



BEC 2004

Baltic Electronics Conference

Post-Graduate Student Session

Tallinn University of Technology
October 3-6, 2004
Tallinn, Estonia

ORGANIZED BY:



Department of Electronics
of Tallinn University of Technology

SPONSORED BY:



IEEE - The Institute of Electrical and
Electronics Engineers



CYBERNETICA



IN COOPERATION WITH:

Faculty of Information Technology

C O N T E N T S

Post-Graduate Student Session

A Compact DSP Performance Testing System for the Academia.....	3
Andres Mellik, Tallinn University of Technology, Estonia	
Mini-measuring system with dynamic software adjustment based on zooming delta-sigma ADC.....	5
Vladimir Kaulio, Saint-Petersburg State Polytechnical University, Russia	
Moving studies to e-environments – the approach of e-EDU.....	7
Tarmo Robal, Tallinn University of Technology, Estonia	
Creating Graphs for Optimization Tasks.....	8
Indrek Roosileht, Tallinn University of Technology, Estonia	
Heuristic Algorithms for Weighted Graph Partitioning.....	9
Allan Aasma, Tallinn University of Technology, Estonia	
Sorting of bite vectors with help of self-organizing maps algorithm.....	10
Uljana Boiko, Tallinn University of Technology, Estonia	
Deductive Fault Simulation on Structurally Synthesized BDDs.....	11
Margit Aarna, Raimund Ubar, Jaan Raik, Tallinn University of Technology, Estonia	
Graphical User Interface for Turbo Tester Toolset.....	12
Elmet Orasson, Tallinn University of Technology, Estonia	
An Underwater Vehicle Able to Follow a Bottom Profile Using the Sonar.....	13
Deivid Pugal, Tartu University, Estonia	

A Compact DSP Performance Testing System for the Academia

A. Mellik¹

¹Department of Electronics, TTU, Ehitajate tee 5, 19086 Tallinn, Estonia, E-mail: amellik@yahoo.com

ABSTRACT: The present article is a proposal for introducing a way of integrating several tools of electrical and computer engineering education, existent in academic institutions, to a compact and complete DSP performance testing environment, enabling CS/EE/CE students to gain experience in related fields through simply combining some of the existing hardware and software.

1 Introduction

Tools like LabVIEW, Matlab/Simulink and Texas Instruments Code Composer Studio with DSP development boards (e.g. TI C6711 DSP Starter Kits) are employed widely among universities.

Till now most of these tools have been used in their specific domain: LabVIEW for measurement, data acquisition and testing, Matlab/Simulink for running simulations and control models, CCStudio in programming DSP-s.

2 Approach

Combining the functionality of those and possibly several other environments would create synergies worth investigating. One of these synergies is given as an example of such integration below.

National Instruments strongly supports integration with tools from other leading vendors and the open source community (incl. <http://www.openg.org>). Many of these tools are a *de-facto* standard in the academic community. Adding the fact that LabVIEW, Simulink, etc all run on Linux-systems and that LabVIEW can communicate with Perl scripts, we can essentially drive the cost of such integrated engineering education environments down to the cost of academic licenses of the mentioned tools.

Moreover, as LabVIEW essentially stands for *Virtual Instrumentation*, the need for actual hardware becomes less important. Digital and analog I/O modules are necessary for interacting with real-world (test-)objects (which could also be emulated till the prototype phase), but signal generators, oscilloscopes, etc are easily emulated in software.

The additional software developed is straight-forward and modular, enabling to get started with a minimum effort.

As such commercial systems usually employ high-end hardware and in-house developed software they tend to be inaccessible to the academia in terms of corporate interests and financial resources available.

An important side effect is that students with different backgrounds can actually utilize best their specific skills and

satisfy their academic and professional interests. Computer science majors can work on code optimization either in C/C++ or assembler using CCStudio. Students familiar with Matlab can experiment with various algorithms from a wide variety of domains in digital signal processing (the Least-Mean-Square algorithm for adaptive noise cancellation is given as an example below). Electrical engineering students can develop the concept further by introducing some customization and/or extra features to the present implementation. The latter can be realized in LabVIEW or through an imported DLL or ActiveX component.

The author proposes to integrate the system as a lab assignment into a DSP, DAQ/Measurement, Performance Evaluation or similar courses either at 3rd-4th year bachelor or master level studies (in order to benefit the most from the process, the students should have sufficient background knowledge.)

Course projects could use the system either as the environment for experiments or develop the system further by adding functionality.

Moreover, the students can compare different approaches to the task at hand. That in itself is a suitable learning object, investigating and understanding the benefits of system integration and the modularity involved. LabVIEW is a perfect environment for experimenting with those subjects. It enables the student to see the not always explicit connections and relations between different tools in a systematic way. By introducing modularity they can begin with simpler sub-tasks, while always preserving the structure of the system. Finally, the student is expected to see the synergies, which appear from such an approach and more importantly, to find new ones.

A prototype implementation of such a system exists (described below), developed within a master thesis at the Advanced Learning and Research Institute, University of Lugano (<http://www.alari.ch>) in co-operation with National Instruments and Texas Instruments.

That implementation is a sufficient starting point for proof of concept and future developments.

