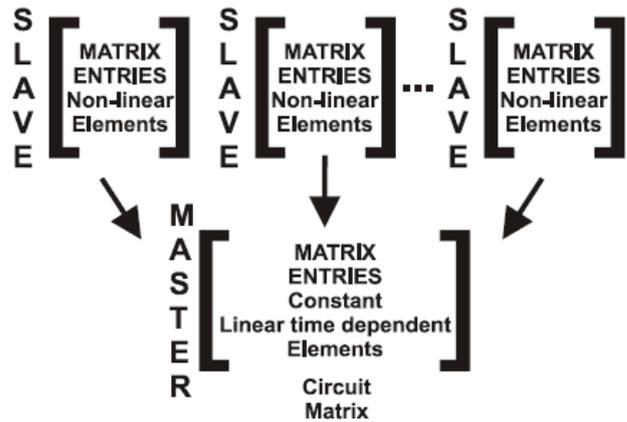


Fig. 1. Parallelization of equation formulation

III. PARALLEL SIMULATOR IMPLEMENTATION

The presented parallelization of equation formulation process is implemented in the simulator Alecsis [6]. It is a mixed-signal and mixed-domain simulator with proprietary hardware description language AleC++ [7] capable for modeling and simulation of complex systems containing different kinds of devices and subsystems [8]. The developed simulator with parallel simulation capability is called pAlecsis (Parallel Analog and Logic Electronic Circuits Simulation System).

The implementation of parallelization in the pAlecsis simulator on a computer cluster is shown in Fig. 2. Parallel equation formulation is implemented using one of the most common of parallel algorithm prototypes, master-slave algorithm [9]. In this algorithm calculation of matrix contributions for non-linear circuit elements at each time instant and iteration is distributed to multiple slave processes and they are calculated simultaneously. At the same time master process calculates matrix entries for specific number of nonlinear elements as well as for constant and linear time dependent elements. Master and slave processes execute on different cluster nodes (PC workstations).

Since multiple cluster nodes calculate contributions for different elements in parallel, the time necessary for equation formulation decreases.

In order to minimize communication between cluster nodes, appropriate data structures for all elements of the circuit are generated on all nodes simultaneously during compilation of the AleC++ model. In that way all cluster nodes have the information necessary to generate matrix contributions for all elements. Each node of the cluster performs equation formulation and calculation of matrix entries for equal number of nonlinear circuit elements. When entries for all elements on one slave are generated, they are sent to the master node using appropriate MPI routines (Fig. 2).

When the master node receives matrix entries from all slaves, it flushes them to the system matrix and performs one simulation step. In order to enable calculation of matrix

entries on slave nodes, the master node should send to the slaves' vectors of solutions of the system of equations for the two past time instants and previous iteration (denoted with $vp1$, $vp2$ and vi respectively in Fig. 2). Appropriate MPI routines for transferring data are used to send and receive these vectors.

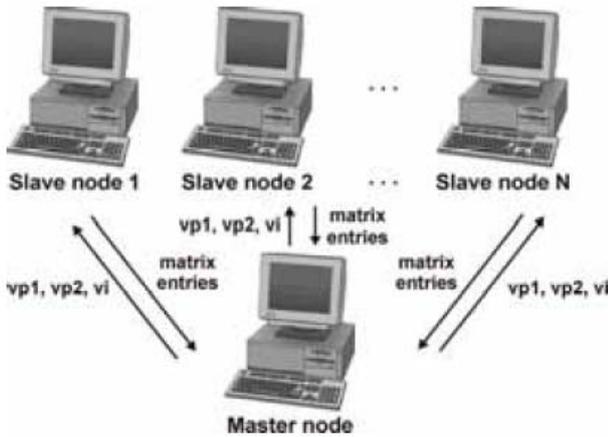


Fig. 2. Implementation of the pAlecsis simulator on a computer cluster

IV. PARALLEL SIMULATION PERFORMANCES

Sequential simulation algorithms executing on a single workstation are tested for correctness usually by only seeing whether they give the right result. For parallel programs, that is not enough, but one wishes to reduce the simulation time. Therefore, measuring of simulation time is part of testing the parallel simulator to see whether it performs as intended. Usually performances of the parallel simulator are specified as speedup. If parallel simulation executes on N single processor cluster nodes, speedup is normally defined as [9]:

$$Speedup = \frac{\text{Simulation time on 1 node}}{\text{Simulation time on } N \text{ nodes}} \quad (1)$$

Implemented parallel simulation algorithm reduces simulation time for bigger circuits when time necessary to calculate matrix entries for all elements at every time instant and every iteration exceeds time necessary to calculate matrix entries on slave nodes and send them to master node over the interconnecting network. For such circuits the parallel simulation on the cluster is faster than the simulation on a single processor workstation.

In order to determine the size of circuits in number of transistors for which there is a speedup in simulation on a cluster with two nodes, parallel simulations using the presented algorithm were performed on circuits consisting

of various number of MOSFETs. These circuits are generated by successive replication of bilinear SC filter circuit with MOSFET operational amplifiers. Then speedup is calculated according to (1).

TABLE I
BEOWULF CLUSTER STRUCTURE

| Component | Specification |
|-------------|---|
| Master node | PC Pentium IV, 2.4GHz, 1GB RAM, 240GB HDD |
| Slave nodes | 8× PC Pentium IV, 2.4GHz, 512MB RAM, 80GB HDD |
| LAN | 1Gbit Ethernet |

TABLE II
SPEEDUP OF PARALLEL SIMULATION IN PALECSIS FOR BEOWULF CLUSTER STRUCTURE

| Number of MOSFETS | Simulation Speedup (2 cluster nodes) |
|-------------------|--------------------------------------|
| 740 | 1.1 |
| 1480 | 1.5 |

TABLE III
NEW COMPUTER CLUSTER STRUCTURE

| Component | Specification |
|-----------|--|
| | 8 × Two quad-core Intel Xeon E5420, 2.5GHz, 4GB RAM, 250GB HDD |
| | 1.4TB RAID5 network attached storage |
| LAN | dual 1Gbit Ethernet |

TABLE IV
SPEEDUP OF PARALLEL SIMULATION IN PALECSIS FOR NEW COMPUTER CLUSTER STRUCTURE

| Number of MOSFETS | Simulation Speedup |
|-------------------|--------------------|
| 740 | 0.9 |
| 1480 | 1 |
| 1700 | 1.05 |

In this paper simulation results from two different computer clusters are compared. The first one is Beowulf cluster whose structure is given in Table 1. The generated simulation results are given in Table 2 [10].

New computer cluster structure is given in Table 3. This structure consists of 64 processors, and it was expected to give much better results than the first structure. The generated simulation results are given in Table 4. As it

can be seen there is no speedup for 740 and 1480 transistors, and there is a slight speedup for 1700 transistors. Total time simulation time is about 30% less. Given results are for simulations performed on 2 and 3 cluster nodes. For more cluster nodes simulation time is even greater.

VI. CONCLUSIONS

In this paper we compared simulation time results for two different computer clusters. We may conclude that clusters with better performances do not always give better results, because there is a problem in communication among cluster nodes. It is obvious here that communication time can exceed time necessary to calculate matrix entries on slave nodes. Our next step is to perform simulations on more complex circuit, because we then expect much better results.

REFERENCES

- [1] <http://www.cs.sandia.gov/xyce/>
- [2] Fröhlich, N., Riess, B. M., Wever, U., Zheng, Q., *A New Approach for Parallel Simulation of VLSI-Circuits on a Transistor Level*, IEEE Transactions on Circuits and Systems, Part I, Proc. Int. Conference on Parallel and Distributed Processing Techniques and Applications, pp. 601-613, Vol. 45, No. 6, June 1998.
- [3] Martin, D. E., Radhakrishnan, R., Rao, D., Chetlur, M., Subramani, K., Wilsey, P., *Analysis and Simulation of Mixed-Technology VLSI Systems*, Journal of parallel and distributed computing, vol. 62, No 3, pp. 468-493, 2002.
- [4] Savić, M., Anđelković, B., Litovski, V., *Parallel Mixed-Mode Simulation – Preliminary Study*, Proc. INDEL 2004, Banja Luka, pp. 76-79, 2004.
- [5] Litovski, V., Zwolinski, M., *VLSI Circuit Simulation and Optimization*, Chapman and Hall, London, 1997.
- [6] Mrčarica, Ž., et al., *Alecsis 2.3, the simulator for circuits and systems. User's Manual*, Laboratory for Electronic Design Automation, Faculty of Electronic Engineering, University of Niš, Yugoslavia, LEDA-1/1998, <http://leda.elfak.ni.ac.rs/projects/Alecsis/alecsis.htm>
- [7] Litovski, V., Maksimović, D., Mrčarica, Ž., *Mixed-Signal Modeling with AleC++: Specific Features of the HDL*, Simulation Practice and Theory 8, pp. 433-449, 2001.
- [8] Mrčarica, Ž., Ilić, T., Glozić, D., Litovski, V., Detter, H., *Mechatronic Simulation Using Alecsis: Anatomy of the Simulator*, Proc. Eurosim'95, Vienna, Austria, pp. 651-656, 1995.
- [9] Gropp, W., Lusk, E., and Skjellum, A., *Using MPI: Portable Parallel programming with the Message-Passing Interface*, second edition, MIT Press, 1999.
- [10] Anđelković, B., Litovski, V., *Parallel Transistor Level Simulation based on Parallel Equation Formulation implemented on a Beowulf Cluster*, Simulation News Europe 17/3-4, pp. 55-58, December 2007.

High Level Simulator of Spatial to Auditory Mapping System for Blind and Visually Impaired

Miloš Petković and Goran S. Đorđević

Abstract – In this paper a high level simulator for conceptual testing of aid system for blind and visually impaired people is described. System converts spatial map of near by objects into corresponding auditory map for a user to hear.

Keywords - High Level Simulator, MATLAB, Blind Aid System.

I. INTRODUCTION

In order to help blind and visually impaired people to move more freely and independently a system for converting spatial map into corresponding auditory map was considered. System comprises of two functionally independent parts as shown in Fig. 1. The first part of the system creates spatial map of surrounding objects by using some of 2D or 3D ranging and positioning methods/systems. The second part of the system converts spatial map into corresponding audio signal that contains enough information about positions of nearby objects but in the same time doesn't confuse a user with too much of data for interpreting. Since finding a fine balance between these opposing objectives is not trivial the goal of this paper is to introduce a simulator for testing different ideas. MATLAB was used as designing platform for this purpose [1].

The paper is organized as follows. The subsequent section describes different types of spatial to auditorial conversion. The third section explains the functionality of the simulator, while the fourth section presents the implemented algorithm. The simulation results are given in the fifth section.

II. AUDITORY MAP

Finding a right way to convert spatial information of surroundings into easy to interpret audio signal which creates vague but full picture of nearby objects in user's mind is rather empirical. Nevertheless, the good guiding line is human ear characteristic [2, 3] and Head-Related Transfer Function (HRTF) [4, 5, 6, 7]. Normally, people can guess the position of a sound source by comparing

what they hear on the left and right ear. The difference can occur due to slight time delay because sound travels a little longer to one ear than another. This is called *interaural time difference* – ITD. For higher frequency sounds the human head acts as an obstacle which causes intensity difference between sounds heard on left and right ear. This is called *interaural intensity difference* – IID. IID differs significantly from person to person since it depends on shape of outer ear, head and torso. By ITD and IID people can guess the azimuth of the sound source. ITD and IID help to determine the distance to the source although the overall intensity of the heard sound also counts. The sound intensity is inversely proportional to the square of distance from the source. Therefore, one tends to guess how far the source is by overall decrease of sound intensity when known the intensity of sound at the source. However this is also guessed by listening at different radiuses from the source. Since humans have two ears in-line then it would seem that determination of the elevation angle of sound source is impossible. Fortunately, the shape of outer ear helps in different frequency attenuation depending on the elevation of the source. Finally, it is important to mention that people learn to guess sound source position or better to say learn their auditory map since birth. It is also assumed that children blind from their birth have trouble with doing this as they lack visual feedback.

