Journal of EIectronic System and Programming

Electronic Systems and Programming Center

Journal

of

Electronic System

and Programming

Issue No: 1 - 2018

Editorial:

First Issue – Journal of Electronic System and Programming

On behalf of the Editorial Board, it is with great pleasure that I am writing this message to announce the publication of the first issue of the **Journal of Electronic Systems and Programming**. Launching this new journal would not have been possible without the great and much appreciated contributions from the Editorial Board members and from Electronic Systems and Programming Center (EPC).

Any journal cannot draw its profile and direction since the first or second issue. With this journal, we want to provide a common platform for researchers to share their novel results and latest developments in all areas of electronic systems and programming.

In this journal we encourage publication of papers in English, which makes the audience of our research results much broader. We apply a double-blind peer-review by at least two anonymous reviewers.

Topics suitable for JESP include, but are not limited to, wireless communication protocols, communication media, communication protocols, mobile communication, data security, network topology, sensor network, wireless network, language design and implementation, software and application development, parallel computing, educational applications and Computing-Related Fields.

Finally, we hope you enjoy the first issue of JESP and invite you to submit your best papers for publication.

Dr. Khari A. Armih
Editor-in-Chife

Table of Contents

1

A Cost Model for Heterogeneous Skeletons for CPU/GPU Systems

A Cost Model for Heterogeneous Skeletons for CPU/GPU Systems

Khari A. Armih
College of Computer Technology, Zawia, Libya
khari.armih@gmail.com

Mustafa K. Aswad
Faculty of Engineering, Sabratha University, Libya
mustafaasawd@gmail.com

Abstract

Algorithmic skeletons are widely used to manage multi-processor computations but are most effective when deployed for regular problems on homogeneous systems, where tasks may be divided evenly without regard for processor characteristics. With the growth in heterogeneity, where a multicore is coupled with GPUs, skeletons become layered and simple task distribution becomes sub-optimal. We explore heterogeneous skeletons which use a simple cost model based on a small number of key architecture characteristics to find good task distributions on heterogeneous multicore architectures. We present a new extension to an existing skeleton library associated cost model that enable GPUs to be exploited as general purpose multi-processor devices in heterogeneous multicore/GPU systems. The extended cost model is used to automatically find a good distribution for both a single heterogeneous multicore/GPU node, and clusters of heterogeneous multicore/GPU nodes.

General Terms Algorithms, Design, Performance

Keywords Parallel, Skeleton, Heterogeneous, Cost model, multicore, GPU

1. Introduction

Graphical Processing Units (GPUs) were designed as specialized processors to accelerate graphics processing. Recently, however, the architectures that are comprised of multicores of GPUs have become ubiquitous and cost effective platform for both graphics and general-purpose parallel computing as they offer extensive resources such as high memory bandwidth and massive parallelism [4]. Compared to a conventional core, the performance of a GPU comes from creating a large number of lightweight GPU threads with negligible overheads, where in general purpose multicore the limited number of cores limits the number of data elements that can be processed simultaneously. Today's GPUs enable non-graphics programmers to exploit the parallel computing capabilities of a GPU using data parallelism. With the programmability available on the GPU, a new technique called General Purpose computation on GPU (GPGPU) has been developed [7]. Many parallel applications have achieved significant speedups with GPGPU implementations on a GPU over the CPU [3].

We have been exploring heterogeneous skeletons which use a simple cost model, based on a small number of key architecture characteristics, to find good task distributions on heterogeneous multicore architectures. We have constructed the *HWSkel* library [2] for heterogeneous architectures composed of distributed memory clusters of shared memory multi-core processors. The library, coupled with a simple cost model, offers good speed up for regular data parallel programs.

In this paper, we present the *GPU-HWSkel* extension to our skeletons, which enable GPUs to be exploited as general purpose multi-processor devices in heterogeneous multicore/GPU architectures. We also present an extension to our *HWSkel* cost model which may be used to automatically find a good distribution for both a single heterogeneous multicore/GPU node, and cluster of heterogeneous multicore/GPU nodes.

2. *GPU-HWSkel*: A CUDA-Based Skeleton Library

GPU-HWSkel is an extension of the *HWSkel* library [1, 2] which was designed for heterogeneous multicore cluster architectures. The *HWSkel* library provides data parallel heterogeneous skeletons such as *hMapReduce* and *hMapReduceAll*. They are novel in supporting execution on heterogeneous architectures, and facilitate performance portability, using an architectural cost model to automatically balance load across heterogeneous components of the architecture. *HWSkel* cost model characterise components of the architecture by the number of cores `C`, clock speed `S`, and crucially the size of the L2 cache `L2`, where the relative strength `Strength`[1] of node i of heterogeneous multicore cluster is given by:

$$\mathtt{Strength}_i = \mathtt{C}_i * \mathtt{S}_i * \mathtt{L2}_i \qquad (1)$$

It is important to note that:

- our model excludes network communication costs;
- L2 is used as a shorthand for the top level shared cache which may be L3 on some processors

The new library is designed with the aim of providing a high-level parallel programming environment to program parallel heterogeneous systems including single- and multicore CPU, and GPU architectures. The *GPU-HWSkel* library is based on the CUDA programming model to make GPGPU accessible on NVIDIA GPUs. This makes our approach limited to NVIDIA architectures. The new library implements the same set of data-parallel skeletons that are provided by the base *HWSkel* library [2], *hMap* and *hReduce* parallel skeletons, *hMapReduce* skeleton, and *hMapReduceAll* skeleton. These skeletons provide a general interface for both GPUs and CPUs since the library is based on OpenMP and MPI to support CPU implementations, as well as CUDA for GPU implementations.

[1] In [2] we termed Strength “Power” but we now feel that this has inappropriate absolute connotations

3. A GPU Workload Distribution Cost Model

We next introduce an extension of our approach to account for the GPU as an independent processing element and to automatically find a good distribution in heterogeneous multicore/GPU systems. A new model is designed to provide a generic load-balancing strategy, so our skeletons will fully automate the distribution process on an integrated multicore/GPU architecture, which in turn makes the task of workload distribution much easier for the skeleton programmer.

3.1. Related Work

Several performance cost models have been developed to utilise the high performance of heterogeneous parallel systems. However, to the best of our knowledge little research has been done in considering the use of multiple cores and a GPU card simultaneously in heterogeneous architectures. This section briefly describes related models that consider using the heterogeneous multicore/GPU systems.

In [6] a performance cost model has been introduced in conjunction with a 2D-FFT library for finding the optimal distribution ratios between CPUs and GPUs. The model predicts the total execution time of a 2D-FFT of arbitrary data size. Firstly, the FFT computation is split into small steps, and then the model predicts the execution time for each execution step using profiling results on a heterogeneous multicore/GPU system, and finally the model determines the optimal load distribution ratio as the shortest predicted execution time.

Moreover, the model attempts to overcome the limitation in the memory sizes of GPUs by iterating GPU library calls.

An adaptive mapping technique is implemented in the heterogeneous programming system called Qilin [5] for computation placement on heterogeneous multiprocessors. It is a fully automatic approach to find the optimal computation mapping to processing elements of a heterogeneous system. Qilin has a capability to use any heterogeneous platform, since it does not require any hardware information for its implementations. This technique uses execution time projections stored in a database to determine the execution times of both CPU and GPU for a given program, problem size and hardware configuration. Further, the determined execution times are used to statically partition the workload among the CPU and GPU. Thus, the first step in the Qilin programming system is to conduct a training run to add data to a database.

An optimisation framework is introduced in [8] to improve the load balance on heterogeneous multicore/GPU systems. Instead of using static partitioning, the model applies a new adaptive technique that dynamically balances the workload distribution between the CPU cores and the GPU in a single node. At the beginning of execution, the model measures the performance of both the CPU and the GPU, and then the measurement is used to guide the workload distribution in the next step. In addition, the model tries to hide the communication overhead of transferring the data between CPU and GPU by providing software pipelining to overlap data transfers and kernel execution.

3.2. Discussion

Load-balancing at the multi-node heterogeneous hardware level can either be done dynamically or statically before program execution is started. Static cost models incur less overhead than dynamic models due to their simplicity and lack of runtime overhead. Besides, heterogeneous integrated multicore/GPU systems are nonetheless highly distributed. Hence, we wish to develop an accurate cost model and prediction mechanism to balance the workload distribution across the CPU and the GPU in each node as well as between the nodes in a cluster. The new cost model inherits all the features of the *HWSkel* cost model presented in [2]. However, in contrast to the performance cost models described previously, our cost model provides the following new features:

Heterogeneous-mode. The performance cost models in [5, 6, 8] measure the performance of both the CPU and the GPU in a heterogeneous system by using profiling. Our performance cost model is based on two different type of performance measurements for the CPU and GPU. Since the performance of the GPU is changed by changing the data size while the performance of the CPU can be more stable for different data sizes, we measure the performance of the GPU with a training run, while the CPU performance is calculated using the hardware parameters.

Hardware-auto-selection. Since our performance cost model can provide enough information about the CPU and GPU performance capability, our heterogeneous skeletons can choose to use either the multicore CPU or a GPU card to execute the ongoing program. This feature will be discussed in more detail in the future

3.3. Methodology

The new model is viewed as two-phase since the underlying target hardware consists of two levels of heterogeneous hardware architectures. The model is divided into two main components:

- *Single-Node*, to guide the workload distribution across the multiple cores and the GPU device inside each node in the integrated multicore/GPU system;
- *Multi-Node*, to balance the workload across the nodes in the cluster.

In general, we focus on predicting the runtime of the application code on the GPU device and use the *HWSkel* cost model [2] for measuring the processing strength of the CPU. In addition, since the workload is statically distributed across the multiple cores and the GPU and also between the nodes at the beginning of program execution, the model does not allow for any communication between the CPU cores and the GPU or between the nodes in the system other than via the skeleton.

3.3.1. Single-Node Cost Model

We base the workload distribution on the performance ratio between the core and GPU in the integrated multicore/GPU computing node. So the cost model aims to predict the execution time of a single core vs. the GPU device for arbitrary data sizes, and calculates the chunk size for a CPU core and the GPU by using this performance ratio. To facilitate our discussion, let us introduce the following notation:

TC: Program runtime on a single core.
TG: Program runtime on the GPU.
Strength: The relative strength of computational unit.
C : Number of cores in a single node.
D : Data Size.

We start by calculating Strength, the relative strengths of the GPU and a single core:

$$\mathrm{Strength} = \mathrm{TC} \,/\, \mathrm{TG}$$

If the GPU is allocated DGPU units of data then the multicore will

$$D_{GPU} * \mathrm{Strength} \,/ (C - 1)$$

units. As the node comprises a multicore and a single GPU, the total data size is

$$D_{total} = D_{GPU} + D_{GPU} * \mathrm{Strength} \,/ (C - 1)$$

Factoring out DGPU, the data allocated to the multicore is

$$D_{multicore} = D_{total} \,/\, (1 + \mathrm{Strength} \,/\, (C - 1))$$

and the each core is allocated

$$D_{core} = D_{multicore} \,/ (C - 1)$$

3.3.2. Multi-Node Cost Model

The *Multi-Node* cost model is based on the *Single-Node* cost model to determine the chunk size for each node in the system. As a heterogeneous cluster might have different kinds of computing nodes, the key idea of the *Multi-Node* cost model is to measure the relative strength for each node in the cluster. Hence, the total available strength `Strength` for `n` nodes is given by:

$$Strength_{total} \sum_{i=1}^{i=n} Strength_i$$

So for data size D_{total}, the chunk size for node `i` is:

$$(Strength_i / Strength_{total}) \ * \ D_{total}$$

Nodes may have different architectures, and hence strengths. The relative strength of a node `i` that consists of a GPU and multiple cores is the sum of the relative strengths of the cores, $Strength_{core}$, and the GPU $Strength_{GPU}$:

$$Strength_i = (Ci-1) * Strength_{corei} + Strength_{GPUi} \qquad (2)$$

if there is only a single core, i.e. `C = 1`, it follows directly that

$$Strength_i = Strength_{GPUi} \qquad (3)$$

To calculate $Strength_{core}$ and $Strength_{GPU}$, we first measure T_{Cbase}, the runtime of the program on core of the system, and use it follows.

$$Strength_{GPUi} = TC_{base} / TG_i \qquad (4)$$

In practice we predict the relative strengths on the base core, $\mathtt{Strength}_{\mathtt{Cbase}}$, and on the cores of node $\mathtt{i}$, $\mathtt{Strength}_{\mathtt{Ci}}$ using the *HWSkel* cost model, i.e. Equation 1 in Section 2:

$$\mathtt{Strength}_{\mathtt{Ci}} = \mathtt{S}_{\mathtt{i}} * \mathtt{L2}_{\mathtt{i}} \qquad (5)$$

$$\mathtt{Strength}_{\mathtt{Cbase}} = \mathtt{S}_{\mathtt{base}} \_ \mathtt{L2}_{\mathtt{base}} \qquad (6)$$

Hence the relative strength of a core on node $\mathtt{i}$ is:

$$\mathtt{Strength}_{\mathtt{corei}} = \mathtt{Strength}_{\mathtt{Ci}}/\mathtt{Strength}_{\mathtt{Cbase}} \qquad (7)$$

Substituting equations (4) and (7) in (2) gives the cost equation used in the *GPU-HWSkel* library:

$$\mathtt{Strength}_{\mathtt{i}} = (\mathtt{C}_{\mathtt{i}}\mathtt{-1})*\mathtt{Strength}_{\mathtt{Ci}}/\mathtt{Strength}_{\mathtt{Cbase}} + \mathtt{Strength}_{\mathtt{GPUi}} \qquad (8)$$

The key point is that we need only measure $\mathtt{TG}_{\mathtt{i}}$ and $\mathtt{TC}_{\mathtt{base}}$ to parametrise the model.