A detailed description highlighting the integration of LabVIEW with other educational tools is given further.

3 System Description

Till now, testing the code running on a DSP and derivatively the performance of the algorithm-architecture combination included probing the individual registers on a

general purpose DSP. The newly released tools from National Instruments ease that process by a significant margin in terms of test time.

The system aims at easing the test development and operation of DSP performance testing. The code running on a DSP is compared to a Matlab model executing in parallel on the PC. The model DLL communicates with LabVIEW via the Simulation Interface Toolkit. Interaction with the DSP is carried out by the DSP Test Integration Toolkit, which automates the IDE functions of TI Code Composer Studio (CCS) and RTDX-communication with the DSP development board (TI 6711 Starter Kit).

The outputs of both are compared by the test module and thus it is easy to detect any differences in the behavior of the two, which is dependent on code optimization for both the C++ code generated by Real-Time Workshop in Matlab, the DSP code and the specific target architecture.

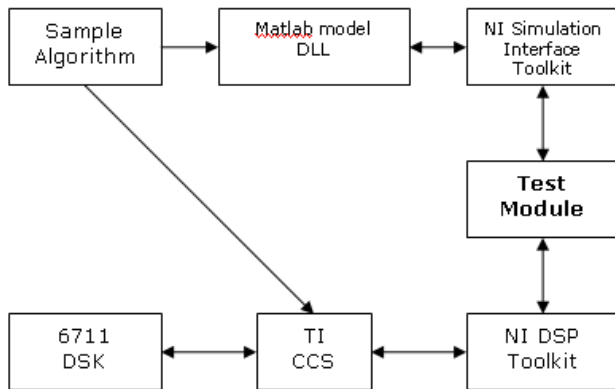


Figure 1 – System Architecture

The current solution is targeted at TI C67XX series DSPs, though the concept can be applied to virtually any custom DSP, as well. In that case the CCS and NI DSP Test Integration Toolkit are replaced by digital I/O hardware (i.e. NI 6533/6534). Another possibility for integration at universities is to program an FPGA to perform DSP functions.

For instance, Xilinx has released two FPGA families (*Virtex-II Pro* and *Spartan-3*) for DSP designs.

The test modules can either be executed stand-alone or put in a test sequencer for iterative algorithm testing and comparison.

4 Sample Application

Essentially the algorithm can be anything that one can model in Matlab using NI Sinks and Sources for inputs and outputs, provided the DLL builds fine and operates correctly in LabVIEW.

An LMS algorithm for adaptive noise cancellation was chosen as the sample application for the system.

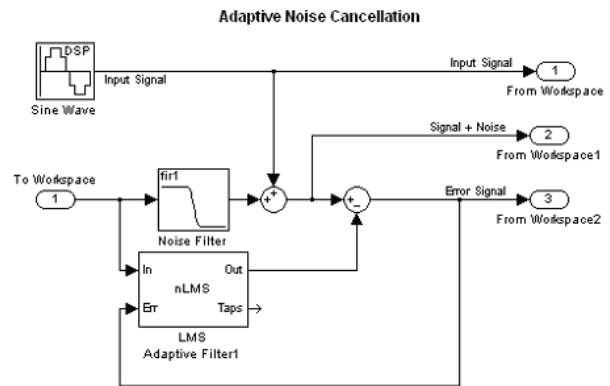


Figure 2 - An LMS algorithm

Notice, that in the above Matlab model the inputs and outputs of the system are replaced by components from “NI Sinks and Sources”.

After such a model is complete, Real-Time Workshop is used to generate the corresponding C++ code. In order to maintain better grasp of the process, first choose to generate only code in Matlab and later compile and link the DLL in VC++ manually. LabVIEW DLL Target (*nidll.tlc*) is to be chosen as the Target Configuration in Real-Time Workshop. The Simulation Interface Toolkit automatically generates a LabVIEW driver VI (Virtual Instrument), which can then be customized freely to meet the requirements of the developer.

Communication with the DSP development board is handled via RTDX. LPT was used as the actual hardware channel in this implementation, but the use of JTAG is proposed for future developments. Nevertheless, in an academic environment LPT should be considered sufficient.

5 Conclusions

The proposed solution has in practice proven its viability. Together with scalability for future developments, such as power consumption measurements, such approaches can be effectively adopted for various embedded devices.

In the given example, the results obviously heavily depend on the level of code optimization done for both the code running on the DSP and the DLL. With the latest DSP clock speeds reaching 1 GHz, the DSP likely outperforms the PC in terms of the specific application.

The applied solution is effective in reducing the time required for performance-tuning of DSP devices.

Mini-measuring system with dynamic software adjustment based on zooming delta-sigma ADC

V. Kaulio

*S.-Petersburg State Polytechnical University, Polytechnicheskaya str., 29, 195251 S.-Petersburg, Russia,
E-mail: v_o_v@yahoo.com*

ABSTRACT: the application of modern integral technologies in microprocessor electronics realized on Sensing Machine XE88LC05 with integrated zooming $\Delta\Sigma$ ADC to increase the accuracy of force measurements.