Generally speaking people are best at determining azimuth of sound source. Accuracy at determining distance is a little bit worse. People are pretty limited at guessing elevation and not that accurate. Having all these in mind several methods for generating audio signal for detected object was made.

The main idea for system was to create a feeling in users mind that objects around are emitting sound. To be more precise, that some points on surface of objects are producing sound. Like that some virtual *audio markers* are placed at certain points on surface of objects. The sound they produce could be one frequency sine wave - *monotone*, various frequency sine wave - *multi-tone* or even music. Every of these three has it's own advantages and disadvantages. When only one frequency sine wave is used a user can adjust the frequency that fits him the best and thus still be able to hear the rest of naturally generated sound by surroundings. A loudness control of generated sound is implied in all three mentioned cases. Since only one frequency is in use then only one object-point (audio marker) could play sound at a time. This would mean that some order in witch audio markers will play themselves

Miloš Petković is PhD student of Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: misa5ko@yahoo.com.

Goran S. Đorđević is with the Department of Control Engineering at Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: goran.s.djordjevic@elfak.ni.ac.rs

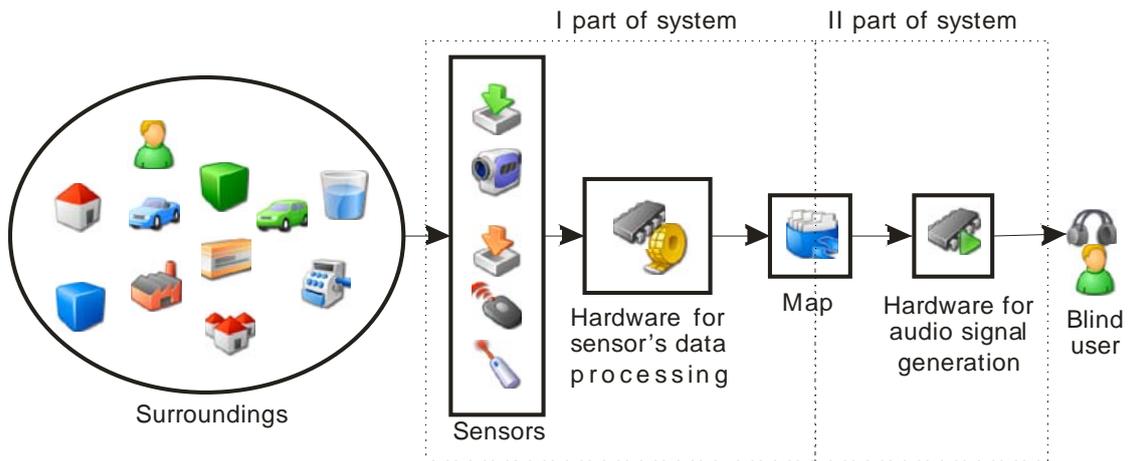


Fig. 1. Block diagram of system in use

has to exist. As said in introduction, position map of surroundings is created by processing data from 2D or 3D sensory system. 3D mapping is preferred so to eliminate possible misdetection of objects. However it is highly likely that 3D maps would have too much data for audio. What now may come as a question is which plane is going to be used in sound generation. That should be left for user to choose. Since only one plane is analyzed then the order in which audio markers are played could be defined as from right to left (anticlockwise) or from left to right (clockwise) from the users point of view. That would mean that for anticlockwise directions first would be heard objects on the most right of the user and then the ones at the left.

In order to create this azimuth position feeling principle something similar to IID can be used. By balancing sound intensity on left-right stereo audio signal user will have impression of horizontal displacement of an object. Intensity difference of left and right signal dependant on azimuth angle is given by Eqs. (1) and (2):

$$a_r = \cos(\frac{1}{2}\theta) \quad (1)$$

$$a_l = \cos(\frac{1}{2}\theta), \quad (2)$$

where θ is azimuth angle while a_r and a_l are coefficients with which are right and left audio channel signals multiplied. This is solid enough approximation of IID. As said before IID differs significantly from person to person and would need to be measured for each user. Finally by making sound intensity inversely proportional to square of distance from user to the source user could guess position of object. This eliminates completely measurements of IID and ITD.

The fact that human ear is more sensitive to frequency than intensity changes could be used in three different ways. Instead of sound intensity the frequency of sine wave can be distance dependant. For instance when an object is near the sound will be of high frequency and on the contrary when it is far frequency will be low. This could boost distance guessing accuracy. Similarly may be done with azimuth. When object is at the right corresponding

signal representation. Humans have very narrow elevation angle detection range. Therefore data only from one plane of 3D map is used for generating audio signal. It is left for system to detect collision and to warn a user that an object not visible in current plane of view is on the user's way. markers sound will be low, and vice versa. That could boost azimuth angle guessing accuracy. The second advantage of azimuth-frequency relation is that since for one azimuth angle can exist only one closest marker then different markers will have different frequencies. So all markers can be played at the same time thus increasing amount of information transmitted by sound.

Since mentioned sine wave sources can be irritating to listen music can be used as a substitute. As with mono tone sound source intensity would be proportional to the azimuth by Eqs. (1) and (2) and intensity would be inversely proportional to the square of distance.

III. SIMULATOR DESCRIPTION

The goal of this paper is to describe a simulator able to correlate spatial and auditory map.

Simultaneously it should help system designers to examine different ideas and to help potential users to adapt to the system. It has to be able to switch from one correlation method to other. Besides the simulator has to allow creating virtual environment with moving objects. Finally it must have ability to measure distances to objects so as replacement for real world environment and sensors readings.

Main functionalities of the simulator are:

- a) to be able to create, save and modify virtual maps,
- b) to simulate movements in virtual environment,
- c) to measure distances between virtual object and virtual user,
- d) to convert the position information into audio signal.

The simulator was made with only one plane or better to say with 2D map view of virtual world from above. This was done in order to speed up simulator so it could work in

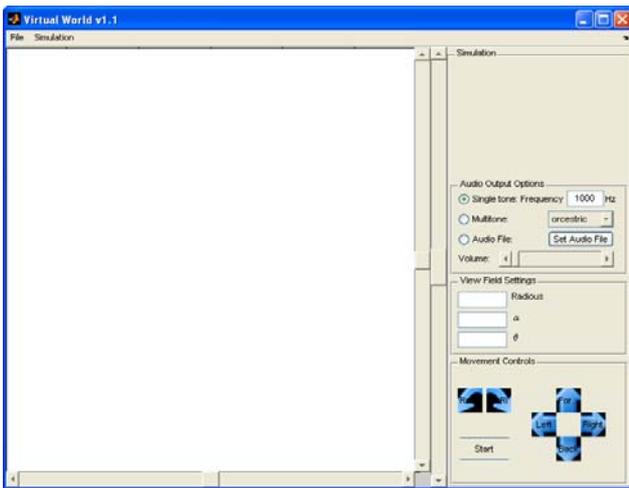


Fig. 2. Simulator's window after start up

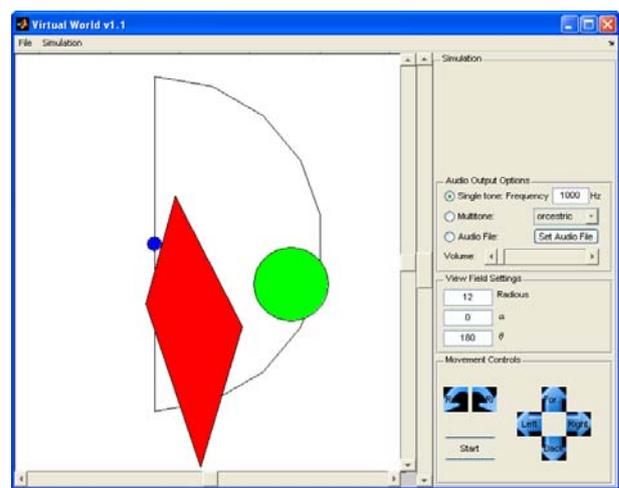


Fig. 3. Simulator's window with abstract map loaded

real time. The additional feature of possible collision detection with objects not seen in this plane is left out as currently irrelevant. In Fig 2. simulator window is shown after start up. The central plane is blank because no map has been loaded jet.

An saved map consists of array of objects with their properties, the user and its parameters and simulation time step.

There are two object types: polygon and circle. Polygon is defined with array of (x, y) pair of coordinates. There is no limit of number of pairs but the speed of simulation is inverse proportional to their number. That is why circle shape objects are included in simulator. They make creation of rounded objects easier than by using polygons. Also simulation speeds up when circle is used than multi line polygon replica of circle. These two types of objects are shown in Fig. 3 as red rectangle and green circle. Blue circle represents user and broken line circular sector is *audio filed of view* - AFV. AFV is area of space where object has to be in order for sound to be generated for it. This is very good way for a user to control amount of information converted to sound. The user can adjust AFV radius and the field of view. That means when a user is in open space radius can be bigger since there is no that much of objects around. To the contrary, when user finds himself at tight space with a lot of objects he can not only reduce the radius but also reduce the angle of view. Object detection in AFV is done with certain angular resolution. In other words there are certain number of angels at which an object presence is checked. This exist with two reasons. The first is to control amount of data converted to sound. The other one is to simulate impact of various sensors resolutions. Angular resolution is expressed in number of sub angles in AFV angle of view, not in degrees. While radius and field of view of AFV could be changed during simulation angular resolution can not. This is done because increasing a resolution increases time for completing right to left playing cycle that exist for cases when only one audio marker emits a sound. Therefore, normally real

angular resolution can be increased by narrowing angle of view.

When creating an object its significant coordinates can be inputted numerically or by clicking with a mouse pointer on the map. For a polygon significant coordinates are its endpoints' coordinates. Significant coordinates for circle are coordinates of centre and one point on its circumference.

To simulate realistic object movements segmented linear and rotational movements of objects can be defined. On every segment linear speed is constant. If segmented enough every type of movement can be achieved. For every linear segment a rotational speed can be defined as well as initial angle rotation at the beginning of each segment. Segment definition is relative, not absolute. This means that a segment path is defined by polar coordinates relative to position of object at the beginning of the segment. Polar coordinates can be inputted numerically or by clicking with mouse pointer at the map at the endpoints of the segment. When object comes to the end of its path it can be set to do one of four things: a) Stop – stop moving, its state is stop, b) Back – object continue to move backwards, reversing the linear segments sequence and their angles, but not the angles of rotational movements, c) Reverse – object continue to move backwards, reversing the linear segments sequence and their angles, as also as angles of rotational movements, d) Forward – object continue to move again as at the beginning of its path.

For moving the virtual user and changing AFV parameters an easy to detect under fingers "f", "j" and numpad's "5" keys were used as also as their neighbouring keys. This was done in order to make simulator friendly enough for blind people to use it too. That way direct users of the system in develop can give their opinion about it at an early stage of project thus reducing costs.