4. *GPU-HWSkel* Evaluation

4.1. Benchmarks

The performance of each *GPU-HWSkel* skeleton is evaluated using two applications: the first is the widely used matrix multiplication, and the second is an iterative Fibonacci program.

Matrix Multiplication. A well-known representative for a wide range of high-performance applications is the problem of multiplying two matrices. There are a number of different techniques to multiply matrices. Here, the number of multiplications performed is reduced by breaking down the input matrices into several submatrices.

Fibonacci Program. Fibonacci is a function that computes Fibonacci numbers. In our experiment, we use a simple program that calculate the Fibonacci value for an array of integer numbers with fixed constant by replicating the fib function in the original sequential program. In the parallel version, the array of the integers is split into chunks using a split function which employs the cost model for load distribution, and then the fib function is mapped in parallel across each chunk.

4.2. Platform

We conduct our experiments on a heterogeneous cluster with a number of different integrated multicore/GPU nodes located at Heriot-Watt University as described in Table 1. Each of the machines is connected to an NVIDIA GeForce GT 520 GPU device. The device has 1 GB of DRAM , one multiprocessor (MIMD unit) clocked at 810 MHz, and 48 processor cores (SIMD units) running at 1620 MHz with 16 KB of shared memory. CUDA version 4.0 was used for the experiments. The CUDA code was compiled using the NVIDIA CUDA Compiler (NVCC) to generate the device code that is launched from the host CPU.

Table 1: Experimental Architectures.

Name	CPU				GPU			
	archi	Cores	MHz	L2	archi	SM	Cores	MHz
lxpara	Xeon 5410	8	1998	6144KB	GT520	1	48	1620
lxphd	IntelE8400	2	1998	6144KB	GT520	1	48	1620
linuxlab	2 DuoCPU	2	1200	2048KB	G 520	1	48	1620
brahma	Xeon(TM)	4	3065	512KB	GT520	1	48	1620

4.3. Performance Evaluation

For all the measurements that are performed on the two-core (such as *linux* and *lxphd*) machines, we follow the common practice of increasing the input-data size to evaluate the behaviour consistency of the *hMap* skeleton with our performance cost model. While in the case of using machines with an eight-core processor (such as *lxpara*), all programs are measured with a fixed data size on 1,2,3,4,5,6, and 7 cores together with a single GPU device. We measure the runtimes for the *hMap* skeleton implementation, with a fixed data size of 1500 x 1500 for the input matrices, and 80,000 elements of Fibonacci (1,000,000).

4.3.1. Single multicore/GPU Node Results

The single-node experiments have been carried out our on *linux* lab, *lxphd*, and *lxpara* as single nodes.

Table 2 and 3 show the *hMap* runtime for matrix multiplication and Fibonacci on *linux* lab and *lxphd* respectively. The measurements report the runtime on 1 core, GPU, GPU plus 1 core, and show the percentage improvement of hMap using the CM2 cost model. The *hMap* Fibonacci has an improvement of 95% over the sequential time and improvement of 4% over the GPU time on *linux* lab and *lxphd* using CM2 , while the *hMap* matrix multiplication has an improvement of 68% over the sequential time on both *linux* lab and lxphd, and improvement of 32% on linux lab and 20% over the GPU time on *lxphd*.

Table 2: 1 Core *hMap* Runtimes (*linux lab*).

Data size	Run-Time (s)			1 Core+GPU Improvement%	
	1 Core	GPU	1 Core+GPU	1 Core	GPU
800x800	2.31	1.40	1.32	42%	5%
900x900	3.30	1.77	1.54	53%	12%
1000x1000	4.52	2.09	1.80	60%	13%
1100x1100	6.02	2.73	2.12	64%	22%
1200x1200	7.82	3.26	2.54	68%	22%
1300x1300	9.94	4.29	3.19	67%	25%
1400x1400	12.41	5.37	4.00	67%	25%
1500x1500	15.26	7.23	4.91	67%	32%

(a) matrix multiplication

Data size	Run-Time (s)			1 Core+GPU Improvement%	
	1 Core	GPU	1 Core+GPU	1 Core	GPU
1000	3.36	0.19	0.17	94%	10%
2000	6.77	0.34	0.32	95%	5%
5000	17.02	0.79	0.75	95%	5%
10000	34.17	1.53	1.47	95%	3%
20000	67.93	3.06	2.91	95%	4%
30000	103.30	4.55	4.39	95%	3%
40000	137.08	6.071	5.79	95%	4%
50000	170.80	7.55	7.24	95%	4%
60000	207.33	9.05	8.69	95%	4%
70000	243.79	10.51	10.12	95%	3%
80000	278.04	12.05	11.52	95%	4%

(b) Fibonacci

Table 3: 1 Core hMap Runtimes (lxphd).

Data size	Run-Time (s)			1 Core+GPU Improvement%	
	1 Core	GPU	1 Core+GPU	1 Core	GPU
800x800	4.28	1.47	1.41	67%	4%
900x900	6.09	1.84	1.66	72%	9%
1000x1000	8.37	2.25	1.98	76%	12%
1100x1100	11.12	2.88	2.36	78%	18%
1200x1200	14.43	3.49	3.07	78%	12%
1300x1300	18.34	4.46	3.90	78%	12%
1400x1400	22.91	5.79	4.88	78%	12%
1500x1500	28.25	7.61	6.02	78%	20%

(a) matrix multiplication

Data size	Run-Time (s)			1 Core+GPU Improvement%	
	1 Core	GPU	1 Core+GPU	1 Core	GPU
1000	3.27	0.20	0.19	94%	5%
2000	6.53	0.36	0.34	94%	5%
5000	16.36	0.79	0.77	95%	2%
10000	32.75	1.55	1.48	95%	4%
20000	65.47	3.07	2.93	95%	4%
30000	98.13	4.55	4.39	95%	3%
40000	130.97	6.07	5.77	95%	4%
50000	163.57	7.53	7.22	95%	4%
60000	196.44	9.06	8.67	95%	4%
70000	229.18	10.55	10.06	95%	4%
80000	261.77	12.00	11.52	95%	4%

(b) Fibonacci

Table 4 shows the runtime of matrix multiplication with data size of 1500 x 1500 and Fibonacci with data size 80,000 elements with a value of 1,000,000 using *hMap* on *lxpara*. The measurements show that the *hMap* Fibonacci has improvement of 77% over 8 cores, while the *hMap* matrix multiplication shows that there is no improvement after 6 cores. The parallel performance is measured as the absolute speedup of using both the GPU device and the multiple cores within a single machine. Here the experiments have been carried out on *linux*, *lxphd*, and *lxpara* machines as single nodes.

Table 4: Multiple Core hMap Runtimes (lxpara).

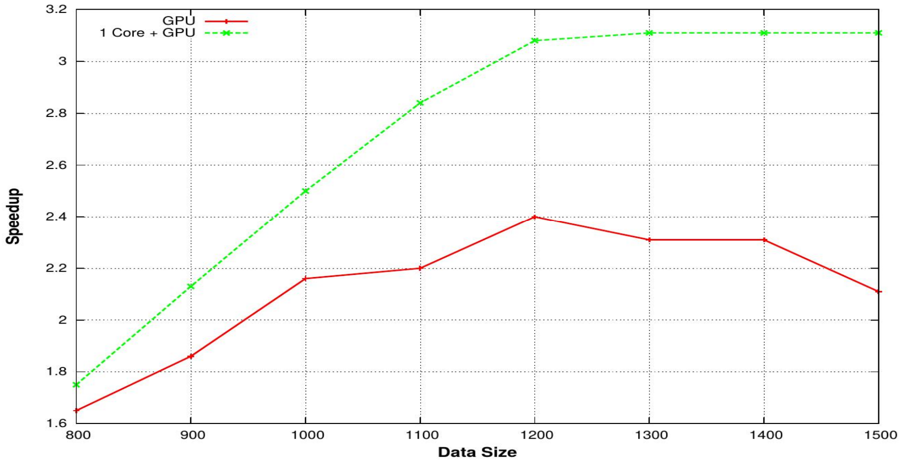
Data size	**Run-Time (s)**			**1 Core+GPU Improvement%**	
	Cores	**GPU**	**(Core-1)+GPU**	**Cores**	**GPU**
1	19.60	7.26	7.26	62%	0%
2	9.82	7.26	5.31	45%	26%
3	6.55	7.26	4.20	35%	42%
4	4.93	7.26	3.48	29%	52%
5	3.94	7.26	3.09	21%	57%
6	3.29	7.26	2.92	11%	59%
7	2.89	7.26	2.84	1%	60%
8	2.54	7.26	2.78	-9%	61%

(a) matrix multiplication

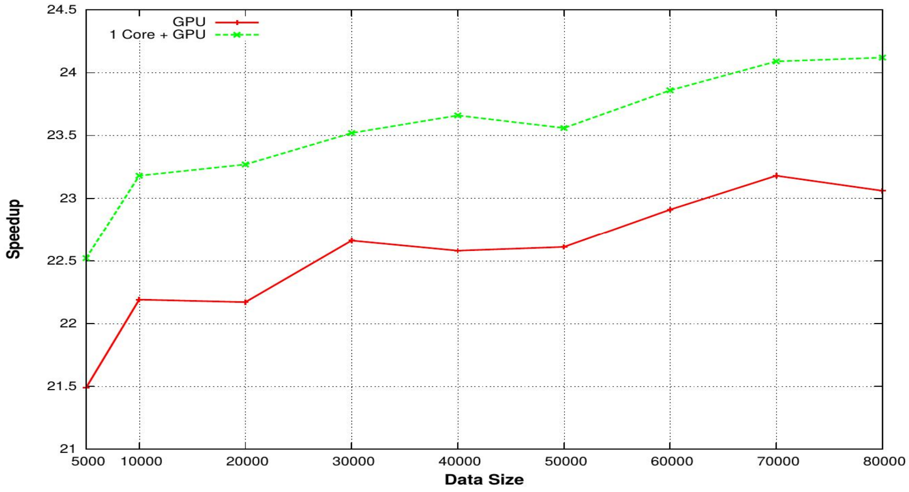
Data size	**Run-Time (s)**			**1 Core+GPU Improvement%**	
	Cores	**GPU**	**(Core-1)+GPU**	**Cores**	**GPU**
1	344.37	12.03	12.03	96%	0%
2	172.01	12.03	11.60	93%	3%
3	114.85	12.03	11.28	90%	6%
4	86.06	12.03	10.90	87%	9%
5	68.99	12.03	10.59	84%	11%
6	57.43	12.03	10.27	82%	14%
7	49.26	12.03	9.96	79%	17%
8	43.09	12.03	9.69	77%	19%

(b) Fibonacci

Figures 1 and 2 show the absolute speedup achieved for the Fibonacci and matrix multiplication programs with different input data sizes on the two-core *linux* and *lxphd* machines respectively. The graphs in Figures 1 and 2 compare the absolute speedup curve for one CPU-core plus single GPU implementation with the curve for GPU implementation.