1 Introduction

Delta-sigma analog-to-digital converters ($\Delta\Sigma$ ADCs) are widely used in measuring channels of information-measuring systems to measure slight electrical signals. Measuring $\Delta\Sigma$ ADCs differ from other converters in high stability and accuracy of analog-to-digital conversion. The resolution of large-scale manufacture $\Delta\Sigma$ converters achieves 24 bits [1].

The tendency of modern measuring systems developing is aimed to the integration of adjustable analog conventional converters, $\Delta\Sigma$ ADCs and powerful microprocessors on the chip. The advances of that way are decreasing the size and power consumption of measuring system and unification of electrical characteristics of elements, for example, to provide the temperature compensation.

Modern force measurements as many other require flexibility of parameters: accuracy, measuring range, speed and etc. The using of mixed signal system on chip XE88LC05 with integrated Zooming $\Delta\Sigma$ ADC by XEMICS is described in this paper.

2 Structure of the mini-measuring system

The sources of input signal (V_{in}) in presented measuring system [Fig.1] are strain gauges with the voltage sensibility 2mV/V. It requires an amplification of analog signal before the analog-to-digital conversion. XE88LC05 has the zooming $\Delta\Sigma$ ADC constructed from the programmable analog conventional amplifier and $\Delta\Sigma$ ADC based on the second order $\Delta\Sigma$ -modulator and digital filter. The signal is built-up with effect of ZOOM:

$$V_{inADC} = V_{in} \cdot PGA1 \cdot PGA2 \cdot PGA3 - V_{off2} \cdot PGA3 - V_{off3} \quad (1)$$

where V_{inADC} – analog signal at the input of $\Delta\Sigma$ ADC, PGA_i and V_{off_i} – analog gain and offset. That approach allows the best way to adjust the measuring scale.

The values of registers and variables, need to measure at every channel, are stored in external EEPROM-memory at the step of the mini-system calibration and loaded through the initialization after power on. The programmable adjustment of such parameters as type of channel, measuring range, accuracy and speed of conversion are made dynamically before the measurement. It allows create flexible measuring systems for different types of sensors.

The system has two processing modes: measuring-calibration and measuring-control. The mode is defined in first 200ms after the power supply. The presence of 50 pulses at the input of digital input of system indicate go to the measuring-calibration, else system works in measuring-control mode. In the measuring-calibration mode the control of the system is executed by RS-232 interface. That mode is used to test of quality of measuring channel and calibrate sensors. Digital output may be switched off to debug with digital control only.

In the measuring-control mode measuring works stand-alone and digital control switched off to prevent some noise influence to the algorithm of execution. Physical value is converted by the strain gauge to the analog electrical value – voltage, corrected through analog conventional converter, sampled to N-bits output code of $\Delta\Sigma$ ADC, corrected through digital linear converter, saved to internal registers and goes to the internal or external DAC. Then that operation is repeated for the next channel. Measurements are executed in series switching the ADC and DAC channels with the dynamic software adjustment. If the adjustments of active channel are different from the previous one, parameters of analog chain must be changed.

3 Dynamic adjusting

The control of adjusting is realized through a special developed author's software for the personal computer and the microprocessor embedded in XE88LC05. The addition software digital calibration of measuring range developed to increase the accuracy of range borders setup. Every measurement is executed in three steps: linear conversion of input analog signal in embedded

analog conventional converter, analog-to-digital conversion in $\Delta\Sigma$ ADC, software linear conversion of digital sample. The coefficients of analog and digital chains are calculated in the computer.

The control of adjusting is realized with software of personal computer and embedded in XE88LC05 microprocessor. The addition software digital calibration of measuring range added to increase the accuracy of range borders setup. Every measurement is executed in three steps: linear conversion of input analog signal in embedded analog conventional converter, analog-to-digital conversion in $\Delta\Sigma$ ADC, software linear conversion of digital sample. The coefficients of analog and digital chains are calculated in personal computer.

Coefficients for the analog linear conversion (1) are calculated in two steps:

- calculating of accurate coefficients (2) and (3):

$$A_{adc} = \frac{0.8V_{ref}}{U_n - U_0} \quad (2)$$

$$B_{adc} = -A \cdot U_0 \quad (3)$$

- searching of tabulated register values for coefficients (4) and (5):

$$A_{adc} = Vin \cdot PGA1 \cdot PGA2 \cdot PGA3 \quad (4)$$

$$B_{adc} = -Voff2 \cdot PGA3 - Voff3 \quad (5)$$

The digital linear conversion of input signal ADC_OUT is calculated in formula (6):

$$X = A \cdot ADC_OUT + B \quad (6)$$

where X – result of one time measurement. Coefficients A and B of the linear conversion calculated in formulas (7) and (8):

$$A = \frac{U_{min} - U_{max}}{U_0' - U_n'} \quad (7)$$

$$B = U_{min} - A \cdot U_0' \quad (8)$$

Target minimal U_{min} and maximal U_{max} values of output signal should be set in computer software by user. Signal values U_0 and U_n are measured under the zero and nominal load to the sensor with the analog gain equal to 100 followed by software division into 100 to minimize the offset error of embedded analog amplifiers. Signal values U_0' and U_n' are measured in the same way after the correction of the analog linear conversion coefficients in accordance with formulas (4) and (5). The coefficient values are duplicated on the hard disk of computer in a special file of the measuring transaction.