As it can be seen in Fig. 3 on the right side of simulator's window are audio output controls. There are three different options for sound source type as described in the previous section: single tone, multi tone and audio

file. When single tone is selected audio marker's sound source is sine wave of desired frequency which is defined by the input field adjacent to single tone radio button. Sound intensity is inverse proportional to square of distance from user to the source. Azimuth angle information is turned into intensity difference of left and right audio channels according to Eq. (1) and (2). Only one audio marker is played at a time. With multi tone audio option audio marker use various frequency sine wave as a source. There are three possible variants that are chosen from adjacent drop down list box: *orchestric*, *angular* and *radial*. When *radial* is chosen the frequency depends on distance of user to marker. So when an object is near the frequency will be high and vice versa. Azimuth angle information is turned into intensity difference of left and right audio channels according to Eq. (1) and (2). Only one audio marker is played at a time. In *angular* variant the frequency of marker depends on the azimuth angle. So for objects to the right the frequency is low and for the ones on the left it is high. Sound intensity is inverse proportional to square of distance to the source. Only one audio marker is played at a time.

Orchestric is similar to *angular*, frequency of marker depends on azimuth angle. Since for one particular azimuth angle there can be only one nearest audio marker than all audio markers will have different frequencies. This means that all audio markers can be played at the same time which is the case with this option. Sound intensity is inverse proportional to square of distance to the source.

If an audio file is to be used as a marker source it has to be firstly chosen by clicking on Set Audio File button. Only wav files are supported. Song is cut in appropriately long pieces that are successively played by markers. In order for the song to be heard constantly rather than in torrents a default song level is defined. So if for a particular azimuth angle marker doesn't exist a corresponding piece of song is played with a default volume level. As intensity of sound is inverse proportional to square of distance to the source, a default volume level was defined to be like the marker was on the outskirts of AFV. Azimuth angle information is again turned into intensity difference of left and right audio channels according to Eqs. (1) and (2). During simulation in the visual feedback screen a red line connects the virtual user and the obstacle that caused generation of audio sound. When there is no object in currently observed azimuth angle a green line stretches along from user to the border of AFV. This visual representation of currently observed azimuth was introduced to make it easier to track it.

IV. SIMULATION ALGORITHM

Simulation algorithm is given in Fig. 4 as Matlab pseudo code.

When the simulation is started copy of all objects data is made. This is done from stability reason imposed by Matlab. Next, proper graphical interface is initialized as

well as windows audio device. At the beginning of every simulation step it is being checked if user has issued a command for simulation to stop. Then in every cycle update of data is done. That implies movement commands for virtual user like forward, backward, left, right, rotate left/right, increase/decrease view angle, increase/decrease AVF radius. Now, a new position of user is calculated with assumption that it didn't collide with any static object. The following simulation loops through every static objects and checks if there were any collisions. Only static objects are considered because modelling collisions with different moving objects would consume much of simulation time. To every object type one should specify the elasticity parameter simulating the type of material. The purpose of this simulator is not to completely simulate real world interactions but to check the ability and usefulness of the system. Since only collisions with static objects are checked, then it is done by inspecting if there is any cross-sections between line which endpoints are current and new position of user and any line of object. There are two distinguished types of object so there are two slightly different methods for their collision detection. If collision is detected with a current object in loop the length of user movement is reduced so that user stops in front of the object. Next, simulator loops through every moving objects in order to update their positions. As said, the collision is not checked. First step in this loop is to calculate new position of object with assumption that it will still be on the same current segment of the movement path. Then the assumption is checked. If it is not true the rollover for the next segment must be calculated. If that was the last segment the at end command for the object is checked. As said before, one of four different actions is performed. After recalculation of new position with new segment, its validity is evaluated again. After new position is found, the object is rotated accordingly for every segment. Rotation of a circle objects is meaningless and not performed.

After all repositioning is done a loop for finding the nearest obstacles in AFV is executed. First it is checked if current object of loop cycle is visible from current view angle of user. If it is true a distance is calculated and if it is shorter than previously found one it is set as a new one. When looping is finished, the appropriate audio signal is generated according to the current settings and sent to PC audio card.

V. SIMULATION RESULTS

Several simulation with real life situation settings were performed. Two of them are shown in Figs. 5 and 6.

Fig. 5 represents every day situation of navigation through a typical living room. There is entry to the room in the bottom left corner. Pieces of furniture are tagged. Settings is completely static. Easy navigation was accomplished.

Fig. 6. presents a common situation of navigation on the street. This is rather dynamical and unknown to user

setting. Objects are tagged. An important conclusion was made after this simulation. It may happen that markers are played too slow for car moving in AFV to be detected by user. In another words since car moves too fast its audio markers never get their chance to be played. Nevertheless the potential usability of system was confirmed.

VI. CONCLUSION

In order to help blind and visually impaired people a system for converting position information of near by objects into stereo audio signal was conceptually tested by an original simulator. The simulator was built in MATLAB inheriting all its advantages and flaws.

Several conversion methods were tested and some showed more potential than others. The one that stood out was the single tone method. However this conclusion is strictly subjective. One can prefer one method more than another. Without further testing with more users nothing is final. Of course system should have flexibility for user to select the method that best works for him.

In order to make simulator more realistic for rotational movements of user, a rotational encoder can be place on rotatable chair. Readings from encoder would then be input for the simulator. This way the simulator will give a realistic figure in case when user rotates himself on the chair.

```

copy_data();
initialize_interface();
setup_audio_device();

while simulation_not_terminated
    update_data();
    %move user
    calculate_users_new_position();
    for i=1:n
        if object_is_static
            if line_defined_with_old_and_new_user's_position_and_object_line_have_cross-section
                reduce_users_move_so_there_is_no_cross-section();
            end
        end
    end
    %move objects
    for i=1:n
        if object_is_moving
            calculate_new_position_for_current_segment();
            while new_position_not_on_current_segment
                if there_is_no_next_segment
                    switch at_end_object_should
                        case 'Stop'
                            change_object_state_to_static();
                        case 'Back'
                            reverse_object's_segments_order_but_not_rotation_angles();
                        case 'Reverse'
                            reverse_object's_segments_order_and_change_rotation_directions();
                        case 'Forward'
                            go_back_to_the_first_segment_of_object's_movement_path();
                    end
                else
                    go_to_next_segment();
                end
                calculate_new_position_for_current_segment();
            end
            rotate_object_accordingly();
        end
    end
    %find distances of object in AFV
    initialize_solutions();
    for i=1:n
        calculate_distances_of_this_object_in_AFV();
    end
    %generate audio signal accordingly
    generate_sound_output_for_current_audio_output_settings();
    output_sound_on_audio_device();
end

```

Fig. 4. Simulator's algorithm written in pseudo MATLAB code

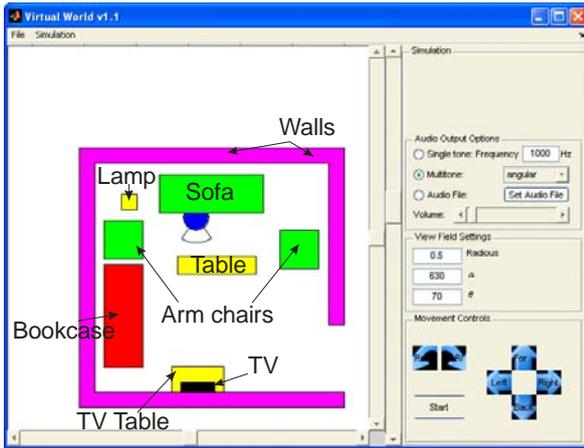


Fig. 5. Simulator's window with abstract map loaded

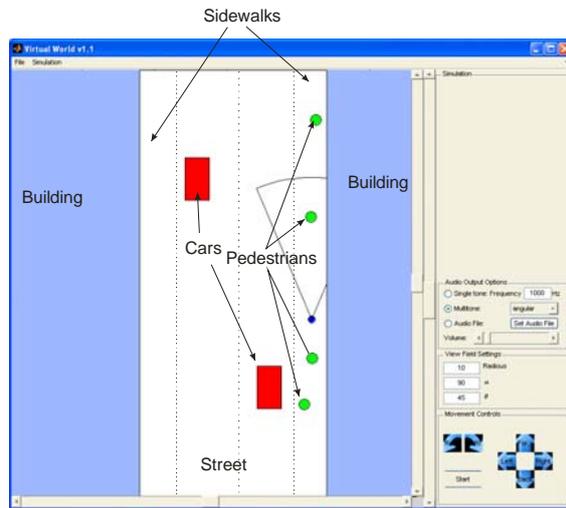


Fig. 6. Simulator's window with abstract map loaded

REFERENCES

- [1] Marchan P., Holland T., *Graphics and GUIs with MATLAB*, Chapman & Hall, 2003.
- [2] HyperPhysics, „Sensitivity of Human Ear“, *Sensitivity of Human Ear*, 2005. [Online] Available on: <http://hyperphysics.phy-astr.gsu.edu/Hbase/sound/earsens.html>. [Accessed: 11. Jul 2009.]
- [3] Wikipedia, „Equal loudness contour“, *Equal loudness contour*, 2009. [Online] Available on: http://en.wikipedia.org/wiki/Equal-loudness_contour. [Accessed: 11. Jul 2009.]
- [4] Christopher J. Plack, *The sense of hearing*, Lawrence Erlbaum Associates, 2005.
- [5] Wikipedia, „Sound localization“, *Sound localization*, 2009. [Online] Available on: http://en.wikipedia.org/wiki/Sound_localization. [Accessed: 11. Jul 2009.]
- [6] Wikipedia, „Head-related transfer function“, *Head-related transfer function*, 2009. [Online] Available on: http://en.wikipedia.org/wiki/Head-related_transfer_function. [Accessed: 11. Jul 2009.]

Computer Based Power Factor and Distortion Measuring for Small Loads

Marko Dimitrijević and Vančo Litovski

Abstract - Power factor and distortion measuring usually requires dedicated and expensive equipment. Computer-based acquisition modules and software provide for a possibility to create simple and non-expensive methods and instruments for power factor measurement and distortion characterization of small loads and bring all advantages of virtual instrumentation. A new approach to power quality characterization by measuring power factor, distortion and several other parameters of small electric loads (up to 0.5kW) will be described in this paper. Besides the low price maximum versatility and adaptability is provided without any loss in accuracy.

Keywords – Power factor, distortion, virtual instrument

I. INTRODUCTION

Power quality is a relatively ambiguous concept, limited mostly to conversations among utility engineers and physicists, but as electronic appliances take over the home, it may become a residential issue as well [1].

According to a recent study [2], nowadays we are witnessing changes in the demand and energy use. In fact the new demand determines “new” load characteristics and trends while changes in the nature of the aggregate utility load happen. All of that is mostly due to the electronic plug-ins that became ubiquitous. Table 1 shows the typical household loads and their participation (in average) within the home power consumption.

TABLE I
THE STRUCTURE OF THE ENERGY NEEDS FOR ELECTRONIC EQUIPMENT IN AVERAGE HOUSEHOLD

| Type of consumption | Percentage |
|------------------------|------------|
| Entertainment | 60 |
| Information technology | 31 |
| Small appliances | 5 |
| Miscellaneous | 4 |
| Type of consumption | Percentage |

Note that it is presumed that the overall household consumption for electronic appliances will rise with a rate of 6% per year so reaching 29% of the total household consumption in the year 2030. In the same time the household consumption is expected to reach 40% of the

Marko Dimitrijević and Vančo Litovski are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: (marko.dimitrijevic,vanco.litovski)@elfak.ni.ac.rs.

overall electricity demand.

Electronic loads are strongly related to the power quality thanks to the implementation of AC/DC convertors that in general draw current from the grid in bursts. In that way, while keeping the voltage waveform almost unattached, they impregnate pulses into the current so chopping it into seemingly arbitrary waveform and, consequently, producing harmonic distortions. The current voltage relationship of these loads, looking from the grid side, is nonlinear, hence nonlinear loads. The existence of harmonics gives rise to interference with other devices being powered from the same source and, having in mind the enormous rise of the number of such loads, the problem becomes serious with serious, sometimes damaging, consequences and has to be dealt with properly.