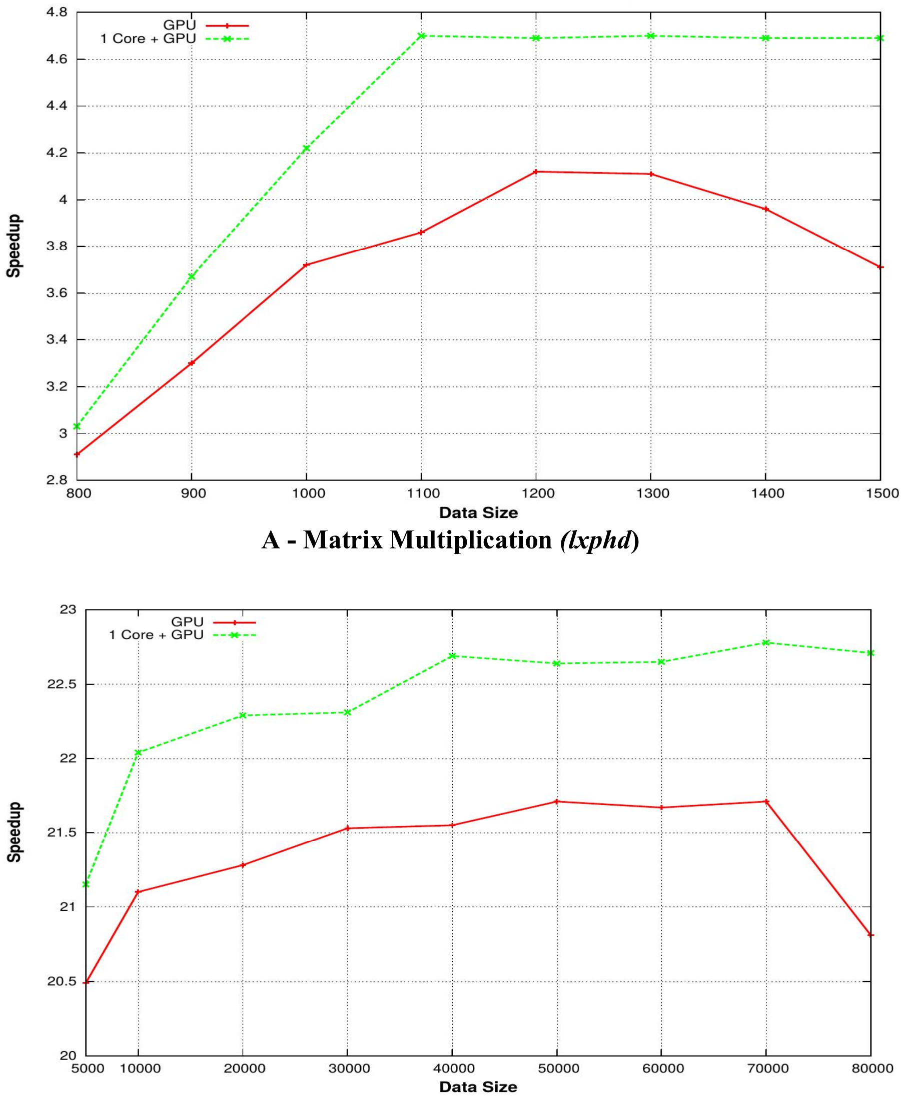
A - Matrix Multiplication *(linux-lab*)



B - Fibonacci (*linux-lab*)

Figure 1: *hMap* Absolute Speedup on (*linux lab*)

Although the computing capability of GPU is relatively large compared with the computational strength of a single CPU-core, results show that using CPU-cores together with a GPU can deliver an expected and acceptable speedups on both machines.



A - Matrix Multiplication *(lxphd*)

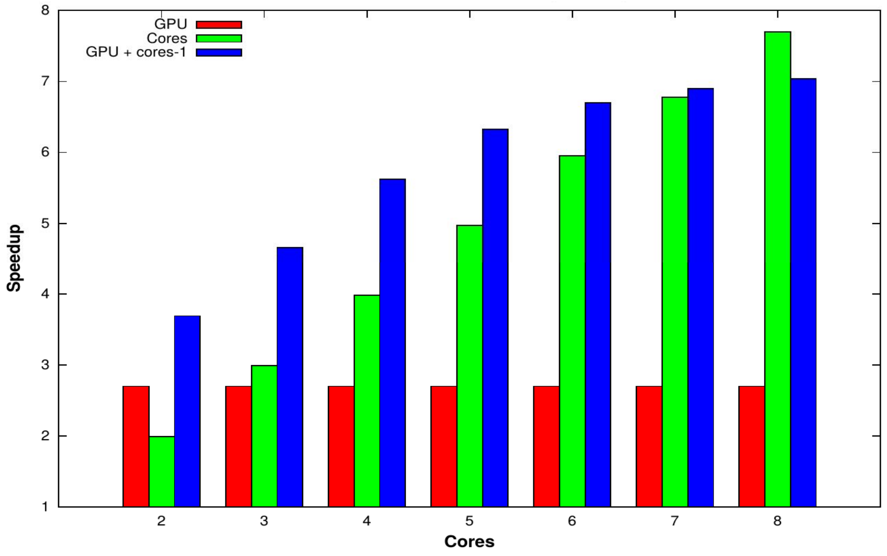
B - Fibonacci (*lxphd*)

Figure 2: *hMap* Absolute Speedup on (*lxphd*)

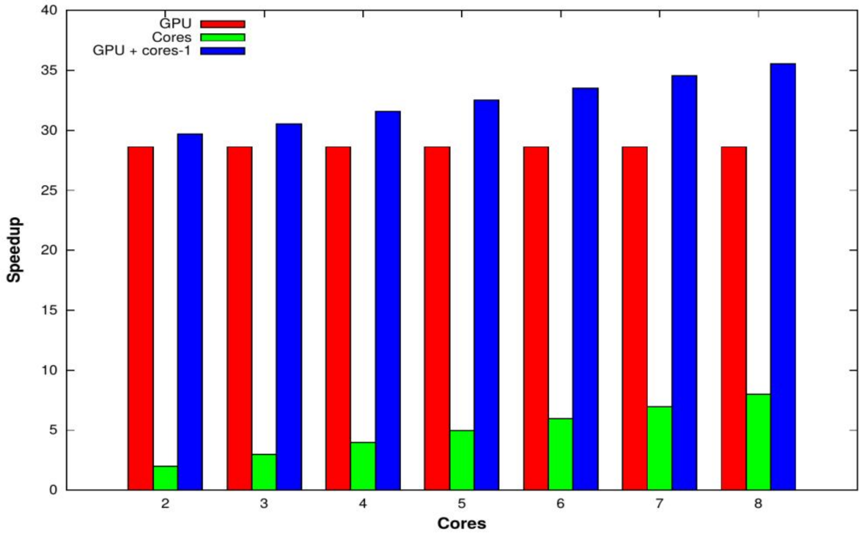
Our results also suggest that using the performance cost model for determining granularity and data placement on different heterogeneous architectures can provide a good load balance for data distribution between CPU-cores and a GPU. This is reflected in the speedup graphs where the curves are broadly similar for both programs with different input data size on different parallel heterogeneous architectures. Next, to investigate the impact of the placement strategy on parallel performance of a varying number of CPU-cores with a single GPU, the experiments have been run with the Fibonacci and matrix multiplication programs on a machine with eight CPU-cores (*lxpara*). Figure 3 compares the absolute speedups of both Fibonacci and matrix multiplication programs on only CPU-cores and GPU, and CPU-core+GPU of the *lxpara* machine. This shows that in both programs a good performance has been obtained as anticipated.

Firstly, the results presented in Figure 3 are consistent with others that obtained for both programs on the *linux* and *lxphd* machines, where the speedup is increased by using one CPU-core plus the GPU.

Secondly, we have obtained almost linear speedup with parallel efficiency of about 99% in both programs on CPU-cores. However, in the matrix multiplication program the speedup has a slight degradation to 95% parallel efficiency after six cores due to decreasing the chunk size. The results show that our skeleton delivers 28x from the GPU compared to a single CPU-core in the Fibonacci program, while we report nearly 2.8x speedup over a CPU-core by using a GPU in the matrix multiplication application. The variation in speedup between programs is due to the *GPU-HWSkel*-based parallel algorithm used for each program. Since the major problem with GPU implementations which affects the performance efficiency is the size of data being transferred between CPU and GPU, the algorithm requires too much data communication, which in turn increases the CPU/GPU communication overhead. Therefore, it is obvious that the algorithm for matrix multiplication is more suitable for multicore processors than a GPU implementation, while the Fibonacci program makes a good GPU program.



A- Matrix Multiplication *(lxpara)*



B - Fibonacci (*lxpara*)

Figure 3: *hMap* Absolute Speedup on (*lxpara*)

4.3.2. Clusters of multicore/GPU Nodes Results

We evaluate the performance of an early version our cost model and its effect on our *hMap* heterogeneous skeletons on different combinations of the architectures outlined in Section 4.2. Figure 4 plots the speedups for different configurations with different processing elements calculating Fibonacci(1000000) 1500,000 times. The graph compares the speedups of three different kinds of computing units (i.e. CPU-cores, GPU-device, and GPU-device plus CPU-cores) on different numbers of given machines. Figure 4 shows that the results are consistent with those that were presented in Section 4.3.1. However, the performance of our *hMap* skeleton has been improved by exploiting the CPU-cores along with GPU in each host node. We suggest once again that our performance cost model has provided a good strategy of data placement for heterogeneous architectures. The graph shows that the implementation of our *hMap* skeleton can deliver good scalability, where the upper speedup curve shows improved performance results for using our cost model for data placement between the heterogeneous nodes as well as within each node between multiple cores and GPU.

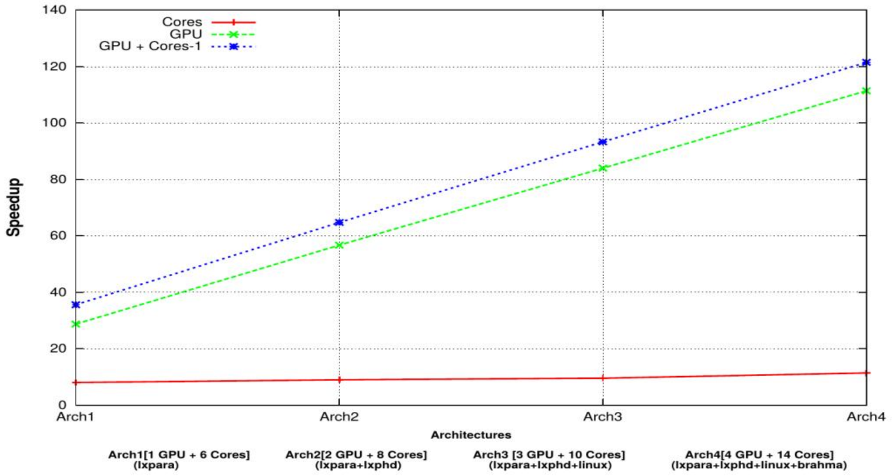


Figure 4: Speedups for the *hMap* on a Heterogeneous Cluster

5. Conclusions and Future Work

In this paper, a new performance cost model has been presented for heterogeneous integrated multicore/GPU systems. The purpose of the new cost model is to balance the workload distribution between the nodes on heterogeneous cluster as well as between multiple cores and GPU device inside each node in cluster. Our cost model is viewed as two-phase, the Single-Node phase guides workload distribution across a CPU core and a GPU using the performance ratio between the CPU and GPU in the integrated multicore/GPU computing node, and the Multi-Node phase balances the distribution of workload among the nodes on heterogeneous integrated multicore/GPU cluster. In general, we focus on predicting the runtime of the application code on the GPU and use an architectural performance cost model for measuring the processing strength of CPU to calculate the performance ratio. In summary, our experimental results show that using multiple cores together with a GPU in the same host with our skeleton and cost model can deliver good performance either on a single node or on multiple node architecture. Our work has a number of limitations, which we propose to address in future work:

- As noted above, our cost models do not take account of communication costs. We will explore how our simple notion of strength can be extended to account for communication characteristics.
- Our library, being based on CUDA, is NVIDIA specific. We will modify our library to use the OpenCL standard.