4 Software development

The software part of the system realized as a single-document Windows-application for personal computer based on "document/view" architecture on C++ language using Microsoft Foundation Classes library in Visual Studio 6.0 under Win32 platform. Decoding, processing and graphic presentation acquired data are executed in real-time mode by several parallel software threads.

The control of threads organized through the message exchange between thread functions. To supply the reliability it also uses embedded synchronization methods and critical sections lock of program code. Measured data from 2 channels are presented in digits and diagrams from the number of measurements. User can select the range of viewed data and scale transformation to convert measured data samples to the physical values.

Moving from the real-time measuring to the processing phase, measuring data are presented in table form to store it on hard disk or copy to Excel or MATLAB applications via the clipboard. During the processing phase user have a possibility of navigation inside of measured data to find some specific signal phenomenon.

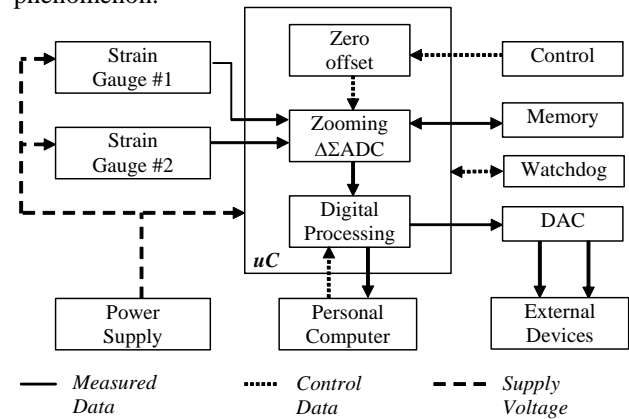


Fig. 1. Structure of the mini-measuring system.

5 Conclusion

The result of application XE88LC05 is a compact two channels force measuring system. The presence of integrated analog conventional converter is one of cardinal features of this development, because the dynamic software adjustment of its parameters allows use integrated zooming $\Delta\Sigma$ ADC most effectively for differential signals from strain gauges. Also it provides a dynamic zero offset without the measurement accuracy reduction. Presented mini-system has the high speed and the low power consumption, XE88LC05 consumes 300uA at 1MIPS. So that system may be used a long time in stand-alone mode in mobile instruments with battery power supply. Because of $\Delta\Sigma$ ADC has a possibility of changing the analog-to-digital conversion time to the accuracy, the resolution of measurements is in range 10...16 bit, while the input signal is in range 2mV... 5V.

References

- [1] V.V.Kaulio "The development of digital filters in delta-sigma analog-to-digital converters for measuring channels". PhD Dissertation. SPbSPU, S.-Petersburg, 2003.
- [2] Datasheet XE88LC05. "Data acquisition microcontroller", XEMICS SA, www.xemics.com, 2002.

Moving studies to e-environments – the approach of e-EDU

Tarmo Robal

Department of Computer Engineering at Tallinn University of Technology

tarmo@pld.ttu.ee

The rapid development of several software technologies has made its impact to today's teaching means as well. Web-based teaching environments are coming more popular. E-learning offers new possibilities for attaining knowledge, where time and distance are not the issue any more. Without technologies and tools available, e-learning would have no possibilities for providing education. Also, traditional learning is moving towards the use of web-based tools.

There have been many different educational and learning support systems, some of them have failed, and some have been more or less successful.

Over the years, universities have developed several different kinds of information systems, including systems that were supposed to support learning processes via providing materials, links and other information valuable during an educational course. The general practice among lecturers has been to use available tools or in case they do not satisfy lecturers' real needs, develop their own systems to manage tasks, study results, materials. In the following, a brief overview of an online learning support service environment e-EDU is provided.

The e-EDU [1] stands for a service environment being developed at the Department of Computer Engineering at Tallinn University of Technology (TUT). Its purpose is to provide support services for students and lecturers during study processes. The system provides a general approach to solve problems like course registration management, students management, course materials management, assignments and results management, taking into account the real needs of the lecturers of TUT. The e-EDU system proposes a new and localized approach as a service to enliven daily studies at the university as well as offers means for distance learning. The system is developed according to the proposed methodology of RUP technology [2].

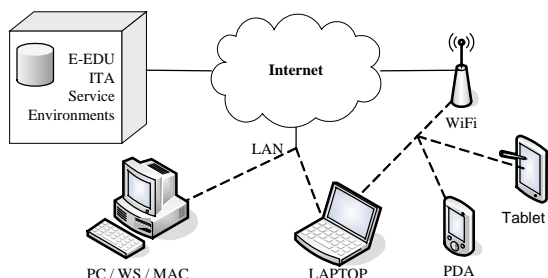


Fig. 1. Accessing the services of e-EDU

Today, e-EDU is being used as an active learning support system in seven subjects taught by the department. The subjects are Informatics I and II, Computers I and II, Digital Systems Design and Test, Testing of digital systems, Fault-tolerant systems.