In linear circuits, consisting of linear loads, the currents and voltages are sinusoidal and the power factor effect arises only from the difference in phase between the current and voltage. In single phase system it is referred to as the displacement power factor or $\cos(\varphi)$. When nonlinear loads are present one should introduce new quantities in the calculations emanated by the harmonics and related power components [3]. Now the power factor can be generalized to a total or true power factor where the apparent power, involved in its calculations, includes all harmonic components. This is of importance in characterization and design of practical power systems which contain non-linear loads such as rectifiers, and especially, switched-mode power supplies [4].

Industry standards regulate the limits (minimum) of power factor. One of the most paradigmatic examples is personal computer that typically includes switched-mode power supply (SMPS) with output power ranging from 150W to 500W. SMPS with passive power correction factor (PFC) can achieve power factor of about 0.7–0.75, SMPS with active PFC—up to 0.99, while SMPS without any PFC has power factor of about 0.5–0.65 in the best. The current EU standard EN61000-3-2 imposes that all SMPS-es with output power more than 75W must include at least passive power factor correction [5].

Since the problem of distortion becomes ubiquitous one has to democratize the measurement of the properties of this kind of loads. There should be an opportunity to characterize every single product wherever convenient to a customer. Contemporary measurement of power factor and distortion, however, usually requires dedicated equipment.

For example, use of a classical ampermeter will return incorrect results when attempting to measure the AC

current drawn by a non-linear load and then calculate the power factor. A true RMS multimeter must be used to measure the actual RMS currents and voltages and apparent power. To measure the real power or reactive power, a wattmeter designed to properly work with non-sinusoidal currents must be also used. Accordingly, a set of measurement devices, some of them palm-held, are available on the market performing many measurements while offering connection to a computer. To our knowledge the price of such devices is considerably above the price of an average lap-top computer.

We are promoting here a new concept of measurement of the power quality parameters of small loads. The main idea is to perform the measurement in two phases that impose partitioning both the hardware and software subsystems in two. The parts will be referred to here as the Power Factor and Distortion Measurement Device (PFDDMD) and the computer. The former is a "sampling" device while the other is the processing device. In that way only minor activities (sampling and A/D conversion of the current and voltage) are performed within the sampling subsystem the computer taking the main role in data processing of all kind. In this way we introduce a robust and versatile system with practically unlimited possibilities. The price of the system is defined by the sampling device to be incomparably lower than the palm-held devices now available on the market, and the price of the software, that, in our opinion, may be easily written so lowering the total price of the overall measurement system. We will demonstrate in the next that practically every conceivable quantity related to power measurement may be obtained.

The practical implementation of the new system was based on two broadly available components: the National Instruments NI USB-9215A acquisition module and the LabVIEW software platform.

In the next we will first introduce the basic definitions that are expressing how the measured quantities are calculated from the current and voltage waveforms. Then, the hardware and software components of the system will be described briefly. Finally, measurement results will be given to demonstrate the method and to give some information to the public about the properties (quality) of some small loads now available on the market.

II. THE DEFINITIONS OF THE FUNDAMENTAL QUANTITIES

Traditional power system quantities such as effective value, power (active, reactive, apparent), and power factor are defined for pure sinusoidal condition. In the presence of nonlinear loads, however, the system no longer operates in sinusoidal condition and use of fundamental frequency analysis does not apply any more. They will be numerically calculated from sampled voltage and current sequences [6]. In the next we will first introduce definitions that were used

as basis to the development of the new measurement tool and for characterization of the measured quantities.

Power factor is simply defined as the ratio of real power to apparent power, or:

$$TPF = \frac{P}{S} \quad (1)$$

The real power, P , is the average, over a cycle, of the instantaneous product of current and voltage:

$$P = \frac{1}{T} \int_{t_0}^{t_0+T} v(t) \cdot i(t) \cdot dt \quad (2)$$

where t_0 is arbitrary time (constant) after equilibrium, and T is the period (20 ms in European and 1/60 s in American system, respectively). The apparent power is the product of the root mean square value of current times the root mean square value of voltage:

$$S = V_{RMS} I_{RMS} \quad (3)$$

The RMS value is calculated according to the well known formula:

$$X_{rms} = \sqrt{\frac{1}{T} \int_{t_0}^{t_0+T} (x(t))^2 dt} \quad (4)$$

According to that definition for the RMS values of the voltage and current one may produce the following

$$V_{rms} = \sqrt{\sum_{i=1}^m V_{rms i}^2} \quad (5)$$

and

$$I_{rms} = \sqrt{\sum_{i=1}^m I_{rms i}^2} \quad (6)$$

where $V_{RMS i}$ and $I_{RMS i}$ are the RMS values of the i^{th} harmonic of the voltage and current, respectively, while m represents the number of harmonics taken into account.

For a pure sinusoidal case, i.e. when $v(t) = V_m \cdot \cos(\omega \cdot t + \alpha)$ and $i(t) = I_m \cdot \cos(\omega \cdot t + \beta)$, where V_m and I_m are the amplitudes, and α and β the phase angles of the voltage and current, respectively, the power factor is readily obtained to be $PF = \cos(\alpha - \beta) = \cos(\varphi)$. If harmonics are present the *power factor relates only the main (first) harmonics* of the current and voltage and will be referred to as the displacement power factor. New quantity is introduced to express the influence of the harmonics to the power losses. It was referred to as the distortion power factor. In the next we will first introduce its definition.

The total harmonic distortions, THD , are calculated from the following formula:

$$THD = \sqrt{\frac{1}{y_1} \sum_{i=2}^m y_i} \quad (7)$$

where y_i , $i=1, 2, \dots, m$ stands for the i^{th} harmonic of the current or voltage. One usually computes the current THD from:

$$THD_I = \frac{1}{I_{\text{rms}1}} \sqrt{\sum_{i=2}^m I_{\text{rms}i}^2} \quad (8)$$

and the voltage THD from

$$THD_V = \frac{1}{V_{\text{rms}1}} \sqrt{\sum_{i=2}^m V_{\text{rms}i}^2} \quad (9)$$

The above definition of the THD may lead to values that are higher than 100%. That is the reason why in some proceedings an alternative definition is used:

$$THD_I^* = \sqrt{\frac{I_{\text{rms}}^2 - I_{\text{rms}1}^2}{I_{\text{rms}}^2}} = \sqrt{1 - \frac{I_{\text{rms}1}^2}{I_{\text{rms}}^2}} \quad (10)$$

We define the distortion power factor as the quantity

$$DPF = I_{\text{rms}1} / I_{\text{rms}} \quad (11)$$

and introducing (6) and (8) we obtain

$$DPF = \frac{1}{\sqrt{1 + THD_I^2}} \quad (12)$$

The total power factor defined by (1) is obtained as a product of the displacement power factor and the distortion power factor as

$$TPF = DPF \cdot \cos(\varphi) \quad (13)$$

where φ represents phase difference between first harmonics of voltage and current. $\cos(\varphi)$ is displacement power factor; in case of linear load, total power factor is equal to displacement power factor.

Two more quantities are of interest for more complete characterization of the waveforms encountered. The direct component of the current defined as

$$I_{\text{DC}} = \frac{1}{T} \int_{t_0}^{t_0+T} i(t) \cdot dt \quad (14)$$

and the CREST factor. The later is defined as the ratio of

the peak value and the RMS value in a given time slot:

$$CF = V_{\text{peak}} / V_{\text{RMS}} \quad (15)$$

III. SYSTEM IMPLEMENTATION

A. Hardware implementation

The measurement of the quantities mentioned in the previous paragraph is performed by a system whose components are depicted in Fig. 1. The Power Factor and Distortion Measurement Device (PFDMD) is connected to the grid from one side, and transfers the power to the device under test (load) while sampling the values of the current and voltage waveforms of the load. The sampled values are appropriately conditioned and coded, and then directly delivered to the computer via USB terminal. All computations are performed by the software implemented in the computer. For example, the frequency of the measured waveform is easily extracted by post processing of the signals. In the same time the computer is used as an oscilloscope enabling display of the measured and derived waveforms, as an interactive monitor allowing different quantities to be displayed, as a data storage device creating measurement logs and databases, and a communication means enabling remote control of the measurement and on-line delivery of the results.

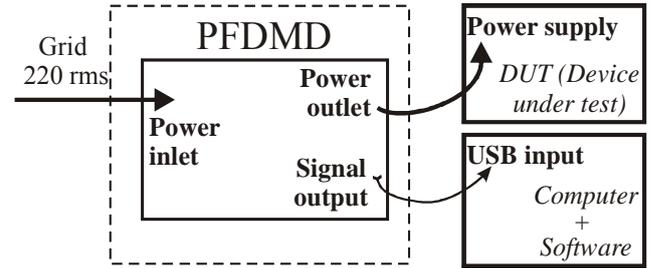


Fig. 1. The PFDMD and its surroundings

Data acquisition within the PFDMD is performed using an acquisition module. It is based on National Instruments NI USB-9215A acquisition module (DAQ) [7]. The data acquisition module has four channels of simultaneously sampled voltage inputs with 16-bit accuracy, 100kS per channel sampling rate and 250V_{RMS} channel-to-earth isolation, suitable for voltage measurements up to 40th harmonic (2kHz). It also provides for portability and hot-plug connectivity via USB interface.

Two channels are used: for voltage and current measurement. The analogue inputs of acquisition module are connected using differential measurement method [8] due to better noise rejection and channel-to-earth isolation.

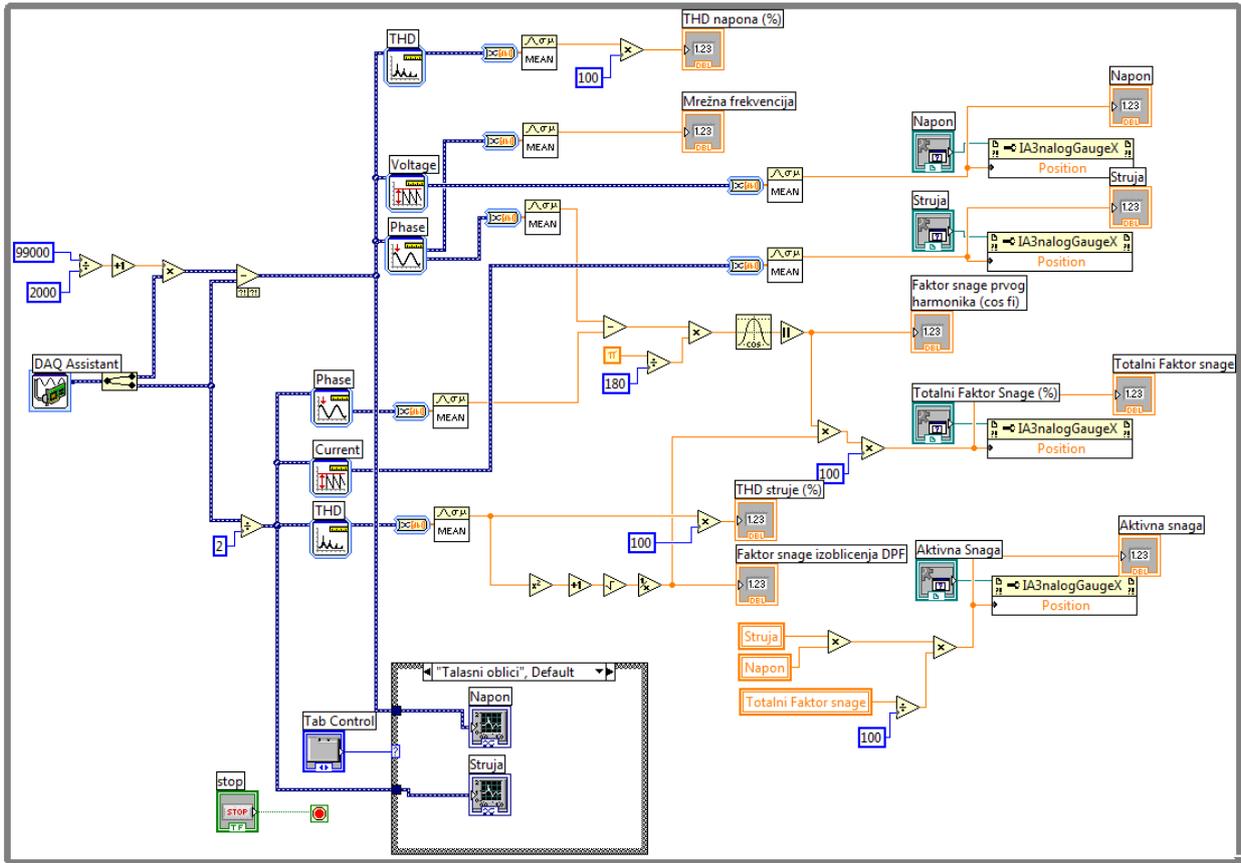


Fig 2. Main thread of application

B. Software implementation

The Software part of the power factor and distortion measuring system is implemented in *National Instruments LabVIEW* developing package (Fig. 2.), which provides simple creation of virtual instruments [9]. Virtual instruments consist of interface to acquisition module and application with graphic user interface.