References

1. K. Armih. Toward Optimised Skeletons for Heterogeneous Parallel Architecture with Performance Cost Model. PhD thesis, Heriot Watt University, UK, 2013.
2. K. Armih, G. Michaelson, and P. Trinder. Cache size in a cost model for heterogeneous skeletons. In Proceedings of the fifth international workshop on High-level parallel programming

and applications, HLPP ’11, pages 3–10, New York, USA, 2011.

3. S. Gupta. Performance Analysis of GPU compared to Single-core and Multi-core CPU for Natural Language Applications. IJACSA-International Journal of Advanced Computer Science and applications, pages 50–53, 2011.
4. D. B. Kirk and W.-m. W. Hwu. Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2010.
5. C.-K. Luk, S. Hong, and H. Kim. Qilin: exploiting parallelism on heterogeneous multiprocessors with adaptive mapping. In Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture, pages 45–55, New York, USA, 2009.
6. Y. Ogata, T. Endo, N. Maruyama, and S. Matsuoka. An efficient, model-based CPU-GPU heterogeneous FFT library. In Parallel and Distributed Processing. IEEE International Symposium on, 2008.
7. E. Wu and Y. Liu. Emerging technology about GPGPU. In IEEE Asia-Pacific Conference on Circuits and Systems, 2008.
8. C. Yang, F.Wang, Y. Du, J. Chen, J. Liu, H. Yi, and K. Lu. Adaptive optimization for petascale heterogeneous cpu/gpu computing. In Proceedings of the 2010 IEEE International Conference on Cluster Computing, CLUSTER ’10, pages 19–28, Washington, USA, 2010.

2

Prediction of short-term load using artificial neural networks

Prediction of short-term load using artificial neural networks

A. Shebani
College of Computer Technology – Zawya
amerelshibani@yahoo.com

Abstract

Neural network can be used for solving the particular problems which are difficult to solve by the human beings or the conventional computational algorithms. The computational meaning of the training comes down to the adjustments of certain weights which are the key elements of the artificial neural netwrok. This is one of the key differences of the neural network approach to problem solving than conventional computational algorithms. This adjustment of the weights takes place when the neural network is presented with the input data records and the corresponding target values. Due to the possibility of training neural networks with off-line data, they are found useful for power system applications. This paper focus on short-term load prediction using three types of neural networks. An accurate short-term forecasting method for load of electric power system can help the electric power system operator to reduce the risk of unreliability of electricity supply. On this paper, radial basis function neural netwrok (RBFNN), Nonlinear Autoregressive model with eXogenous input neural network (NARXNN), and backpropagation neural network (BPNN) were developed to predict the short-term load. Simulation results show that the three types of neural networks can predict the load efficiently. The neural network simulation results were implemented using the Matlab program. The accuracy of load prediction using the neural networks was investigated and assessed in terms of mean absolute percentage error (MAPE).

Keywords: Short-term load prediction, neural networks, RBFNN, NARXNN, BPNN, MAPE, Matlab.

1. Introduction to short-term load prediction using neural networks

Neural networks have been used in a board range of applications including: pattern classification, pattern recognition, optimization, prediction and automatic control. In spite of different structures and training paradigms, all NN applications are special cases of vector mapping. The application of NNs in different power system operation and control strategies has led to acceptable results. During 1990-1996 with them during 2000-2005, the following fields has attracted the most attention in the past five years: load forecasting, fault diagnosis/fault location, economic dispatch, security assessment, and transient stability. The main advantages of using NNs on power system are: its capability of dealing with stochastic variations of the scheduled operating point with increasing data, very fast and on-line processing and classification, and implicit nonlinear modeling and filtering of system data [1]. Commonly and popular problem that has an important role in economic, financial, development, expansion and planning is load forecasting of power systems.

Generally, the load prediction can be categorized into three groups: short-term load prediction over an interval ranging from an hour to a week is important for various applications such as unit commitment, economic dispatch, energy transfer scheduling and real time control.

A lot of studies have been done for using of short-term load prediction with different methods [2], [3], [4], and [5].
Mid-term load prediction that range from one month to five years, used to purchase enough fuel for power plants after electricity tariffs are calculated [6]. Long-term load prediction, covering from 5 to 20 years or more, used by planning engineers and economists to determine the type and the size of generating plants that minimize both fixed and variable costs [7]. Fig. 1 shows the percentage of number of published works during five years in different load prediction types.

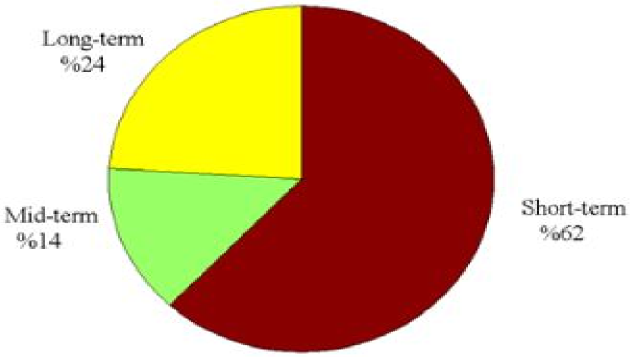


Fig 1 Types of load prediction that done with NN

The load of power grid was predicted using neural networks, and results show that the neural netwrok can predict the load efficiently. The short-term load prediction using neural netwrok carried out on a previous works as: the short-term load predicted using NARXNN [8], and [9]; the short-term load predicted using RBFNN [10]; and the short-term load predicted using BPNN [11], [12]. This paper focus on comparison between the short-term load prediction using the three types of the neural networks (NARXNN, RBFNN, and BPNN) to show the neural networks is a powerful toll to predict the load in power systems, and to show which type of neural netwrok is an accurate to predict the short-term load.

2. Artificial Neural Networks (ANNs)

2.1 Introduction to artificial neural networks

The concept of a neural network was originally conceived as an attempt to model the biophysiology of the brain. At the same time, research engineers were concerned with how to use Artificial neural networks (ANNs) to form controllers from neurons with interesting and powerful computational capabilities. ANNs offer a potential solution for problems which require complex data analysis and promise to form the future basis of an improved alternative to current engineering practice. Many researchers found that neural networks have many applications in various fields of study including modeling and control of linear and nonlinear systems. Neural networks have been developed in different ways, where various algorithms and methods have been applied, such as backpropagation (BP) rule and Radial Basis Function (RBF) [8], and [13].

In most cases the structure of the MLPs is carried out in a fairly heuristic way, so far a certain problem a reasonable number of layers and neurons in each layer are initially selected, based on experience. However, if incorrect number of nodes are selected, then adjustments can be made on a trial and error basis. Moreover, the backpropagation algorithm (BP) suffers from several deficiencies, such as slow convergence and construction complexity. An alternative approach to overcome the limitations associated with the BP algorithm is to use the Radial Basis Function (RBF) network which is discussed in details in the following sections.

2.2 Radial Basis Function Network

The RBF network can be regarded as a special three-layer network including input, hidden and output layers. Full explanations of the connections of these layers together with the activation function are given in the next sections. The performance of the RBF depends on the proper selection of three important parameters, centers, widths and the weights. The radial basis function has been shown to able to solve many problems in different fields, one example is the modeling and controlling of non-linear systems. The RBF neural network has a feed forward structure consisting of three layers as shown in Fig. 2 [14].

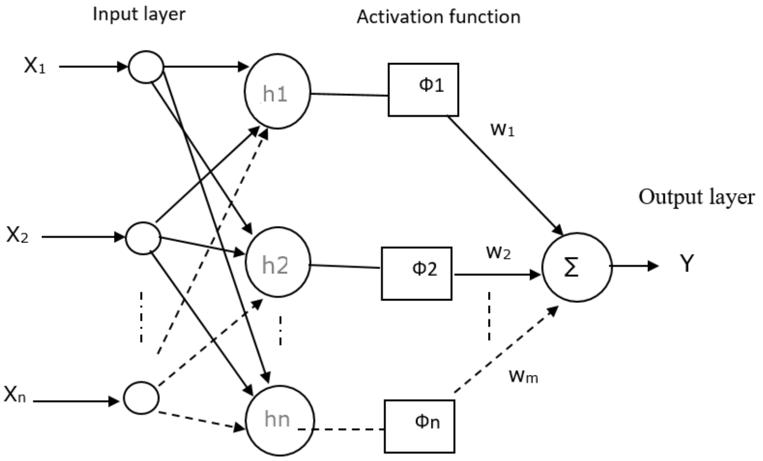


Fig 2 The radial basis function neural network structure

The Radial Basis Function Network consists of three important parameters, centers *(c)*, widths *(σ)* and weights *(w)*. The value of these parameters are generally unknown and may be found during the learning process of the network. There are a variety of methods to allow the RBF network to learn. These processes are generally divided into two stages, as each layers of the RBF perform a different task. The first learning stage involves selecting the centers and the widths in the hidden layer. The second stage is to adjusting the weights in the output layer [5].

2.3 Nonlinear Autoregressive model with eXogenous input neural network (NARXNN)

A Nonlinear Autoregressive model with eXogenous input neural network (NARXNN) was used in this project for wheel wear and rail wear prediction. The NARXNN can be implemented using a feedforward neural network [15]. Fig. 3 shows the structure of the NARXNN which are called NARX recurrent neural networks. This network simply uses a TDL-type network (Tapped delay line) with a feedback connection from the output of the network to the input. The function of the delay line (TDL) or taps is to feed the neural network with the past values of inputs [16].

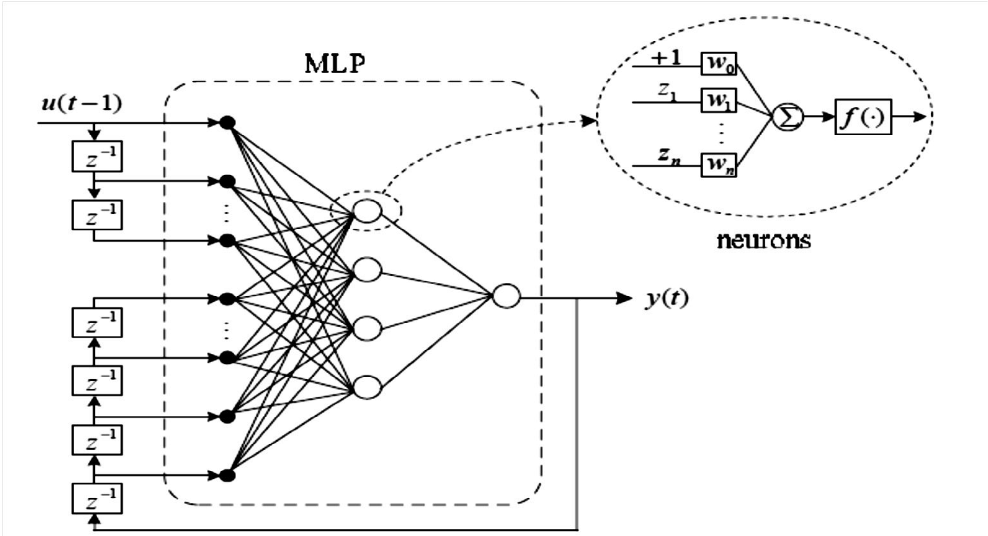


Fig 3 The structure of NARXNN [16], [15]

The output of the NARXNN is represented using the following equation:

$$\mathrm{y(t) = f(u(t-1), u(t-2), \dots, u(t-n), y(t-1), y(t-2), \dots, y(t-m,W} \qquad (1)$$

Where $\mathrm{u(t)}$ is the input and $\mathrm{y(t)}$ is the output of the network at time t, n and m are the input-memory and output-memory order, W is a weights matrix, and f is a nonlinear function.