One of the advantages of the e-EDU system is that it is a web-based system with many interfaces. Therefore, it can be accessed from almost everywhere (Fig.1), regardless of whether you are using a PC, Workstation, PDA, or Laptop. The system is built up on the assumption of service-based architecture in a way that several interfaces are supported. For example, lecturers do not need to have any student information on paper, instead they have their PDA, Laptop or other portable device and can have an access to the IS via LAN or WiFi. The majority of premises of Tallinn University of Technology are covered with WiFi network, which is a good prerequisite for such a system development and exploitation.

The e-EDU system is targeted to three user groups; each of them has a different view (GUI):

- The students' view, as a web application edu.pld.ttu.ee;
- The teachers' view, as a part of the department's intranet application ITA;
- The administrator's view, as a part of the ITA intranet application;

The functionality provided by the e-EDU is as follows:

- User authentication via a general authentication system or using the Estonian Citizen ID-card [3];
- Course registration management;
- Registered students management;
- Course material management;
- Assignments and students' results management;
- Communication between academic personnel and students;
- Tests generation and crediting;
- General course management with archiving and back-up facilities – administrators only;

The e-EDU system provides a new and localized approach to enliven the daily studies at the university. In addition, it provides means for distance learning as well as for e-learning. It also lessens the workload of teachers, providing general means for tasks, materials and results management among with other tools.

References

- [1] Robal, T., Kalja, A. e-EDU – An information system for e-learning services. Scientific Papers University of Latvia, 2004, Vol.672, pp 469-480.
- [2] Robal, T., *et al.* The Rational Unified Process with the "4+1" View Model of Software Architecture – a Way for Modelling Web Applications. In.: Proceedings of the Fifth International Baltic Conference, BalticDB&IS 2002, Tallinn, June 3-6, 2002, Vol. 2., pp.119-132
- [3] Kalja, A., Vallner, U. Public e-Service Projects in Estonia. In.: Proceedings of the Fifth International Baltic Conference, BalticDB&IS 2002, Tallinn, June 3-6, 2002, Vol. 2., pp. 143-154.

Creating Graphs for Optimization Tasks

Indrek Roosileht

Tallinn University of Technology, Department of Computer Engineering

My work is about transforming algorithms to another form that is suitable for optimization, and comparison and analysis of these forms. It is used to improve an experimental high-level VLSI synthesis tool.

Algorithms are generated from a hardware description language files (for example VHDL). Suitable forms for optimization are weighted graphs.

As an example for register optimization task, registers could be as graph nodes and register life-cycle conflicts could be as incompatibility edges and compatibility between registers with nonoverlapping lifetimes that are connected to the same type of functional units as priority edges.

Example:

The square-root approximation of two signed integers a and b by the formula:

$$\sqrt{a^2 + b^2} \approx \max((0.875x + 0.5y), x) \text{ where } x = \max(|a|, |b|) \text{ and } y = \min(|a|, |b|).$$

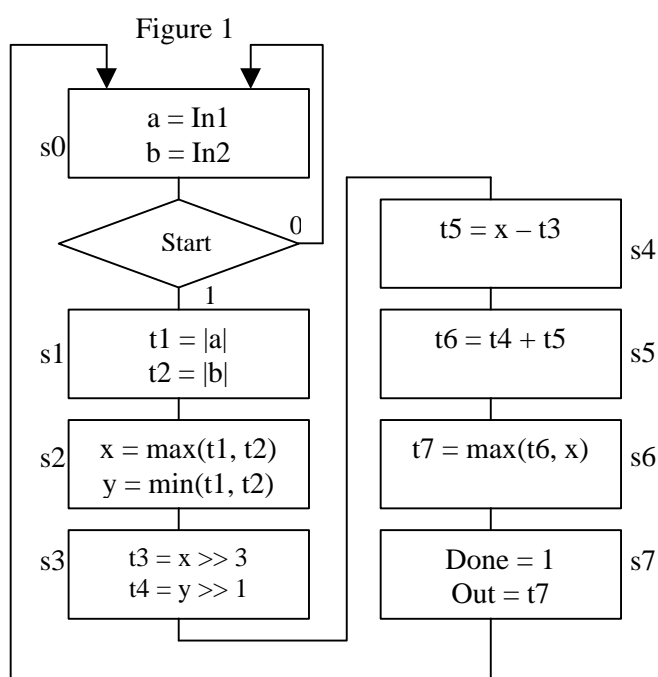
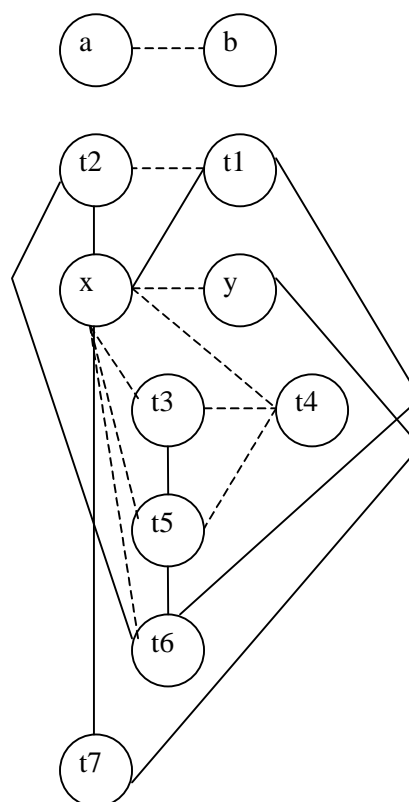


Table 1

	a	B	t1	t2	x	y	t4	t3	t5	t6	t7
s0											
s1	X	X									
s2			X	X							
s3					X	X					
s4					X		X	X			
s5					X		X		X		
s6					X					X	
s7											X

Figure 2



References

[1] Daniel D. Gajski, "Principles of Digital Design"

Heuristic Algorithms for Weighted Graph Partitioning

Allan Aasma

Department of Computer Engineering, Tallinn University of Technology

Abstract

Graph partitioning algorithms are used for several optimization tasks in digital design process. For example, one of the major tasks in datapath optimization involves grouping variables with nonoverlapping lifetimes to common register location. Grouping registers reduces the overall number of storage components.