Interface to acquisition module is implemented as device driver. USB-9215A module is supported by NIDAQmx drivers. All the measurements are performed using virtual channels. A virtual channel is collection of property settings that can include name, a physical channel, input terminal connections, the type of measurement or generation, and scaling information. A physical channel is a terminal or pin at which an analogue signal can be measured or generated. Virtual channels can be configured globally at the operating system level, or using application interface in the program. Every physical channel on a device has a unique name.

For better performance, the main application has been separated into two threads. The first thread has functions

for file manipulation and saving measured values (Fig. 4.). All measured values will be saved in HTML file format.

The user interface of the virtual instrument consists of visual indicators. It provides basic functions for measurement. The indicators — gauges and graphs — show measured values. All measured values are placed in a table, and after the measurement process in appropriate file. User interface also provides controls for data manipulation and saving measured values.

IV. CONCLUSION AND MEASUREMENTS

The main research goal of this paper was to present a simple and non expensive method and instrument for power factor measurement and distortion characterization of small electric loads. The measured values of current and voltage THD, DPF, TPF and other parameters are shown in virtual instrument screenshots (Fig. 3). Virtual instrument shows current and voltage spectra as well as their waveforms in real time.

Virtual instrument is capable for real-time sampling and measuring voltage and current of DUT, providing possibility for transient analysis in time and frequency

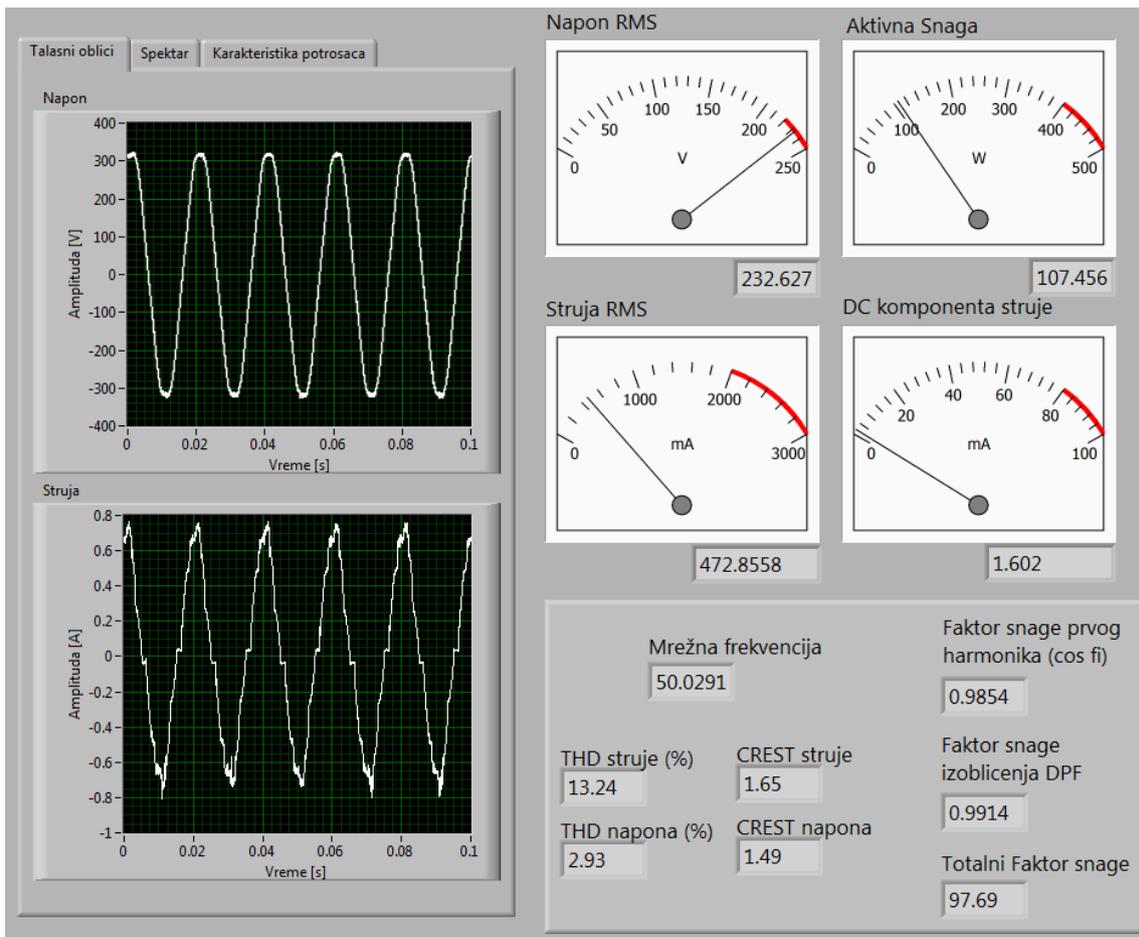


Fig. 3. Virtual instrument — Computer monitor with good distortion correction, with real power of cca 100W. $\cos(\varphi)$ is 98.54%, current THD is 13.24%, voltage THD is 2.93%, DPF is 99.14%, TPF is 97.69% — current and voltage waveforms are shown at the left side of virtual instrument panel

domain. Measuring time depends of storage capacity of host PC.

REFERENCES

- [1] A. T. De Almeida, Understanding Power Quality, *Home Energy Magazine Online* **10** (1993), <http://www.homeenergy.org/archive/hem.dis.anl.gov/ee/hem/93/931113.html>
- [2] L. Freeman, The Changing Nature of Loads and the Impact on Electric Utilities (2009), www.techadvantage.org/2009ConferenceHandouts/2E_Freeman.pdf
- [3] H.W. Beaty, D.G. Fink, *Standard handbook for electrical engineers* (McGraw-Hill, New York, 2007)
- [4] G. Moschopoulos, Single-Phase Single-Stage Power-Factor-Corrected Converter Topologies, *IEEE Trans. on Industrial Electronics* **52** (2005) 23-35.
- [5] Fernández, J. Sebastian, P. Villegas, M.M. Hernando, J. Garcia, Dynamic Limits of a Power-Factor Preregulator, *IEEE Trans. on Industrial Electronics* **52** (2005) 77-87.
- [6] Power Factor Correction (PFC) Handbook, ON Semiconductor, 2004, <http://www.onsemi.com/tech-support>.
- [7] PCI USB-9215 Product Data Sheet, National Instruments, <http://ni.com>
- [8] LabVIEW™ 7 Express Measurement Manual, National Instruments, <http://ni.com>
- [9] LabVIEW™ 7 Express User Manual, National Instruments, <http://ni.com>.

Strategies against Side-Channel-Attack

Milena Stanojlović, Predrag Petković

Abstract – This contribution discusses cryptographic algorithm in hardware that protects the information leaks out of the device through so called „side channels“. This class of attacks is called side-channel attacks (SCA). Important information, such as secret keys, can be obtained by observing the power consumption, the electromagnetic radiation, the timing information etc. There are several types of protection and some will be discussed in this paper. Special attention is paid to Wave Dynamic Differential Logic (WDDL) that was evaluated in terms of load symmetry on an example.

Keywords – Side Channel Attack, Wave Dynamic Differential Logic.

I. INTRODUCTION

Data security becomes very important issue in everyday life. Starting from credit cards, coded alarm systems to all types of cipher-protected data transfer it is necessary to hide code keys from unauthorized misuse. The first defending line is using complex multi-bit ciphers. Crushing them by simple software tools based on proper combination search become very time-consuming. Longer password and more sophisticated coding algorithms result to the bigger number of combinations and therefore the better protection. One can say that the problem of data protection could be solved just increasing the number of combinations. However, the value of hidden data enormously increases. This inspires potential attackers to invest more money and brain in order to crack cipher. It has been shown in [1] that monitoring power helps a lot in finding cipher. Thereafter other methods emerged that makes cipher cracking easier, like Simple Power Analysis (SPA), Differential Power Analysis (DPA) and Electromagnetic Analysis (EMA) [2]. Common to all these methods is analysis of information that leaks from physically implemented hardware. They can be collected only if somebody intentionally uses sophisticated probes to attack crypto-processor. Therefore they are named side channel attack. There are different attack tactics like Fault induction attack, Timing attack, Probing attack [2].

The scientific community responses with new hardware and software based countermeasures.

The aim of this paper is to enlighten some strategies in fighting against SCA. Especially authors are interested in protecting data from power-meters during automatic meter reading [3]. It is expected that new solid-state power-meter designed as ASIC in Laboratory for Electronic Design

Milena Stanojlović and Predrag Petković are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: milenastanojlovic@yahoo.com; predrag@elfak.ni.ac.rs.

Automation at University of Niš, comprise a communication block resistible to SCA. Therefore it is desirable to fight against SCA within standard CMOS technology and preferably using standard cell library. Therefore Waveform Dynamic Differential Logic (WDDL) is in scope of our interest and it will be discussed from implementation point of view. Our goal is to determine the permitted amount of load mismatch that still guarantees resistivity to DPA attack.

The paper is organized as follows. The subsequent section gives a brief survey of countermeasures. The third section presents basics of WDDL. Influence of unsymmetrical load of a WDDL cell to the SCA resistivity is described on example of AND gate in the fourth section together with simulation results.

II STRATEGIES AGAINST SCA

Although power analysis and EMA requires using different probes the source of data leakage is common in both cases. The leakage is the outcome of changes in IDD during logic state transitions. Each change 0-1 requires additional charge to be passed from bias to the output capacitance. In contrary change 1-0 discharges load and no current flows from VDD. This is sufficient to detect what is happening inside IC just by monitoring IDD.

All strategies in fighting against leaking data through power changes rely on hiding correlation between the logic state changes and the waveform of power. Depending on the level where performed they can be sorted as measures at architectural, algorithmic or gate level.

In scope of methodology they can be categorized as randomizing, masking and signal independent power change.

Randomization at algorithmic level relies on frequently change of secret key to avoid possibility of finding the correlation.

Masking methods require additional logic operations to cover real data. It is possible to perform them on algorithmic level and on the gate level, as well. However, higher order power analysis can crack masking.