The output at time t depends on both its past m values and the past n values of the input as well. The Lavenberg-marquardt backpropagation algorithm was used on this work to train the NARXNN. The training of NARXNN automatically stops when the validation error (MSE) begins to increase [17].

2.4 Backpropagation Neural Network (BPNN)

This section describes one of the most common types of artificial neural network. Multilayer feedforward (MLFF) neural network with backpropagation (BP) learning (multilayer perceptron). A general multilayer feedforward (MLFF) network is illustrated in Fig. 4. The MLFF consists of three layers: Input layer, hidden layers, and output layer. The hidden layer is sometimes called the internal layer because it only receives internal inputs then produces internal outputs. It consists of one or more hidden layers [18].

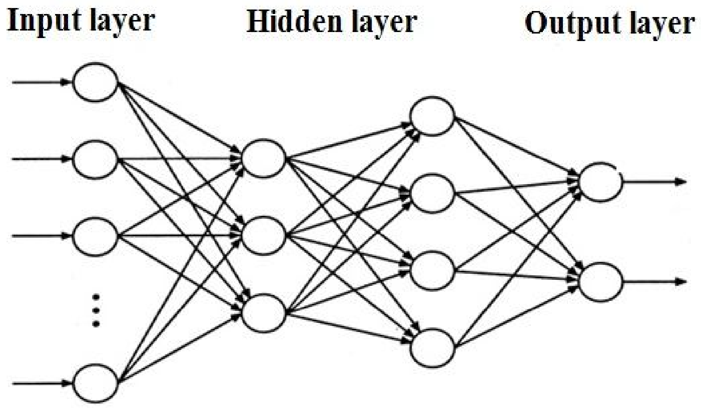


Fig 4 MLFF Backpropagation Neural Network (BPNN) [18]

The backpropagation training process requires an activation function. One of the most common activation functions is the sigmoid function. The most common training algorithm which is used on this work for training of BPNN is a Levenberg-Marquardt algorithm which adjusts the weights to reduce the error [18].

3. Load prediction using RBFNN, NARXNN, and BPNN

This section describes the procedures for training the neural network to learn from the Year 2005 hourly load data published on a previous work were used to train the neural netwrok in order to predict the next day load demand [11]. The Matlab ANN toolbox was utilized in designing the neural networks architecture. The input consists of daily 24-hour load data for 12 months of the year 2005 and daily average maximum temperature altogether making 25 inputs rows by 365 days. The output layer will be a day's 24 hours load forecast for the utility company. The Target data is the same as the input's daily 24 hours load data. The following equation was used to calculate the mean absolute percentage error (MAPE) [19], [20]:

$$\mathrm{MAPE} = \frac{1}{N} \sum_{i=1}^{N} \frac{|A_i - P_i|}{Ai} \text{ X } 100 \tag{2}$$

Where A_i is the actual load, P_i is the predicted load, i is time period, and N is the number of time periods (number of observed values). MATLAB has been recognised as an effective neural network modelling tool and is subsequently used in this paper to implement the three types of neural networks for load prediction. The actual load and load predicted using RBFNN, NARXNN, and BPNN are shown in Fig. 5. Matlab tool box was used on this work to design, train, and test the RBFNN, NARXNN, and BPNN to predict the load.

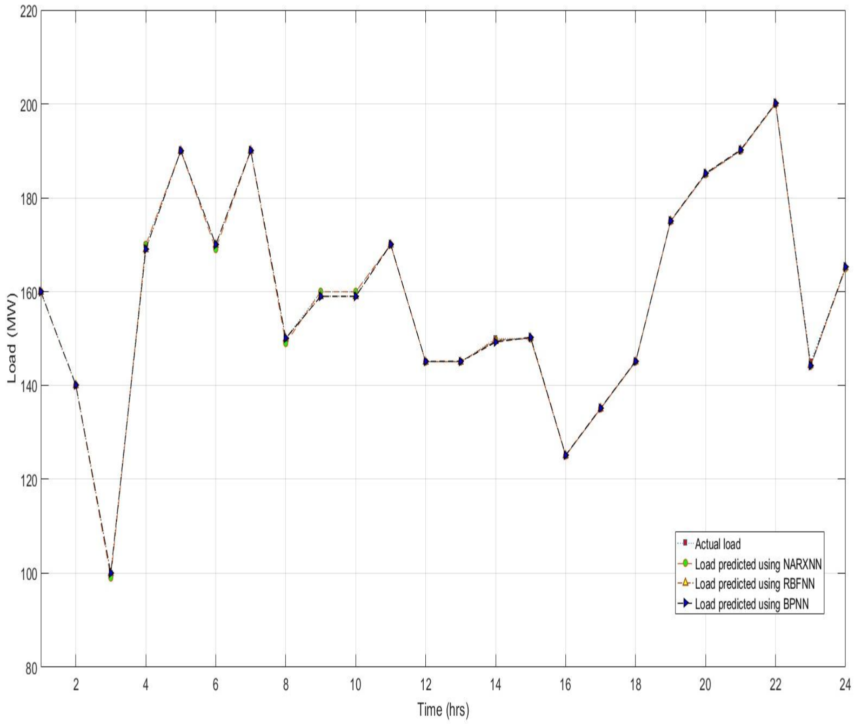


Fig 5 Actual load and load predicted using NARXNN, RBFNN, and BPNN

The simulation results shown in Fig. 5 show that the three types of neural networks achieved good short-term load prediction as: The MAPE was 1.12% for NARXNN, 1.45% for RBFN, and 1.67% for BPNN. Therefore, the accuracy of a three type of neural network for load prediction was greater than 98%. The accuracy of the neural network model was assessed by mean absolute percentage error (MAPE), the accuracy of NARXNN was the best, followed respectively by the RBFNN and BPNN. The NARXNN have an advantage over RBFNN and BPNN, where the output of NARXNN is fed back to the input (closed loop).

4. CONCLUSIONS

This paper has proposed a three types of neural networks to accurately and reliably predict the load of an electric power system. A load prediction using the three neural networks was designed using Matlab program (ANN Toolbox). The implementation of the network architecture, training of the Neural Network and simulation of test results were all successful with a very high degree of accuracy resulting into 24 hourly load output. The simulation results show that the RBFNN, NARXNN, and BPNN can predict the short-term load efficiently with accuracy of 98%. It can therefore be concluded that the RBFNN, NARXNN, and BPNN are accurate models for short-term load prediction. The accuracy of the neural network model was assessed by mean absolute percentage error (MAPE), the accuracy of NARXNN was the best, followed respectively by the RBFNN and BPNN.

References

1. M. T. Haque and A. Kashtiban, "Application of neural networks in power systems; a review," *Power,* vol. 2005, 2000.
2. H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Transactions on power systems,* vol. 16, pp. 44-55, 2001.
3. W. Charytoniuk and M. Chen, "Neural network design for short-term load forecasting," in *Electric Utility Deregulation and Restructuring and Power Technologies, 2000. Proceedings. DRPT 2000. International Conference on*, 2000, pp. 554-561.

4. A. Sinha, "Short term load forecasting using artificial neural networks," in *Industrial Technology 2000. Proceedings of IEEE International Conference on*, 2000, pp. 548-553.
5. G. Chicco, R. Napoli, and F. Piglione, "Load pattern clustering for short-term load forecasting of anomalous days," in *Power Tech Proceedings, 2001 IEEE Porto*, 2001, p. 6 pp. vol. 2.
6. M. Gavrilas, I. Ciutea, and C. Tanasa, "Medium-term load forecasting with artificial neural network models," in *Electricity Distribution, 2001. Part 1: Contributions. CIRED. 16th International Conference and Exhibition on (IEE Conf. Publ No. 482)*, 2001, p. 5 pp. vol. 6.
7. M. Kandil, S. M. El-Debeiky, and N. Hasanien, "Long-term load forecasting for fast developing utility using a knowledge-based expert system," *IEEE transactions on Power Systems,* vol. 17, pp. 491-496, 2002.
8. M. A. Momani, W. H. Alrousan, and A. T. Alqudah, "Short-Term load Forecasting based on NARX and radial basis neural networks approaches for the Jordanian poer grid " *Jordan journal of electrical engineering* vol. 2 No 1 2016, pp. 81-93, 2004.
9. J. Buitrago and S. Asfour, "Short-term forecasting of electric loads using nonlinear autoregressive artificial neural networks with exogenous vector inputs," *Energies,* vol. 10, p. 40, 2017.
10. W.-Y. Chang, "Short-Term Load Forecasting Using Radial Basis Function Neural Network," *Journal of Computer and Communications,* vol. 3, p. 40, 2015.
11. K. Lee, Y. Cha, and J. Park, "Short-term load forecasting using an artificial neural network," *IEEE Transactions on Power Systems,* vol. 7, pp. 124-132, 1992.
12. K. Kalaitzakis, G. Stavrakakis, and E. Anagnostakis, "Short-term load forecasting based on artificial neural networks parallel

implementation," *Electric Power Systems Research,* vol. 63, pp. 185-196, 2002.

13. R. J. Mammone, *Artificial neural networks for speech and vision* vol. 4: Chapman & Hall, 1994.
14. F. LiMin, "Neural networks in computer intelligence," *McGraw-Hill International Series in Computer Science,* 1994.
15. E. Diaconescu, "The use of NARX neural networks to predict chaotic time series," *WSEAS Transactions on Computer Research,* vol. 3, pp. 182-191, 2008.
16. J. Huang, H. Jin, X. Xie, and Q. Zhang, "Using NARX neural network based load prediction to improve scheduling decision in grid environments," in *Natural Computation, 2007. ICNC 2007. Third International Conference on*, 2007, pp. 718-724.
17. A. Khamis and S. N. S. B. Abdullah, "Forecasting Wheat Price Using Backpropagation And NARX Neural Network," *The International Journal Of Engineering And Science (IJES),* vol. Vol. 3, pp. 19-26, 2014.
18. D. W. Patterson, *Artificial neural networks: theory and applications*: Prentice Hall PTR, 1998.
19. B. Mahadevan, *Operations management: Theory and practice*: Pearson Education India, 2010.
20. C. W. Chase Jr, *Demand-driven forecasting: a structured approach to forecasting*: John Wiley & Sons, 2013.

3

Development and Applications Current Driven Bulk Current Mirrors

Development and Applications Current Driven Bulk Current Mirrors

Abedalhakem Alkowash[(1)], Imhammad Abood[(2)], Abdualbaset Asahi[(3)]

(1) University of Sabratha, Faculty of Engineering, Sabratha
(2) University of Gharyan Faculty of Engineering ,Gharyan
(3) University of Sabratha, Faculty of Engineering, Sabratha

Abstract

My work starts with designing a current mirror for low-voltage low power applications. The current source/sink is a basic building block in capacitance Metal-Oxide-Semiconductor-Transistors Integrated Circuit (CMOS IC) design and is used extensively in analog integrated circuit design. Ideally, the output impedance of a current source/sink should be infinite and capable of generating or drawing a constant current over a wide range of voltages. However, finite values of output resistance(r_0)and a limited output swing required to keep devices in saturation will ultimately limit the performance using the conventional gate-driven current mirror and the low-voltage low-power bulk-driven current mirrors and comparative study between them.

Keywords: current mirror, Metal-Oxide-Semiconductor-Transistors (MOSTs)**,** The Bulk-driven and Gate-Driven MOSFTs

1 Introduction

1.1 Ideal Current Mirror:

A current mirror can be used as an active load in the differential stage of the op-amp. It is a circuit that must mirror the current. After the differential stage of the op-amp, it is the most important circuit block. Therefore, it will be discussed in detail. The current mirror is represented in ideal form in Fig.1a. Its most simple realization is shown Fig. 1b. It consists of $two\ transistors$ with identical v_{GS}.
One is connected as a diode and is driven by i_{IN} . The other one provides output current i_{OUT} at a high impedance level. Since their v_{GS} are the same, the ratio of their currents is given by.

$$\frac{i_{OUT}}{i_{IN}} = B = \frac{(W/L)_2}{(W/L)_1} \qquad (1.1)$$

By choosing this ratio, the output current can be set on the required value with high precision. Usually the channel length L is kept the same for both transistors to achieve good matching [1-2]. Then the ratio B is set by the transistor widths W. Several errors occur, however, that cause deviations from ideal behavior. The requirements of an ideal current source are the following:

1- The current ratio B is precisely set by the (W/L) ratio, independent of temperature.
2- The output impedance is very high, i.e., high R_{OUT} and low C_{OUT} . As a result, the output current is independent of the output voltage, DC and AC.
3- The input resistance R_{IN} IS very low.
4- The compliance (voltage) is low, i.e., the minimum output voltage V_{OUTc}, for which the output acts as a current source, is low.