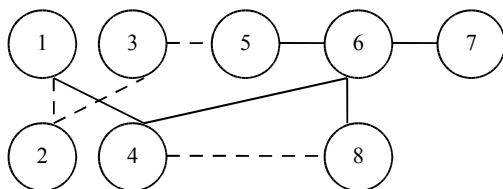
Introduction

The object of the thesis is to implement different weighted graph partitioning algorithms in program (using programming language C++). The goal is to find as fast and as effective solutions as possible. Those algorithms must be added to existing program named PAINTER to expand its features with graph partitioning methods.

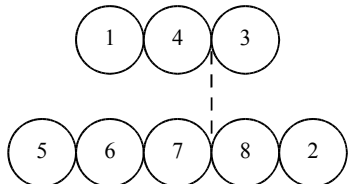
The second goal is to analyze the results of different algorithms and give the effectiveness rating to each algorithms that were used with a given input graphs.

Graph partitioning example

In the following simple example the initial conflict graph consists of nodes and edges. Each node represents a variable and each edge between two nodes represents compatibility or incompatibility (dashed line) in merging those two variables. The task is to partition the given graph to minimize the number of registers.



Initial conflict graph



Final conflict graph

After graph partitioning the final conflict graph needs only two registers (initial graphs has 8 registers).

Algorithms

In this thesis, optimization methods that use different Greedy based algorithms are implemented and analyzed. Greedy based algorithms search at each step the local best result (local minimum). Such methods are very fast, but the final results are not always close to the overall global minimum.

To find the global minimum each possible partitioning combination must be tested, but such solution is in practice not reasonable (testing all possible combinations

takes very much time). The full search algorithm is also implemented in the program – where the partition with the minimum cost are searched from all possible partitioning results.

Two different Greedy based algorithm variants are already implemented:

- 1) the node which has the biggest cost are selected first;
 - 2) the node which has the smallest cost are selected first.
- In addition there will be a solution which partitions the given graph in random sequence.

Extra possibilities

One of the main purpose is to make the implemented algorithms as optimum and as fast as possible – to get the solutions that would be close to the global optimum. In order to get such results partitioning algorithms need very accurate cost function. The cost function indicates the cost of particular design solution.

For example (typical Greedy algorithm):

- 1 step: sort the nodes by given sequence requirement
- 2 step: calculate the initial cost
- 3 step: set the pointer to the first cluster (at the beginning there is one cluster per each node)
- 4 step: find from all other non-conflicting nodes the partition which has the smallest cost
- 5 step: if there is a non-conflicting partition with smaller cost than the previous best one – step 6 else step 7
- 6 step: partition the graph (modify clusters), calculate the new best cost and go back to the step 4
- 7 step: increase the cluster pointer and go back to the step 4 (if it is the last one then algorithm END)

From this example we can see that all partitioning results are depending from the cost function output. Therefore it is very important to make the cost function very accurate. At the moment one extra feature (multi-dimensional cost vectors) are under research to present addition information to cost function.

Cost function parameters:

- 1) two nodes can be in conflict (their lifetime matches at some moment in time);
- 2) their weight (in bits) are different;
- 3) their types are different;
- 4) nodes can be connected to the same bus (the register is the input to the operation or reverse);

At the moment the possible gain from the last point of previous list are under research – using priority weights to indicate that kind of information to cost function (it should give some overall savings as the design needs smaller number of MUX).

Sorting of bite vectors with help of self-organizing maps algorithm

Uljana Boiko

*Tallinn Technical University
Dept. Computer and System Engineering*

The task for this work is to sort byte vectors. Vectors sorting should be done according to the hamming distance. The less hamming distance between 2 byte vectors the better they suit to place near each other. The goal is to reach such order of vectors, where the sum of hamming distances between every 2 vectors would be minimum. This task could not be solved with simple algorithm. Different heuristics are needed. During this work one of the neural network algorithms - the self-organizing maps algorithm is going to be tested for solving this task. Self-organizing maps algorithm is non-deterministic, therefore with the same initial conditions it could give different solutions. During implementation of this algorithm for solving this task one-dimensional space is used. Number of neurons in the map would be the same as number of vectors to sort. Every vector would have a set of parameters, where some of them would be dynamic and others static. Static ones are used to show the hamming distance between every 2 vectors. The dynamic parameters show the distance between neurons according to total hamming distance of vectors in definite order. During the computation neurons change their positions in one-dimensional space trying to find order with minimum hamming distance. The aim of this work is to define structure of the self-organizing maps algorithm to solve byte vectors sorting task.