There are several ways to make power consumption of a cell independent on data flow.

One is to keep constant power consumption all the time. This is possible by inserting analog modules. However the overall consumption of power is considerably high.

The other way is to force all digital cells to have the same power pattern for every logic change. This class of methods is known as *Dual-rail with Precharge Logic* (DPL). All signals are duplicated and have true and false representations. The cells operate in alternated *precharge*

and *evaluation* phases to ensure exactly one switching event per cycle. *Wave Dynamic Differential Logic* (WDDL) [4] is good representative of DPL. It can be implemented with standard CMOS cells and therefore it is good candidate for implementation in standard ASIC technologies.

III WAVE DYNAMIC DIFFERENTIAL LOGIC

The main purpose of a WDDL cell is to provide uncorrelated power consumption to the operated data. Therefore it should have the same number of transitions for every combination of input signals. In case of inverter it means that every change on input must have the same contribution to IDD. This is possible if inverter is realized with two standard invertors (connected to the same VDD) as Fig. 1.a shows.

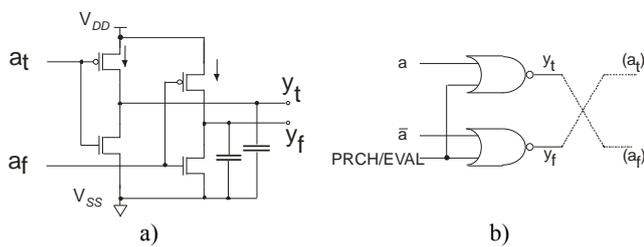


Fig. 1. WDDL inverter

Indexes *t* and *f* denotes true and fault signals, respectively. Knowing that $a_f = \text{NOT}(a_t)$ it is obvious that for same load any change on $a = a_t$ will produce the same IDD.

However, for other types of cells it is not sufficient to have duplicated hardware. Each cell should have own dual cell. This means that for every $y_t = a_t \square b_t$ the complementary output is needed such as $y_f = \text{NOT}(y_t) = \text{NOT}(a_t) * \text{NOT}(b_t)$. Note that \square and $*$ denote different (complementary) operators. For AND operator OR is complementary and vice versa. Fig. 2 represents WDDL AND cell.

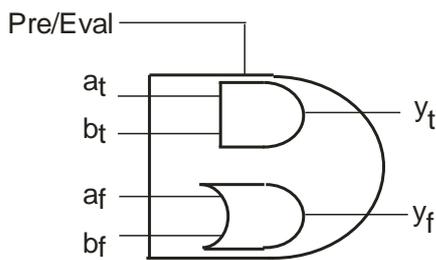


Fig. 2. WDDL AND cell

In order to provide the same IDD for every input change, combinational cells should work in two phases. During *precharge* phase all signals are forced to the low logic level. Thereafter, in *evaluating* phase outputs establish the proper values. Hence, the inverter cell is not realised as in Fig. 1.a but rather as shown in Fig 1.b. The

same architecture is used as generator for waveforms of a_t and a_f from a and $\text{NOT}(a)$ signals (dashed lines).

Figure 3 shows waveforms of controlling Precharge/Evaluation signal and all input and output signals for the case that corresponds to the single-rail AND cell stimulated with patterns $a=1, b=0$ and $a=1, b=1$.

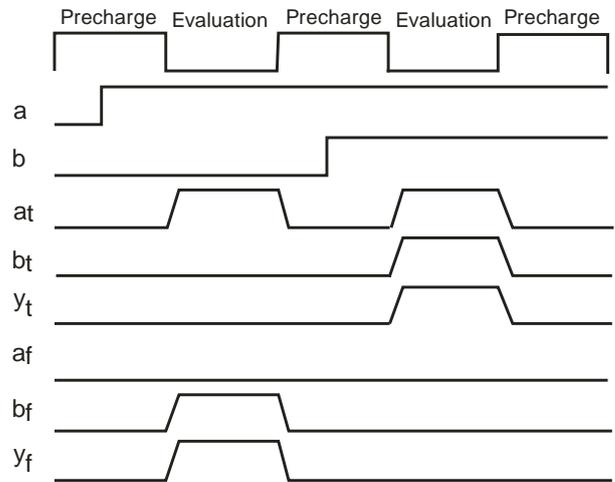


Fig. 3. Waveforms for WDDL AND cell

During precharge phase all signals are set to low level. During evaluating phase only exactly one of outputs goes to the high level. Therefore only one load capacitance will charge from VDD.

Obviously, if input signals come in slightly different moment WDDL architecture implemented for NAND cell will generate glitches observable to attacker. Simultaneously this will produce leakage and all design becomes vulnerable. This is reason why WDDL works only with “positive” gates (AND, OR) and not with negative gates (NAND, NOR). There is modification of WDDL that is capable to work with negative gates named Dual Spacer Dual Rail Logics [5].

So far it is clear that good SCA protection costs duplication in hardware. Unfortunately with sequential gates the price is even higher. To retain good DPA protection it is necessary to quadruple number of flip-flops [6]. In practical realizations in FPGA it is reported that hardware overhead is over five times and that operating frequency is lower for more than twice [6].

This price is acceptable having in mind the security aspect. However, WDDL is reliable only if loads of both “true” and “false” signals are balanced. When that is not case there is leakage due timing difference [7] that jeopardizes the overall concept.

Therefore several algorithms were developed to provide symmetrical routing. The main advantage of WDDL is that it can be implemented with standard cell libraries. Hence, standard routing tools can be utilized. Unfortunately they are not optimized for symmetry and tricky part is how to obtain symmetrical wires with minor intervention in standard routing algorithms

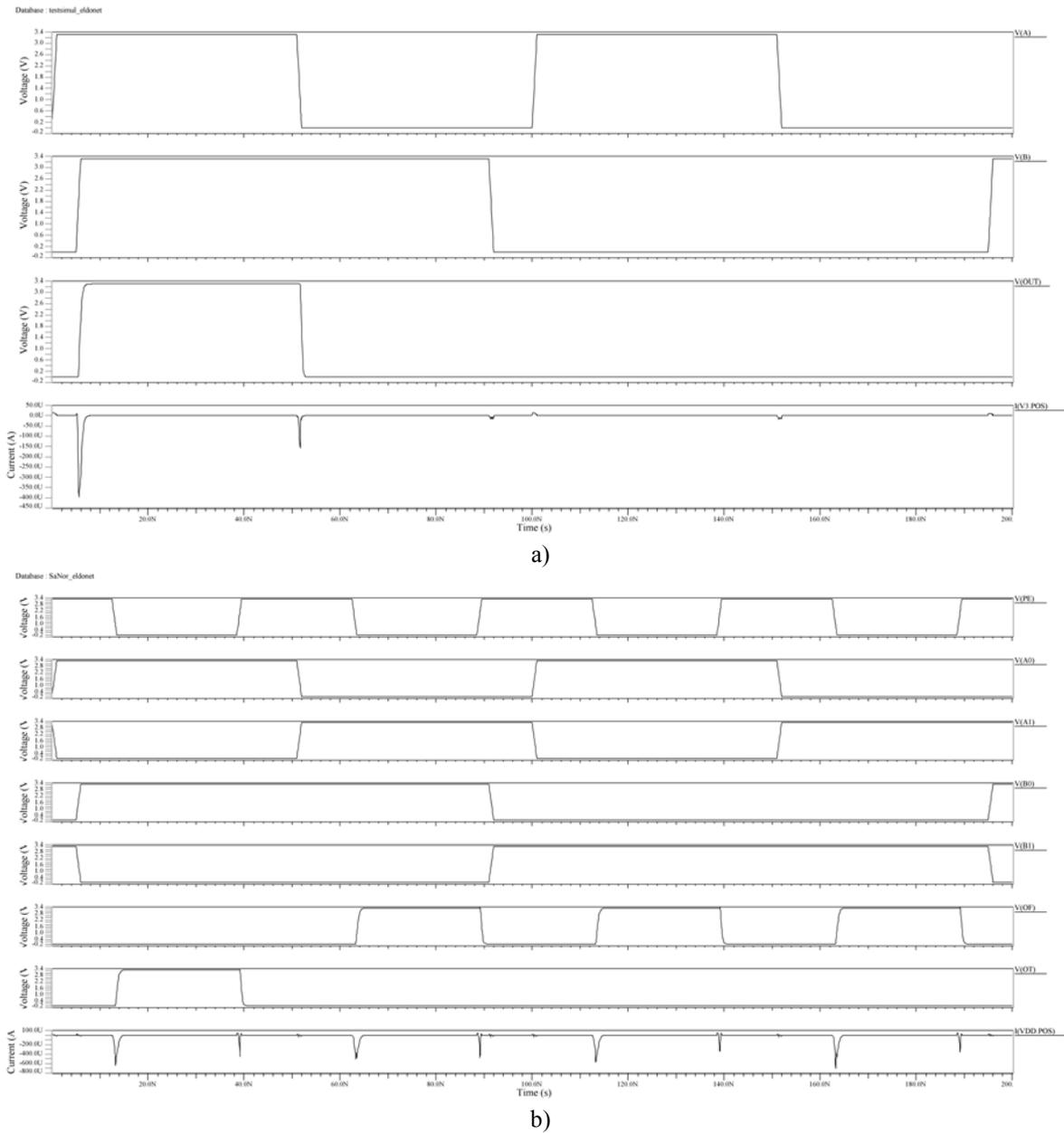


Fig. 4 Waveform of IDD for a) single-real AND gate; b) WDDL AND gate with balanced load

In the following section we will present the influence of mismatched load on power leakage.

IV WDDL RESISTIVITY TO UNBALANCED LOAD

As an example WDDL AND gate will be considered. It is designed in CMOS035 technology. The IDD waveform of a single-rail AND gate designed in the same technology will serve as a reference. Thereafter, an ideally balanced WDDL AND cell is simulated.

Figure 4 depicts waveforms of both gates. IDD waveform of single rail AND gate exploits very clear

difference when output changes state from 0 to 1 and from 1 to 0, as Figure 4.a shows. Therefore, the whole information about state at the output is visible through IDD. In contrary, supply current of WDDL implemented AND gate have regular pattern independently on output logic states as the bottom diagram in Fig. 4.b presents. Consequently it is immune on side channel attack.

It is interested to evaluate what leakage should be expected under different amount of mismatched load. A set of several simulations were done for different rate of mismatch. As measure of mismatch the integral of IDD is used. Actually the integral corresponding to transition 0-1 is compared with that obtained for change 1-0. Obviously

WDDL cells have change on “false” output during neutral transitions (0-0 and 1-1), as well. In order to hide leaking information about output logic state the integrals of current corresponding to all changes should be the same. There are several methods for their comparison and, accordingly, for design apprising. One of them is to compare integrals of IDD during evaluation phase with each other. Particularly WDDL AND gate was analysed for load capacitances unjust of up to ±15%.

Table I summarizes results for different mismatch of load values. Assuming that mismatch of 10% is sufficient to explore observable leakage, one can conclude that it can be reached for load mismatch up to 20%.

TABLE I
WDDL GATE MISMATCHED

| Tran. | Single AND | WDDL Ct/Cf=1 | WDDL ΔC=5% | WDDL ΔC=15% |
|-------|---------------------------------|-----------------|---------------------|---------------------|
| 0-0 | -9.82837E-15 (A=(0->1), B=1) | -4.97E-13 | -5.10E-13 -2.61% | -5.36E-13 -7.86% |
| 0-1 | -5.45165E-14 (A=(1->0), B=1) | -4.99E-13 | -5.12E-13 -2.62% | -5.38E-13 -7.88% |
| 1-0 | 1.01E-14 (A=1, B=(1->0)) | -4.81E-13 | -4.94E-13 -2.71% | -5.20E-13 -8.16% |
| 1-1 | -3.00538E-13 (A=1, B=(0->1)) | -4.86E-13 | -4.99E-13 -2.68% | -5.25E-13 -8.05% |

V CONCLUSION

This paper presented some of countermeasures against SCA. WDDL was particularly examined in scope of unsymmetrical load. The results obtained for ideally matched outputs were compared to several mismatch levels for typical exploitation conditions. The obtained results will be analyzed in scope of technology and geometrical parameters. Actually for known tolerances of particular technology one can estimate appropriate wire width and/or metal level that should be used for best matching false and true signals.