These requirements now considered for the simple CMOS current mirror.

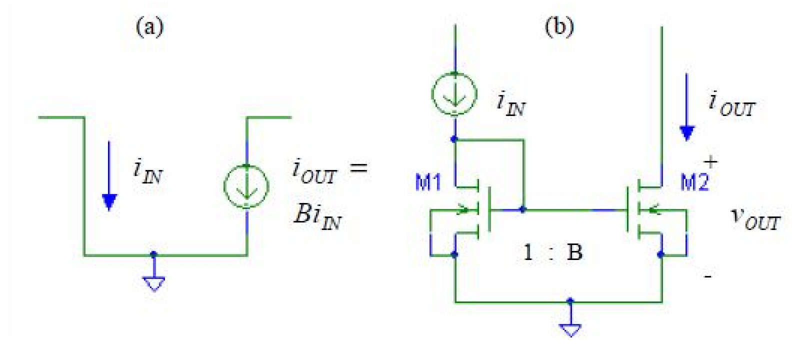


Figure 1: a) principle of current mirrors and 1 b) a simple MOST current mirror.

One of the problems with MOST current mirrors is that a significant voltage must be dropped across the input device. If I design the MOST so that it is operating with the bulk-source slightly forward biased, it can be avoided the requirement for this voltage drop by using the MOSTs from the bulk as the input devices with the gate at a constant voltage. This technique permits the transistor to operate in strong inversion and result in dc currents equivalent to higher voltage designs [3].

1.2 Simple MOST Current Mirror

A N-type version of the proposed low-voltage current mirror` is shown in Figure 2. Instead of the gate-drain diode connection used in the conventional simple current mirror, this new current mirror has a bulk-drain connection. Also, the bulks of M_1 and M_2 are tied together rather than the gates. Instead, the gates of M_1 and M_2 for the N-type version go to the most positive voltage available V_{DD}.

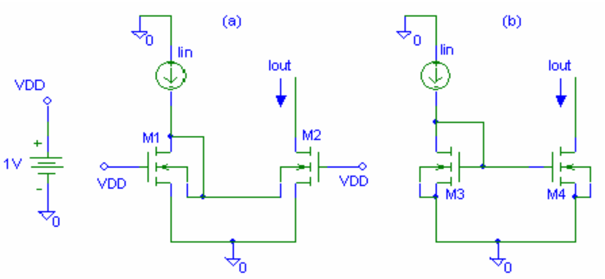


Figure.2. Simple current : a) Bulk-driven, b) Gate-driven.

The drain current versus the bulk-source voltage for a fixed gate-source voltage of the bulk-driven MOST is shown in Figure 2 (a).The drain current versus the gate-source voltage of the Bulk-source voltage is also shown. It is observed that the bulk-driven MOST is equivalent to a Junction Field Effect Transistor (JEFT) where both drain-source current saturation (I_{DSsat}) and Threshold voltage (V_T) are dependent on the fixed gate-source voltage.

There are big similarity between MOST driven from the gate and from the bulk, the following several equations show the drain current in linear and saturation reign for both of gate-driven and bulk-driven MOST.

First order theory gives the drain current i_{DS} of Gate-Driven MOST as

Table 1. The Bulk-driven drain current

Reign	Gate-Driven	Bulk-driven
Linear $v_{DS} < v_{GS} - v_T$	$\boldsymbol{i_{DS} = \beta\left(v_{GS} - v_T - \frac{v_{DS}}{2}\right) v_{DS}}$	$\boldsymbol{i_{DS} = \beta\left(v_{GS} - v_{TO} \mp \gamma\left(\sqrt{2\lvert\delta\Phi_F\rvert - v_{BS}} - \sqrt{2\lvert\Phi_F\rvert}\right) - \frac{v_{DS}}{2}\right) v_{DS}}$
Saturation $v_{DS} \geq v_{GS} - v_T$	$\boldsymbol{i_{DS_{Sat}} = \frac{\beta}{2}(v_{GS} - v_T)^2(1 + \lambda v_{DS})}$	$\boldsymbol{i_{DS_{Sat}} = \frac{\beta}{2}\left(v_{GS} - v_{TO} \mp \gamma\left(\sqrt{2\lvert\delta\Phi_F\rvert - v_{BS}} - \sqrt{2\lvert\Phi_F\rvert}\right)\right)^2 (1 + \lambda v_{DS})}$

Where

$$KP_{n,p} = \mu_{n,p}C_{OX} \quad \text{transconductance parameter} \left({}^{A}\!/_{V^2}\right)$$

$$\beta = \frac{W}{L}KP_{n,p}$$

$$\mu_{n,p} = \frac{v}{E} \text{ mobility in the channel } (cm^2/V.s).\, \mu_n \approx 3\mu_p$$

The small signal input resistance and output resistance of Figure 3 can be found as

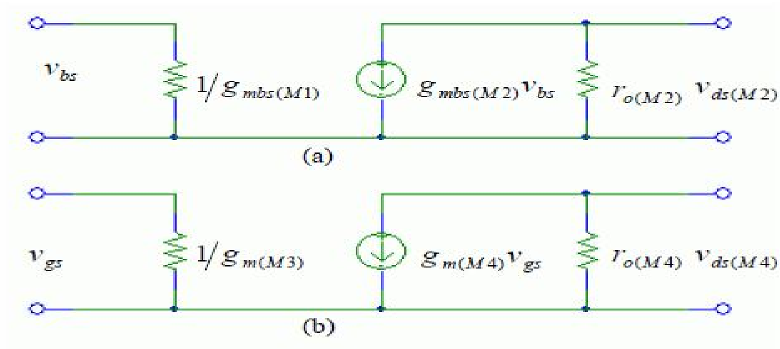


Figure 3: Small signal equivalent circuit of Simple current mirror: a) Bulk-driven, b)Gate-driven

Table 2: The bulk-driven and gate-driven input-output resistance

	The bulk-driven (a)	The gate-driven
r_{in}	$= \frac{1}{g_{mbs}} = \frac{dv_{BS}}{di_D} = \left(\frac{dv_{GS}}{di_D}\right)\left(\frac{dv_{BS}}{dv_{GS}}\right) = \frac{2\sqrt{2\emptyset_F - V_{BS}}}{\gamma g_{ms}}$	$= \frac{1}{g_{m(M3)}}$
r_{out}	$= \frac{1}{\lambda I_{DS(M2,M4)}}$	

Where I_{DS} is proportional to $\left(V_{GS(M2,M4)} - V_T\right)$. These values are in the same range as gate-driven mirrors.

2 Characteristics of Bulk-Driven Current mirror:

The most important characteristic of a current mirror is its current ratio. Therefore it is investigated.

2.1 Current Ratio

The current ratio is given by Eq.(1.1) .An error occurs because the finite output resistance of both transistors is present. Transistor M_1 operates at low $v_{DSI} = v_{BSI}$, whereas M_2 operates at another $v_{OUT} = v_{DS2}$, which is probably much higher [4-5]. Its value is determined by the load, which could be a resistor, a differential stage, etc. Thus, an error in current Δi_{OUT} occurs, as shown, as shown in Fig. 3.4. It is given by

$$\Delta i_{OUT} = \lambda(v_{DS2} - v_{DS1}) = \frac{v_{DS2}-v_{DS1}}{V_{En}-L_2} \qquad (1.2)$$

The error can be reduced by using large values of transistor length L_2 , but especially by enforcing equal v_{DS} values on both transistor. This can be realized by addition of more transistors.

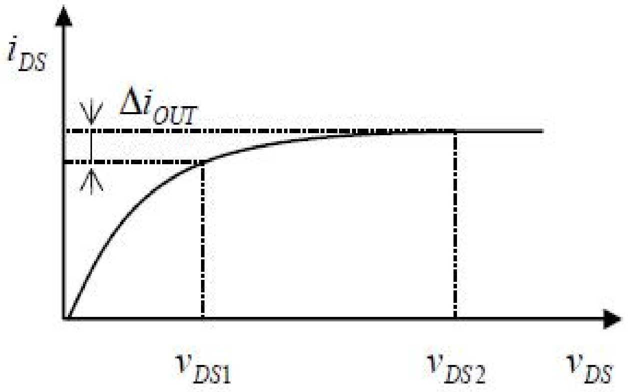


Figure 4: Current error because of different v_{DS}

At high frequencies the current the ratio is impaired as well. The small-signal equivalent circuit of the current mirror seen in Fig. 2a is given in Fig 5. All transistor capacitances are included. It is clear that the current mirror behaves as any other two-node amplifier. Thus it has two poles and one zero.

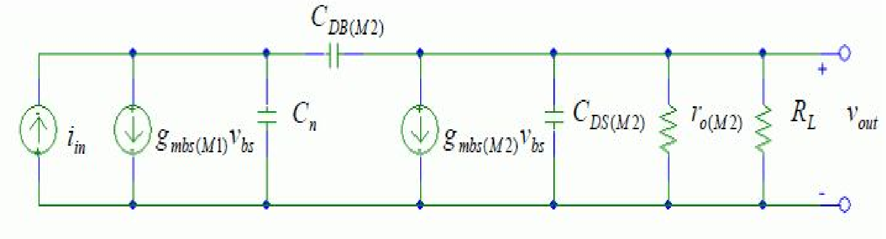


Figure 5: Small-signal high frequency circuit of bulk-driven current mirror

The input node is at an impedance level that is quite low, i.e., $^1/_{g_{mbs(M1)}}$. Therefore the effect of $C_{DB(M2)}$ IS usually negligible.

The dominant pole of the current ratio transfer characteristic is then given by

$$f_p = \frac{g_{mbs(M1)}}{2\pi C_n} \qquad (2.2)$$

In which $C_n = C_{bs(M1)} + C_{bs(M2)} + C_{DS(M1)}$.

This pole is normally situated at quit high frequencies because of the low value of $g_{mbs(M1)}$. Thus the current mirror operates well up to high frequencies.

2.2 Output Impedance

The output impedance is simply the output resistance r_{02} of the output transistor see Fig. 2 in parallel with an output capacitance. It is independent of the impedance of the current source with which the input transistor is driven. The value r_{02} can be made high by increasing the transistor length L. Very high values are difficult to realize, however. Therefore, other configurations are required, such as those used with cascades.

The output capacitance is simply $C_{DB(M2)} + C_{DS(M2)}$ see Fig. 5 in which $C_{DS(M2)}$ is are difficult to achieve; this can be a severe limitation at high frequencies.

2.3 Input Resistance

The input resistance of the bulk-driven is bigger than the input resistance of the gate-driven because it is given by $^1/_{g_{mbs(M1)}}$. Thus, it is easy to design a current source with value i_{IN}, which has an output resistance much higher than the resistance $^1/_{g_{mbs(M1)}}$ [6].

2.4 Compliance V_{OUT}

The compliance voltage V_{OUT} is the minimum output voltage at which the current mirror still provides a high output resistance. It is given by the value of $V_{GSI} - V_T$ can be decreased by taking large values of $(^W/_L)$ [7]. In strong inversion it is given by

$$V_{OUT} = V_{GSI} - V_T = V_{DS_{sat2}} = \sqrt{\frac{I_{OUT}}{K(^W/_L)}} \qquad (3.2)$$

The current mirror of Fig. 2 is the simplest and thus it is used more often than any other. Nevertheless, for precision circuits there is a need for a current mirror with higher output resistance r_0 and with less error in current Δi_{OUT}. I will introduce other configurations that have less error.