The main features and steps of the self-organizing maps algorithm are the following. First of all required number of teaching steps is at least 1000 times the number of vectors it is needed to sort. All the dynamic parameters in initialization phase are chosen randomly within given range. One of the dynamic parameters could initially be just index of the array where certain vector is located. Then the map is needed to learn. The random vector is chosen among all the vectors in array. This vector is compared with all vectors in array with a help of Euclidian distance and the vector, which has the minimum Euclidean distance between it's own and "input" vector, is put as a "winner". Then the dynamic parameters of the "winner" are changed according to special function. The aim of this work is to find this function, which should give desired results. Also the neighborhood area is specified. At the beginning of learning phase the neighborhood area is big and during the learning time the neighborhood area converges to one "winner" vector. And after every step the vectors: "winner" and vectors in neighborhood area – are changing their position in array according to the dynamic parameters. At the end of the learning phase byte vectors must be located so that they would give the min hamming distance between every 2 vectors.

Deductive Fault Simulation on Structurally Synthesized BDDs

M.Aarna, R.Ubar, J.Raik

Tallinn University of Technology, Department of Computer Engineering

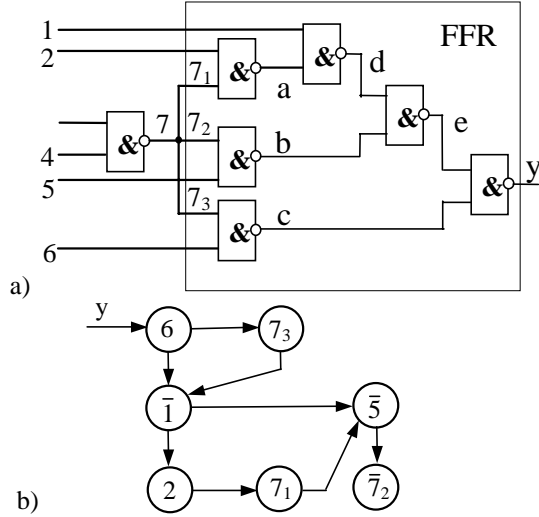


Fig. 1. Logic circuit and its SSBDD

A BDD that represents a Boolean function $y=f(X)$, $X = (x_1, x_2, \dots, x_n)$, is a directed acyclic graph $G_y = (M, \Gamma, X)$ with a set of nodes M and mapping Γ from M to M . Set M consists of two types of nodes: internal (non-terminal) M^N and terminal M^T , that $M = M^N \cup M^T$.

In our approach we have omitted the 0- and 1-terminal nodes from the BDD description and 0- and 1-labels from the edges. Exiting the BDD downwards corresponds to $y=0$ and rightwards to $y=1$, respectively. Downward edges correspond to 0-edges and rightward edges correspond to 1-edges. In addition, nodes are labelled by inverted or non-inverted variables (See Fig. 1b!).

Let us denote the variable associated with node m as $x(m)$. Then m^0 is the successor of m for the value $x(m) = 0$ and m^1 is the successor of m for the value $x(m) = 1$. By the value of $x(m) = e$, $e \in \{0,1\}$, we say the edge between nodes $m \in M$ and $m^e \in M$ is *activated*. Consider a situation where all the variables $x \in X$ are assigned by a Boolean vector $X^t \in \{0,1\}^n$ to some value. The edges activated by X^t form an *activated path* $l(m_0, m^T)$ from the root node m_0 to one of the terminal nodes $m^T \in M^T$.

We say that a BDD $G_y = (M, \Gamma, X)$ represents a Boolean function $y=f(X)$, iff for all the possible vectors $X^t \in \{0,1\}^n$ a path $l(m_0, m^T)$ is activated so that $y=f(X^t) = x(m^T)$.

Structurally Synthesized BDDs (SSBDDs) are a special case of BDDs, which are generated by a superposition procedure that extracts data about structural paths of the circuit differently from traditional BDDs generated by Shannon's expansion. Another difference between the classical and the SSBDD approach is that in SSBDDs we represent a digital circuit as a system of BDDs, where for each fanout-free region (FFR) of the circuit a separate SSBDD is generated.

SSBDD models for gate-level digital circuits are generated as follows. Starting from the output of the FFR (i.e. primary output or a fanout stem), logic gates are recursively substituted by their respective elementary BDDs. The procedure of superposition terminates in those nodes, which represent a primary input or a fanout branch. Fig. 1b presents an SSBDD for the logic circuit in Fig. 1a.

Definitions:

- M - set of nodes at the main activated path $l_{main} = (m_0, m^T)$
- $M_1 \subset M$ is the set of nodes belonging to l_{main} , where $\{m \in M_1 \mid x(m) = f(X)\}$
- $M_2 = M \setminus M_1$
- $m' = m^{x(m)}$ $m'' = \overline{m^{x(m)}}$
- $S(m)$ - fault list propagated to m from previous SSBDDs.

The goal of the deductive fault simulation algorithm is to calculate the set of faults R propagated through the SSBDD G_y by a given input vector X and $S(m)$.