Capacitance and resistance of a wire depend on technological and geometrical parameters.

Therefore, for known amount of the parameter mismatch it is possible to calculate physical dimensions of wires that could keep matching within acceptable limits. Besides layout designer could decide what shape and width of wires to use. It is known that it is easier to match larger patterns. Hence, wire dimensions could be customized for better matching. Tolerances of wire capacitance and

resistance depend on metal layer. It is feasible to establish some kind of design rule that will limit wire length in respect of matching similar to the *antenna rule*.

When analyzing load mismatch it is important to be aware of different timing effects that should open up under different faulty circumstances. In order to get good insight into WDDL vulnerability one needs to perform thorough corner analysis for lower VDD, higher temperature, quicker/slower excitation.

The obtained results will help in making decision on what type of SCA protection should be most appropriate for implementation in integrated power meter.

ACKNOWLEDGMENT

This work was supported by The Serbian Ministry of science and technology development within the project TR 11007.

REFERENCES

- [1] Kocher, P., Jaffe, J. and Jun, B., “Differential Power Analysis,” in Proceedings of CRYPTO’99, ser. LNCS, vol. 1666. Springer-Verlag, 1999, pp. 388–397.
- [2] Quisquater, J.-J., “Side channel attacks State-of-the-Art”, Report, Oct. 2002. Available on: http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/1047_Side_Channel_report.pdf [Accessed 15.12.2009].
- [3] Litovski, V., Petković, P., “Why The Power Grid Needs Cryptography?”, *Electronics*, Vol. 13, No. 1, YU ISSN 1450-5843, June, 2009, pp. 30-36
- [4] Tiri, K., and Verbauwhede, I., “A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation,” Proc. of DATE’04. IEEE Computer Society, February 2004, pp. 246–251, Paris, France.
- [5] Sokolov, D., Murphy, J., Bystrov, A., and Yakovlev, A., “Design and Analysis of Dual-Rail Circuits for Security Applications”. *IEEE Transactions on Computers*, Vol. 54, No. 4, 2005, pp. 449–460, ISSN 0018-9340.
- [6] Selmane, N., Bhasin, S., Guilley, S., Graba, T., and Danger, J.-L., “WDDL is Protected Against Setup Time Violation Attacks”, HAL – CCSD, hal-00410135, version 1 – 17, Aug. 2009
- [7] Guilley, S., et al., “Shall we trust WDDL? in Future of Trust in Computing”, In “Future of Trust in Computing”, Springer, 2009, pp. 208-215, ISBN: 978-3-8348-0794-6 (Print) 978-3-8348-9324-6 (Online).

Multi Channel $\Sigma\Delta$ A/D Converter for Integrated Power Meter

Dejan D. Mirković, Predrag M. Petković

Abstract – This paper describes three architectures for multi-channel sigma-delta ADC IC design. The proposed solution is aimed for the front-end of a three-phase integrated power meter. The previous version of the power meter is to be redesigned by substituting six ADCs with two: one for converting currents and another for converting voltages in the three-phase power system. Therefore one pair of analog 3-to-1 multiplexers precedes ADCs. Discussion of advantages and drawbacks of the proposed solutions is illustrated by simulations using ADMS simulator that is a part of Mentor Graphics design kit.

Keywords – A/D Converter, $\Sigma\Delta$ Modulator, Multiplexer, Power-meter

I INTRODUCTION

The analog frontend of an integrated power meter is relatively small comparing to the digital part, especially when number of transistors is used as measure. However, in all other aspects it represents the crucial part of the overall system on chip (SoC). The functionality of the system depends on the analog part. Moreover, even small inconsistencies in this block considerably ruins measurement characteristics of the SoC. Consequently designing of the analog part requires a lot of time, strict sticking to the design rules and a lot of designer's care and concentration.

In this paper we consider redesigning the front-end of three-phase solid-state power meter. Current version of the chip named IMPEG-2 has been designed in LEDA laboratory within Faculty of Electronic Engineering in Niš, Serbia. It is the third member of IMPEG solid-state power meter series. The first one is IMPEG-1 dedicated for power metering in single phase power systems [1, 2, 3]. It consists of analog front-end, digital filters and DSP block as Figure 1 presents.

The power meter is based on measurement of instantaneous values of voltage and current. They are sampled in two separated channels that consist of two SD oversampled ADCs that launch digital filters and store data in corresponding registers. DSP block calculates active, reactive and apparent powers and energies. Accuracy of the calculated powers and energies mainly depends on precision of measured current and voltage. IMPEG 1 was designed in 2003/04 and prototyped and tested in 2005.

Dejan Mirković and Predrag Petković are with the Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: dejan.mirkovic@elfak.ni.ac.rs; predrag@elfak.ni.ac.rs.

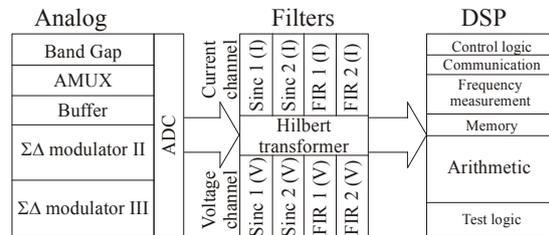


Fig. 1. Structure of IMPEG chip

Encouraged with the obtained results the LEDA team designed three-phase version of IMPEG in 2006/07. It differs from IMPEG-1 mainly in digital part. The analog front-end was triplicate of the analog part of the basic IMPEG version.

LEDA team is motivated to optimize area, power consumption and performances in the new variation of IMPEG. Therefore the digital part has been enriched with temperature self-calibration and low-power-driven design. The aim is to decrease area of the analog part replacing six ADCs with as few as possible. This paper analyses possible architectures for multichannel sigma delta A/D convertors.

The paper is organized as follows. The next section gives a brief overview of the current version of IMPEG ASIC. The third section describes possible solutions for compact analog frontend and the subsequent section presents the chosen architecture based on two ADCs.

II IMPEG-2

IMPEG-2 represents three-phase version of IMPEG-1. However it is not plain triplication of the mono-phase version.

Figure 2 presents the structure of IMPEG 2 [4].

The main differences are in digital part. Computation engine that has relayed on DSP block is accomplished with embedded MCU 8052. Besides, drivers for LCD display, real time clock, digital PLL, original acquisition block and converter for battery driven bias are incorporated as well. Moreover, major innovations have been made in digital filter block [5]. Namely, Sinc, FIR and Hilbert transformer filters implement compact MAC architecture and time multiplexing technique. As result the overall area for digital filters is increased only for 3% in comparison to the mono-phase realization. Practically, all three phases in voltage and current channel separately share the same hardware for digital filtering, as Figure 3 shows.

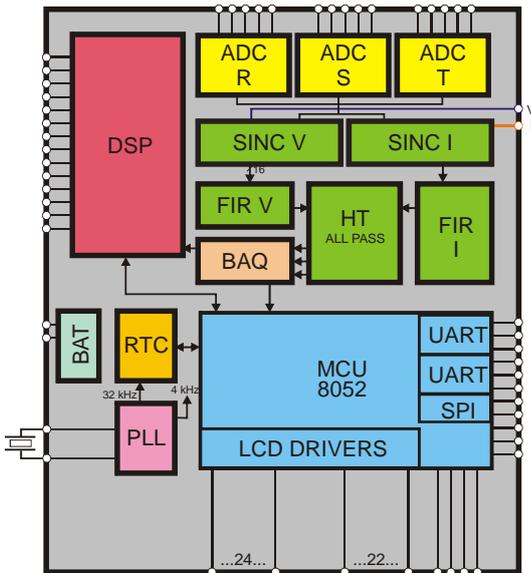


Fig. 2. Structure of IMPEG 2

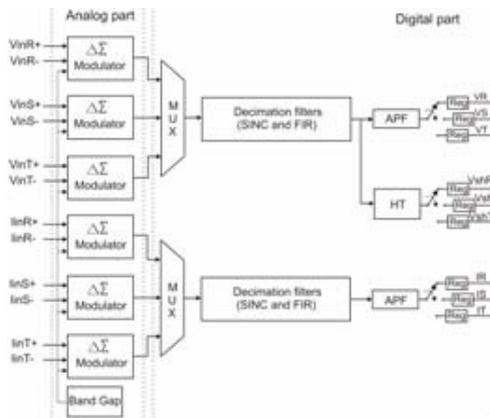


Fig. 3. ADC part of IMPEG 2

Obviously, six sigma-delta modulators spoil the compactness of the solution. Therefore the next section considers possibilities for designing more compact solution. All of them are based on multiplexed approach.

III MULTIPLEXED SD MODULATOR

The basic idea is to spare chip area for as many modulators as possible without lost of functionality. Simultaneously it is desirable to retain already designed and tested full custom designed macro-cells. In the scope of analog frontend it means not to change the order of modulator and its basic blocks: integrators, opamps, band gap reference, quantizer and single-bit DAC.

The structure of used $\Sigma\Delta$ modulator is shown in Fig. 4

[3].

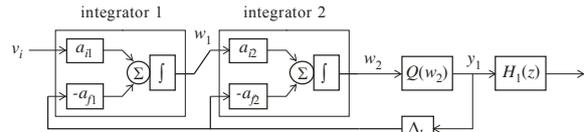


Fig. 4. Structure of second order $\Sigma\Delta$ modulator

Integrators are realized using switch capacitor (SC) method [8]. Although differential architecture is practically used, for the sake of simplification it will be illustrated in single-ended version as Fig. 5. presents.

Observing Fig. 3 and Fig. 5 one can conclude that the most compact solution would be to replace all six modulators with one as Fig. 6 illustrates. The proposed architecture is inspired by [7, 8].

All six inputs, three for voltages of three phases denoted with VR, VS and VT and three for the corresponding currents denoted as IR, IS and IT comes to the input of multiplexer 6 to 1 (MUX6-1 in Fig. 6). However, to retain the same sampling frequency as the original circuit has (Fig. 3) and to persist in-phase sampling of all six signals, the multiplexer is driven from sample and hold circuits. All analog signals should be sampled with the same clock at frequency of 524288 Hz. The hold time should last long enough to allow conversions for all six channels. To accomplish this task one needs at least six times faster switching in modulator than the sampling frequency. Namely it requires clock of 3.14 MHz.

Digital signals are sent to the appropriate registers at the output where clock rate is returned to the sampling frequency rate of 524288 Hz. Therefore the rest of SoC will not be disturbed.

Knowing limitations of the used folded cascode operational amplifier of GBW=7.3 MHz and slew-rate of 5V/ μ s [6], this means that the proposed architecture is close to the acceptable upper margins of the design.

Opposite solution is to multiplex only two channels: one for voltage and another for current within a single modulator. Three-phase power meter needs three such architectures, one for every phase. Supposing that sampling data rate remains 524288 Hz this implies that modulator has to be switched with rate of 1.05 MHz. This is below limits of implemented opamps and therefore feasible solution. However the overall spare in area is only 1/2 of the solution presented in Fig. 3. Besides, successive sampled values on SCs will differ considerably due to the different order of magnitudes and dynamic signal ranges in voltage and current channels.