I provided current mirrors using both bulk-driven and gate-driven MOSTs for low power applications. I provided the simulation by Orcad Pspice using model $0.7\ \mu m$ [8]. The results show that the input voltage drop for the bulk-driven mirrors can be much less as depicted on the following figure (6).

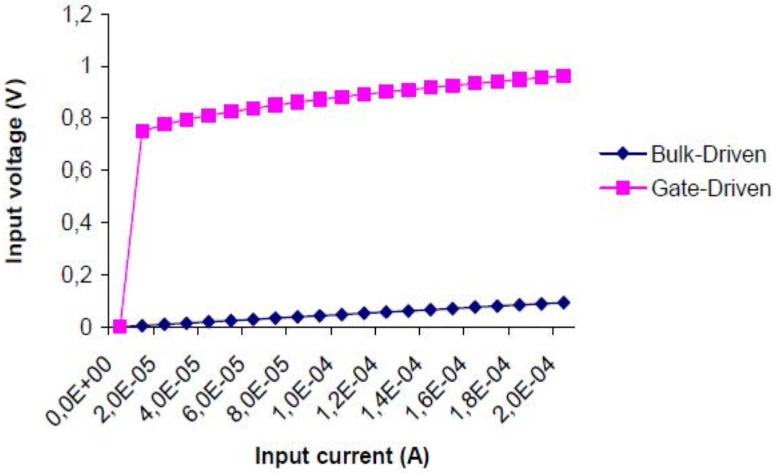


Figure 6: Input voltage vs. Input current characteristics

I can give an example, $200\mu A$ input current, the value of $V_{DS(M3)}$ for the gate-driven mirror was 0.96V and the bulk-driven mirror it was 0.09V. The small signal output resistances are approximately the same. The input-output current linearity of the gate-driven mirrors is absent in the bulk-driven mirror because the output transistor M_2 is operating in saturation see Figure 7and see Figure 8.

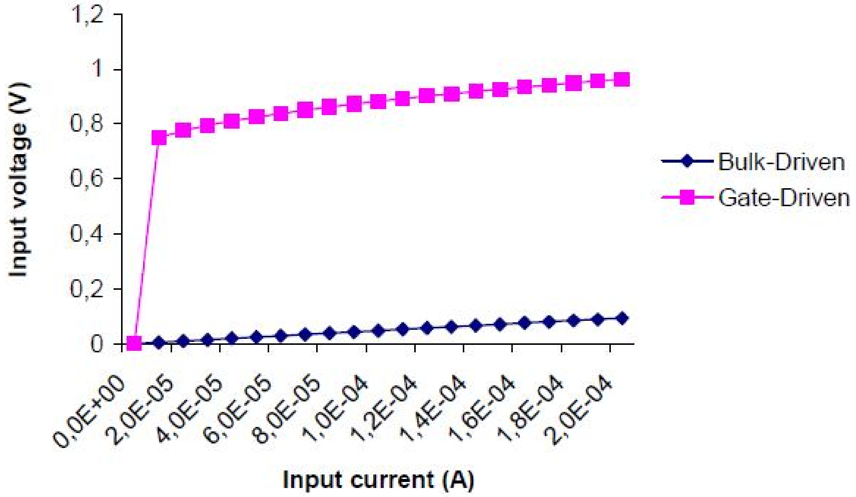


Figure 7: Input-output transfer characteristics.

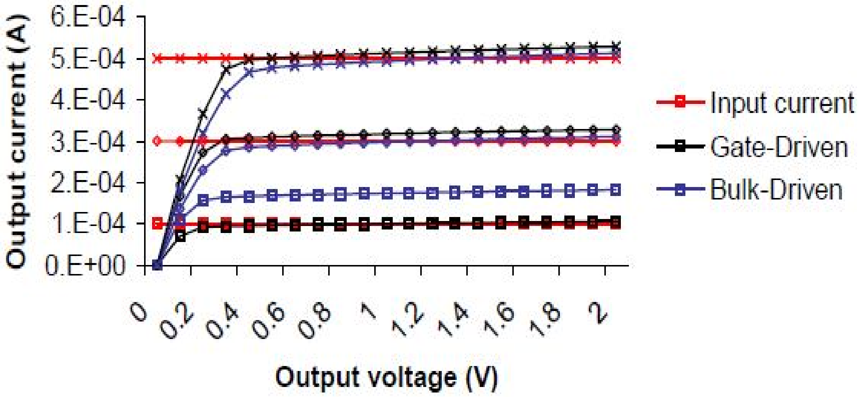


Figure 8: Drain current vs. Bulk-source voltage Characteristics.

3 Conclusions

The aim is to design low-voltage low- power bulk-driven current mirror and current source, explanation about the principle of the bulk-driven current mirror is provided, and current mirrors using both bulk-driven and gate-driven MOSTs have been designed. Since the output resistance is one of the most important performance parameters for a current mirror a folded cascade has been introduced too. Last subsection was the enhanced bulk-driven current mirror which remove the main drawback of the bulk-driven current mirror that occurred, and they are mainly the input-output current linearity of the gate-driven mirrors is absent in the bulk-driven mirrors.

References

1. Kenneth R, Willy Sansen: Design of Analog Integrated Circuits and Systems, 1994, ISBN: 0-07-113458-1
2. Johan H. Huijsing: Operational Amplifiers Theory and Design, by Kluwer Academic Publishers, 2001,ISBN: 0-7923-7284-
3. Kimmo Lasanen, et al.: A 1-V 5 µW CMOS-Opamp with Bulk-Driven Input Transistors. 2000, Proc.43rd. IEEE Midwest Symp. On Circuits and Systems.
4. Cheng-Fang Tai , Member, Jui-Lin Lai and Rong-Jian Chen, Senior Member,"Using Bulk Driven Technology Operate in Subthreshold Region to Design A Low Voltage and Low Current Operational Amplifier",IEEE 2006.
5. G. Raikos and S. Vlassis, "0.8 V bulk-driven operational amplifier, "Analog Integr. Circuits Signal Process, vol. 63, no. 3, pp. 425–432, 2010.
6. F. Khateb, D. Biolek, Bulk-driven current differencing transconductance amplifier, Circuits, systems, and Signal Processing, 2011, pp. 1-9.
7. Liang Zou, Member and Syed K.Islam, " Low-Voltage Bulk-Driven Operational Amplifier with Improved Transconductance" IEEE Transation on circuits and sustem-1 2013.

8. Blalock BJ and Allen PE. 1995 ” A low-voltage, Bulk- Driven MOSFET current Mirror for CMOS Technology” IEEE International on Circuit and System, VOL.3,1995,PP.1972-1975.

Sematic of Parallel Primitive Haskell Programming Language

4

Sematic of Parallel Primitive Haskell Programming Language

Dr. Mustafa Kh. Aswad
Department of Computer Engineering & IT
Subratha University, Subratha, Libya
Email: mustafaasawd@gmail.com

Abstract

Nowadays, heterogeneous multi-core has become the mean-stream computer architecture. It emerging hundreds of cores and combining accelerators like GPUs on traditional chip. Programming software models need to exploit all the resources at the hand of a programmer with minimum efforts. This paper describes the Semitics of parallel primitives of Haskell Programming Language. Primitives are exploiting the underlying architecture resources. New primitive added to a programming language we need to prove its expected behavior. This paper specified the expected behaviors of the constructs by Haskell functions, achieving an executable specification. We have formulated several properties as Haskell predicates and used ***Quickcheck*** to check them on random input. The three basic properties represent sanity checks of the semantics. Two proposed implementation relevant properties did not hold, and counter examples extracted from ***Quickcheck*** identified diffusion of sparks to be the problem. In the implementation, we avoided this problem by resetting the boundaries after one fishing stage. The final property, checked with ***Quickcheck***, shows that with this modification, the desired property holds.

1. Introduction

The introduction of multi-core processors has renewed interest in parallel functional programming and there are now number of parallel programing models that explore the advantages of a functional language for writing explicitly parallel code or implicitly paralellizing code written in a pure functional language [2]. There are more developments in the areas of softwares such as transactional memory and nested data parallelism [6] [11].

Parallel programs are written to gain performance. This goal is achieved by exploit the potential of a real parallel computing resource like a multi-core processor [8]. Concurrency is a programming techniques that allows us to model computations as hypothetical independent activities. Those computations can communicate and synchronize [10]. This paper describes in Section [2.1] number of parallel programming model include explicit, semi-explicit and implicit parallel programs. It will know that fully implicit parallelism leads to very small tasks which con not hide the over hide communications. Also the explicit parallel program is too difficult for non-experience programmer. This paper propose the semi-explicit parallel programing as good approach for writing parallel programming, and easy to control the granularity of parallel tasks on modern operating systems and processors. Haskell function language is a semi-explicit parallel programming model constrains performance and simplicity. It is particularly fox on formal semantics for the new constructs.

2. Background

This section presents an introduction to number of parallel programming models. It in mainly constrain on the techniques for writing concurrent parallel programs. The parallel program is written gain performance. In other-words it written to reduce the execution time. Therefore, we need to exploit the potential of a real parallel architecture platform like multi-core cores.

2.1. Parallel Applications

The development of parallel programs increases the number of challenges that already exist in developing sequential programs. A parallel programs aims to achieve performance cross different parallel platform without changes and hide the latency between cores and I/O operations to disks and network devices. A parallel program should remain easy to constructs.

As mentioned the new mainstream microprocessors are move towards hundreds of cores or more in one platform. To achieve performance from each individual core, it is necessary to split a given work into tasks and distribute these tasks across multiple processing cores. Splitting a work to number of tasks requires a program language capable to automatically parallelize the sequential code or semi-explicit or explicitly parallel program which is then scheduled onto multiple cores by the operating systems. Haskell function language is semi-explicit parallel programming.

2.2. Haskell Functional Language

Haskell Functional Language represents code in mathematical function sense. One of its features is laziness which means functions don't evaluate their arguments. This feature helps to write parallel code very easy in Haskell functional language.

Exploiting Parallelism in Haskell, in general every expression in most functional language can be evaluated in parallel. This can be automatically exploited. But exploiting all parallelism in a program has side effects. It creates too many small tasks. Which cannot be efficiently scheduled and parallelism is limited by fundamental data dependencies in the source program. Haskell provides a mechanism to allow the user to control the granularity of parallelism by indicating what computations may be usefully carried out in parallel. This kind of parallelism calls Semi-Explicit Parallelism. It provides two primitives par and pseq [12].

```
 par :: a -> b -> b
pseq :: a -> b -> b
```

Fig 1: par and pseq Function

The function par indicates to the Glassgow Haskell Compiler (GHC) run-time system that it may be beneficial to evaluate the first argument (a) in parallel with the second argument (b). It returns the second argument as result. The notion of a lazy future provided by Haskell language allow the runtime system to create spark for Frost arguments (a) which has the potential to be executed on a different thread from the parent thread. But not necessarily create a thread to compute the value of the expression (a).

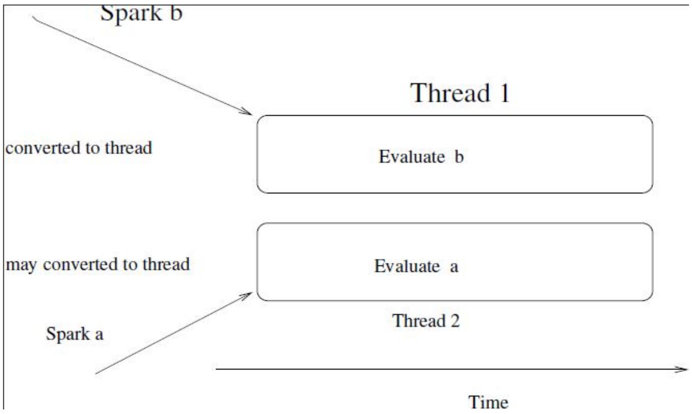


Fig 2: Semi-explicit Parallel Haskell.

Load balance in Haskell is a mechanism used to distribute the work among the participated cores. Haskell uses a work stealing algorithm to distribute a work. Within a multicore it will search for a spark by directly accessing spark pool [4]. A spark is described in the Section [2.2] in a network it sends a fish message searching for work. It is a dynamic mechanism for automatically distributing work and data on a cluster as shown in Figure 3.