Algorithm:

```

R = ∅
for each m ∈ M
    Lm = ∅
end for
for each m ∈ M
    if m ∈ M1 then
        if m'' = ∅ then
            Lm'' = Lm'' ∪ S(m)
        else
            R = R ∪ Lm
        end if
    else /* m ∈ M2 */
        if m' = ∅ then
            Lm' = Lm' ∪ (S(m) \ Lm)
        else if x(m) = f(X) then
            R = R ∪ Lm
        end if
        if m'' = ∅ then
            Lm'' = Lm'' ∪ (S(m) ∩ Lm)
        else if x(m) =  $\overline{f(X)}$  then
            R = R ∪ Lm
        end if
    end if
end for

```

References

- [1] R. Ubar, "Test Synthesis with Alternative Graphs," *IEEE Design & Test of Comp.* Spring 1996, pp. 48-59.
- [2] M.Aarna, J.Raik, R.Ubar. Parallel Fault Simulation in Digital Circuits, *Int. Conf. of Riga TU*, pp.91-94, 2001.
- [3] M.Aarna, et al.. Turbo Tester - Diagnostic Package for Research and Training. *Journal of Radioelectronics and Informatics*, No.3 (24), pp.69-73, July-Sept. 2003.

Graphical User Interface for Turbo Tester Toolset

Elmet Orasson (elmet@pld.ttu.ee), Tallinn University of Technology,
Department of Computer Engineering
Raja 15, Tallinn, Estonia

Turbo Tester (TT) is a set of command line programs, dedicated to digital test problems. Although the choice of user interface depends on users experience and personal taste there are quite many people who like point-and-click type working environments also known as graphical user interfaces (GUI).

Years ago, there was one TT GUI, written in Visual-C. The main problem was the fact, that this interface could be used only on Microsoft Windows operating system and nowhere else, thus new interface was designed using Java programming language. Main reasoning was that Java programs can be run without (or only with little modifications) on range of platforms – making it very simple to create GUI-s for different operating systems.

Another question was scalability – if TT toolkit will get additional tools for working then each of them must be accessible via GUI too. Programming new dialog window may sound easy but doing it for each tool separately certainly is not convenient. Solution to this problem was dialog generation subroutine where dialog setup will be read from (usually) small configuration file. Each TT tool needs one configuration file describing all its options and arguments. Although the dialog generator itself is not trivial it makes GUI quite flexible (it could even be used for executing tasks different from original goal). Consequently there are only few hardcoded menus and all dialog windows are generated from setup scripts. For more complicated dialogs there is a (yet theoretical) possibility of creating accompanying Java control class.



Illustration 1 - Main working window

Depicted on Illustration 1 is the main window consisting of menus, shortcut buttons on toolbar (also configurable), program output and status panels.

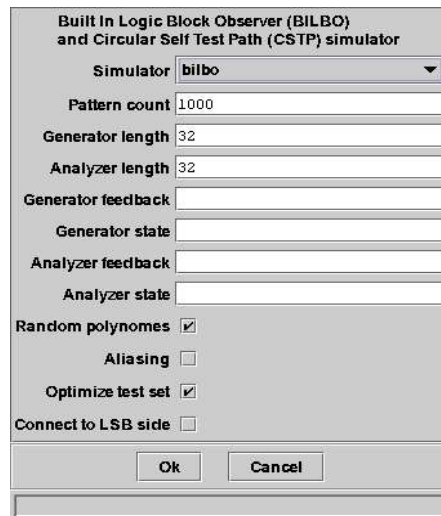


Illustration 2 - BILBO setup dialog

On Illustration 2 is one of the generated dialogs for Built-In Logic Block Observer tool.

An Underwater Vehicle Able to Follow a Bottom Profile Using the Sonar

Deivid Pugal*

Faculty of Chemistry and Physics, Tartu University, Tähe 4, 51010 Tartu, Estonia

The general purpose of our work is to develop an underwater vehicle which mission is to explore and film plantation on the bottom of the sea. The activity area is near the coastline of Baltic Sea in low depth.

Due to several reasons the underwater vehicle will be dragged by a boat. However it is very important to know distance between the vehicle and the bottom of the sea and other underwater objects to avoid close contact. Therefore we have to use few sonars to determine our vehicle distance from surrounding objects . One of problems is to find optimal positions and directions for those sonars so, that they would not influence each other and give maximum information while using minimal amount of detectors.

We use Devantech SRF08 sonars. They are originally designed to work in air, so we had to adjust them for underwater conditions. It includes changing transmitter and receiver with waterproof ones and includes reprogramming sonar's microcontroller because of the different sound speed in water.

The next problem is changing sonar's transmitter frequency. It might be useful for detecting different kind of surfaces on the bottom of the sea like rocky, muddy, sandy and etc. If the frequency could be changed main problem is recalibrating sonar and analyzing signals - how different materials reflect sound with different wavelength. So lot of measurements must be done for that.

All sonars must be controlled by a central unit. Because number of sonars, it is necessary to establish communication protocol to make them all working properly.

* Contact information: david@ut.ee, phone +372 55974170. Graduation date June 2005. Supervisors: Alvo Aabloo (Alvo.Aabloo@ut.ee) and Maarja Kruusmaa (Maarja.Kruusmaa@ut.ee)