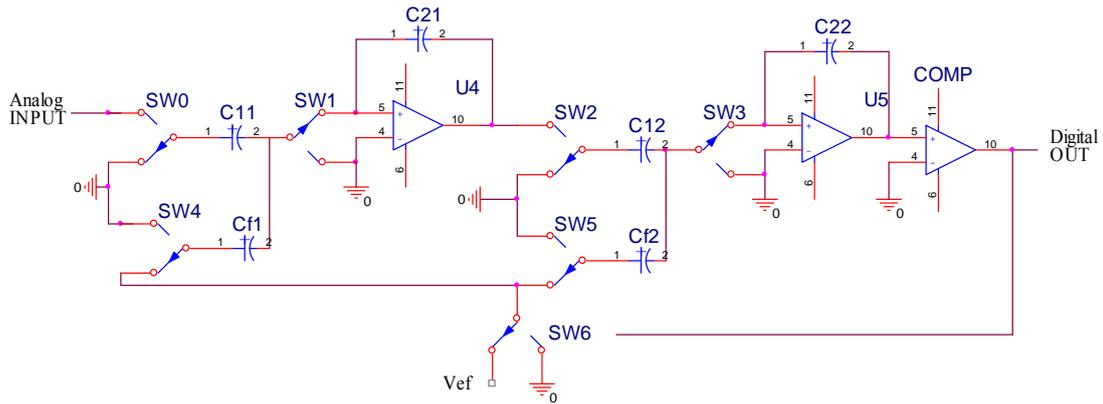


Fig. 5. Second-order $\Sigma\Delta$ modulator realized in SC architecture suitable for standard CMOS

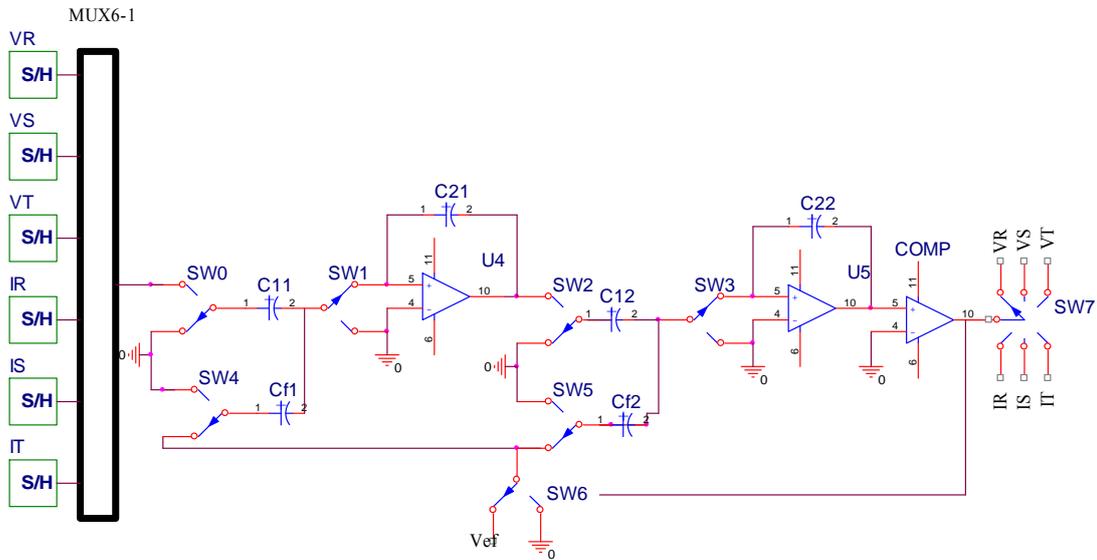


Fig. 6. Single second-order $\Sigma\Delta$ modulator with six multiplexed inputs/outputs

The third solution for multiplexed ADC suitable for the integrated power meter consists of 3 to 1 multiplexer. The system requires two identical blocks. One for three voltages (for each of three phases) while another being for three currents. Therefore it fits well to the general architecture presented in Fig. 3 because voltage and current channels are separated. This pursues the natural data flow to distinct digital filters and driven by different dynamic ratio in current and voltage channels. Moreover, for fixed sampling rate of 524288 Hz this requires moderately increased switching frequency of 1.57MHz. The overall area will be shrunken to 1/3 of the solution presented in Fig. 3.

IV SIMULATION RESULTS

The former architecture has been thoroughly verified by simulation.

Firstly, behavioral VHDL-AMS model has been developed and confirmed. It is used to check timing for S/H and two-phase clock signals needed to switch modulator.

Figure 7 presents the accepted timing of clock

waveforms. SH denotes switching in S/H circuits. Low logic level corresponds to the sampling phase while the high logic level defines the hold status. C1 and C2 denote controlling switching signals within modulator.

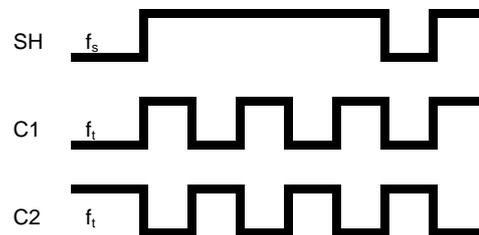


Fig. 7. Timing diagram for driving signals in three input multiplexed ADC

During high level of C1 input signal drives C11 and C12, while during C2 high the charge is transferred to C21 and C22.

Simulations confirmed expected functionality. Fig. 8 illustrates the obtained results.

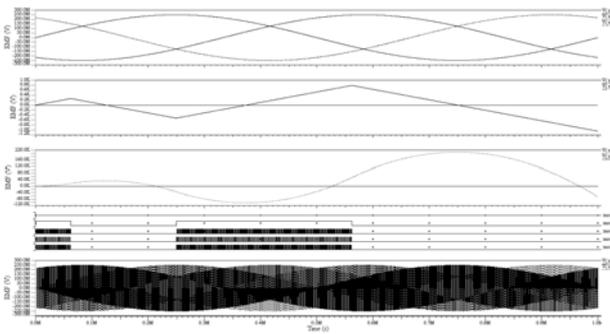


Fig. 8. Simulation results obtained for three channel architecture

V CONCLUSION

Three architectures for multiplexed $\Sigma\Delta$ ADC suitable for implementation within a three-phase solid-state power meter were discussed. All of them rely on using sampling-and-hold circuits at inputs to synchronize sampling of all input signals. The goal was to use as much of already designed analog and digital blocks as possible. Therefore it is necessary to sample SH circuits with the frequency of 524288 Hz and to obtain oversampled digital output with the same rate. This request can be fulfilled only if modulator switching frequency is increased as much as multiplexed signals are driven into it. This opens the issue of performances of used operational amplifiers and the issue of reliability of the overall SoC. Power meter is expected to work persistently and reliably for at least ten years. Therefore it is better to design them to run in modest operation conditions. Timing properties of operational amplifier are crucial for achieving high frequency switching rate of modulator. Eventually it is better to trade small amount of chip area for more robust and reliable component.

As the modulator is realized in SC manner, the increase of switching frequency reflects to the component size. Namely resistors are realized as SC. Hence their value is defined with $T/2C$, where T is $1/f_{sw}$; f_{sw} being the switching frequency. Consequently the increase of f_{sw} requires smaller C for the same R . From other side, the bandwidth of the integrator is defined by RC product. Remaining the same RC can be achieved by decreasing C . In both cases dimensions of one of the implemented capacitors will be decreased. Moreover coefficients a_i in Fig. 4 are defined as ratio of corresponding capacitances. Therefore if one is decreased, the other should be shrinking as well. Fortunately all this leads to the very slight modifications of layout. Namely, due to their dimensions, all capacitors are laid out as matched structures out of the modulator areas. This satisfies the basic request to reuse already designed macrocells.

However, the existing, previously tested opamp has precisely defined GBW and slew rate. This put boundary to the maximum switching rate.

Finally, the chosen solution is based on using 3 to 1 multiplexed ADCs. One is used for three voltages, another for the corresponding currents. This architecture match to the overall SoC concept that consider voltages and currents separately in order to protect sensible current channels from possible crosstalk produced in voltage channels. Moreover, this fits to the basic layout rule stressed in [9] "if it looks nice, it will work".

ACKNOWLEDGMENT

This work was supported by The Serbian Ministry of science and technology development within the project TR 11007.

REFERENCES

- [1] Andrejević, M., Savić, M., Nikolić, M., and Anđelković, B., „Top-Level layout design of solid-state energy meter”, Proc. of ETRAN 2004., Vol. I, ISBN 86-80509-49-3, pp. 13-16.
- [2] Sokolović, M., Nikolić, M., Andrejević, and M., Petković, P., “ADC Testing of an Integrated Power Meter”, Proc. of 5th Symposium industrial electronics, INDEL 2004, Banja Luka, 11-12. November 2004., ISBN 86-7122-014-1, pp. 132-137
- [3] Milovanović, D., Savić, M., and Nikolić, M., „Second-Order Sigma-Delta Modulator in Standard CMOS Technology”, ETRAN 2004, Proc. of ETRAN 2004., Vol. I, ISBN 86-80509-49-3, pp. 17-20.
- [4] Petković, P., Litovski, V. „Concept of integrated power meter“, (in Serbian), 13th International Symposium On Power Electronics, Novi Sad, 02.11.-04.11., 2005, T4-4.6, pp.1-5.
- [5] Marinković, M., Anđelković, B., and Petković, P., „Compact MAC Architecture of FIR Filters in Solid-State Energy Meter“, Proceedings of IEEE Region 8 EUROCON 2005 Conference, Beograd, 21.11.-24.11., 2005, pp. 1683-1686.
- [6] Nikolić, M., „Layout Design of Mixed-Signal CMOS Integrated Circuits“ (in Serbian), Master thesis, Faculty of Electronic Engineering Niš, Serbia, 2006.
- [7] Yunus, M., „Multiplexed Sigma-Delta A/D Converter“, US Patent No. 5150120, Sept. 1992
- [8] Conroy, C., Kim, B., and Erdogan, O., „Multiplexed ADC for a Transceiver“, US Patent No. 7203222B1, Apr. 2007.
- [9] Saint C., and Saint J., “IC Mask Design Essential Layout Techniques”, McGraw-Hill, New York, 2002. ISBN: 0-07-138996-2.

AUTHOR INDEX

| | |
|-------------------------|----------------|
| Andrejević Stošović, M. | 71 |
| Anđelković, B. | 71 |
| Bogdanović, M. | 23 |
| Bojanić, S. | 13 |
| Damnjanović, M. | 49 |
| Davidović, N. | 23 |
| Dimitrijević, M. | 71, 81 |
| Đorđević, G. | 75 |
| Đorđević, S. | 13 |
| Đošić, S. | 56 |
| Filiposka, S. | 27 |
| Jevtić, M. | 56, 61 |
| Jovanović, Bojan | 61 |
| Jovanović, Borisav | 49 |
| Krstić, A. | 23 |
| Litovski, V. | 33, 42, 71, 81 |
| Maksimović, D. | 6 |
| Milojković, J. | 42 |
| Mirković, D. | 89 |
| Nieto-Taldriz, O. | 13 |
| Notermans, G. | 6 |
| Paunović, I. | 17 |
| Petković, M. | 75 |
| Petković, P. | 86, 89 |
| Rajčić-Vujasinović, M. | 67 |
| Stanimirović, A. | 23 |
| Stević, Z., | 67 |
| Stanojlović, M. | 86 |
| Stoimenov, L. | 23 |
| Topisirović, D. | 67 |
| Trajanov, D. | 27 |
| Zerbe, V. | 17 |
| Zwolinski, M. | 1, 49 |