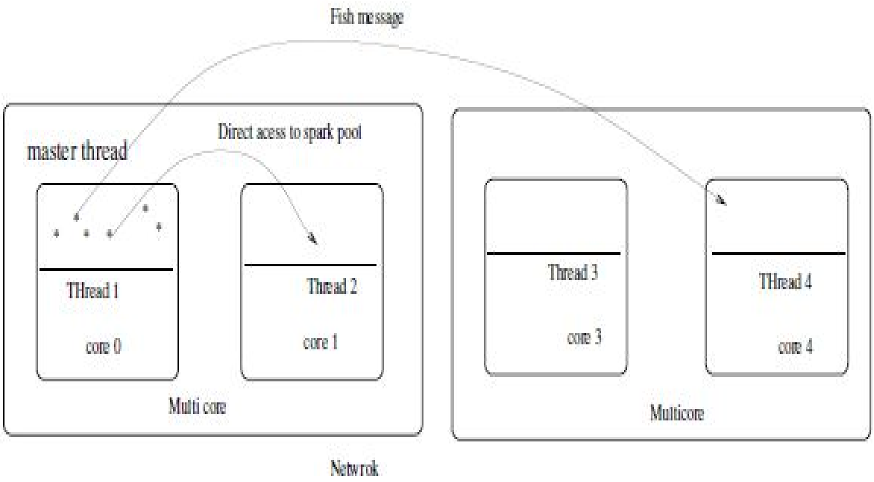


Fig 3: Workstealing Algorithm in Parallel Haskell.

2.3. OpenMP

OpenMP uses an imperative parallel programming style [4]. It is a portable programming interface for shared memory multithreaded programming using C/C++ and FORTRAN as host languages. OpenMP consists of a set of compiler directives, library routines, and environment variables that affect run-time behavior. OpenMP uses a fork-join threading model; a master thread forks a task into a number of worker threads that share the work and then wait until they finish to join before continuing. OpenMP is a scalable model that gives programmers a simple and flexible interface for developing parallel applications for a range of parallel architectures. The model is identified as easy to use and portable. The programmer does not need to put significant effort into parallelizing the existing sequential program. However, this is not always the case, as the multicore resources are not fully utilized if the programmer is not expert in parallel programming.

2.4. Message Passing Interface (MPI)

A standard defines an interface for sending and receiving messages [13]. Specifically the interface includes point-to-point communication functions, ***send*** operations performing a data transfer between two concurrently executing tasks, and ***receive*** operations to accept data from another processor into program memory space. It also has other operations, such as broadcast barriers and reduction that explicitly involve a group of processors. It is heavily used in high performance computing and, with considerable tuning, delivers an acceptable performance across a wide range of architectures. MPI is a MIMD style model. However a shared-memory style can be simulated using send and receive messages of MPI. MPI does not provide the dynamic creation or deletion of processes during a program runtime (the total number of processes is fixed [9].

3. Constructs Semantics

The new constructs are not prescriptive: rather than specifying a single PE for a task, they identify sets of PEs within the communication hierarchy of the architecture. We present a simple Haskell specification of the sets of PEs that each construct identifies when executed on any PE of participating PEs. We first need some mechanism specifying paths and distances in the tree hierarchy.

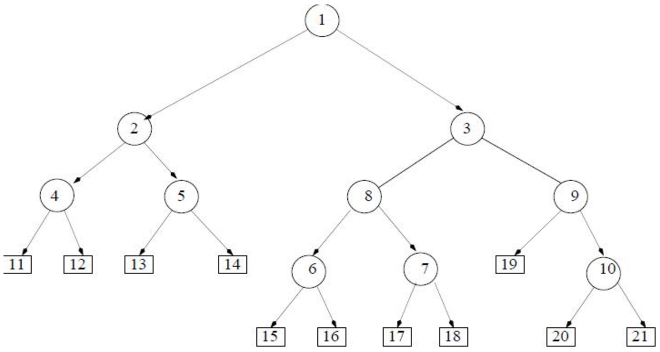


Fig 4: An Example of Hierarchical Architecture.

3.1. Distance Function

In order to illustrate how the ***distance*** function is working, we define a binary tree ***(Tree t)*** structure representing an underlying parallel platform (e.g. the one shown in Figure(4). The definition of the tree data structure is as follows:

```
data Tree a = Node a (Tree a) (Tree a)

                    |  Leaf {pId ::Int} deriving (Eq, Show)
```

A tree is a leaf with a PE Id as value, or a node represents network possibly parametrized with information such as latency, leaves represents PEs. A node has a value and two branches, each of which is a subtree.

The function ***distance t p1 p2*** calculates the distance, defined as the number of steps to the nearest common node in the hierarchy between two leaves in the architecture hierarchy. The function takes a ***tree t*** representing the architecture and two leaves ***p1*** and ***p2*** as input and returns the distance between the two leaves as an integer.

```
distance ::Tree Int ->Int ->Int ->Int
distance  t  p1  p2  =  d1+d2
        where
           pathTo  p1 = path t  p1
           pathTo  p2 = path t  p2
           comNodes = length (prefixOf   pathTo p1 pathTo p2)
           d1 = length pathTop1 - comNodes
           d2 = length pathTop2 - comNodes
path ::  Tree Int ->Int -> [Int]
path (Leaf p ) s
      | p==s = [s]
      | otherwise = []
path  (Node v t u) s
      | v == s =  [v]
      | left== [] && right  == [] = []
      | left /= [] =   [v]  ++ left
      | right  /= [] = [v] ++ right
      where
        left = path t s
                right = path u s
```

Fig 5: Distance Function

The definition of the distance function uses additional auxiliary functions, path shown in Figure 5. The call path ***(Node v t u) p*** calculates the path to leaf p from the root of the tree represented as a list. It takes a tree ***(Node v t u)*** and leaf p and returns list of nodes that lead to the leaf p. The complete Haskell program defining all auxiliary functions, e.g. for simplicity, we use a tree of integers shown in Figure 4 to demonstrate the constructs semantics. Squares in the tree represent PEs in the hierarchy (Leaf). Circles in the tree represent the networks connecting Pes or sub-networks (Node).

As an example, if we need to compute a distance between Leaf 15 and Leaf 20, we proceed by the following steps.

1) Path to Leaf 15 ➔ pathTop1 = [1,3,8,6,15]

2) Path to Leaf 20 ➔ pathTop2 =[1,3,9,10,20]

3) Length of the longest common prefix of pathTop1 and pathTop2 ➔ comNodes = length [1,3] =2

   Node 3 is the nearest common node between Leaf 15 and Node 20

4) Length of path to Leaf 15 from nearest common Node ➔ d1 = length (pathTop1) - length (comNodes) = 3

5) Length of path to Leaf 20 from nearest common Node =) d2 = length(pathTop2) - length(comNodes) = 3

6) The distance between Leaf 15 and Leaf 20 = d1 + d2 = 6, is the sum of steps moving from one leaf to the nearest common parent and down to other leaf.

```
setparDist :: Tree Int ->Int ->Int ->Int -> [Int]
setparDist t m u p
    | ((m<0) || (u<0)) = []
    | ((m==0) && (u==0))= [p]
    | (u > (length (pp)-1)&& (m==0)) =(rLeaf ( t))
    | ((m==u)||(u > (length (pp)-1)))
              = exact ( subexact) p
    | m==0 = [p]++setPes
    | otherwise = setPes
    where
     pp = path  t p
commonnu = last (take (length(pp) - u) pp)
commonnm =last ( take (length(pp) - m) pp)
subu=subTree t commonnu
```

```
subexact= subTree t commonnm
complementtree = (complementTreesubu
commonnm)
setPes= filter (/= commonnm) (rLeaf
                    (complementtree))

subTree ::Tree Int ->Int -> ( Tree Int)
subTree (Leaf  p) s =EmptyTree
subTree (Node v t u) s
   | v == s  = (Node v t u)
   | left== EmptyTree&& right == EmptyTree
                         = EmptyTree
   | left /= EmptyTree = left
   | right  /=EmptyTree =right
   where
    left =    subTree t s
    right =   subTree u s

complementTree ::  Tree Int ->Int -> Tree Int
complementTree (Leaf p1) s = (Leaf p1)
complementTree  (Node v l EmptyTree ) s
    | v==s = EmptyTree
    | otherwise =(Node v (complementTree l s)
EmptyTree)
complementTree (Node v t u) s
      | v==s = EmptyTree
      | otherwise  = (Node v
                 (complementTree t s)
                                (complementTree u s))
```

Fig 6: setparDist Locations Function

3.2. setparDist Function

The most basic primitive we propose is the ***parDist*** primitive. We therefore start by defining its semantics in terms of the possible locations defined by it. A ***setparDist t m u p*** specifies the set of PEs on which a ***parDist m u*** task may be executed from **PE$_p$** in an architecture ***t***. It takes an architecture tree ***t***, a minimum bound ***m***, maximum bound ***u***, and a leaf ***p***, the current location, as input and returns a list of all possible PEs (Figur [6]). For example, if we need to generate sparks intended to be executed between levels 1 and 3 from Leaf 20 of the tree shown in Figure (4), we perform the following:

1) Calculate path to Leaf 20 =) pathTop = [1,3,9,10,20].

2) Calculate the common node distant by u levels from Leaf 20 ➔common u = last (take (5 - 3) [1,3,9,10,20]) ➔ 3.

3) Calculate the common node that is m levels from Leaf 20 ➔common m= last (take (5 - 1) [1,3,9,10,20]) ➔ 10.

4) Calculate the subtree of the common u leaf 3 ➔ subtree u = (Node 3 (Node 8 (Node 6 (Leaf pId = 15)(Leaf pId = 16)) (Node 7 (Leaf pId = 17) (Leaf pId = 18))) (Node 9 (Leaf pId = 19) (Node 10 (Leaf pId = 20)(Leaf pId = 21))))

5) Calculate the complementary tree of the common node 10. The complementary tree is the original tree excluding the subtree of a given node. In our case, we calculate the complement for ***subtree u*** of node 10. ➔complementtree = Node 3 (Node 8 (Node 6 (Leaf pId = 15) (Leaf pId = 16)) (Node 7 (Leaf pId = 17) (Leaf pId = 18))) (Node 9 (Leaf pId = 19))

6) Finally calculate the leaves of the complementtree subtree ➔setPes = [15,16,17,18,19].

```
setparBound ::Tree a ->Int ->  a -> [a]
setparBound t n p = setparDist t 0 n p
```

Fig 7: setparBound Locations Function

3.3. setparBound Function

As mentioned in the previous section, ***parDist*** is the most basic primitive. We can use it to define the other constructs. A ***setparBound t n p*** (Figure 7) specifies the set of PEs that tasks generated by a ***parBound n*** may be executed on, from P Ep in architecture t. For example if we need to generate sparks bounded by two levels from leaf 11 of the tree shown in Figure 4, we just call ***setparDist*** with the following parameters ***t 0 2 11***, where t is the tree. The result is [11, 12, 13, 14], a list of leaves with a distance of at most 2 in the architecture tree (t)

3.4.setparAtLeast Function

A ***parAtLeast*** is similar to ***parBound***, as it takes an additional integer parameter specifying the minimum distance in the communication hierarchy that the computation may be communicated. Therefore, it can be defined in a similar way, as shown in Figure 8.

```
setparAtLeast::(Ord a, Show a)=>Tree a->Int ->  a -> [a]
setparAtLeast t  n  p = setdFun  t  n maxLevel  p
        where
           maxLevel = 3
```

Fig 8: setparAtLeast Locations Function

So, if we need to generate sparks intended to be executed at least two levels from leaf 11 of the tree shown in Figure 4, we just call ***setparDist*** t 2 ***maxLevel*** 1 1, where t is the tree and ***maxLevel*** is the maximum distance that sparks can be sent within the architecture hierarchy. In this example, the result is [15, 16, 17, 18, 19, 20, 21].

3.5. Construct Properties Test

This section presents implementation-relevant properties that the architecture-aware semantics should satisfy. These properties are expressed as Boolean functions in Haskell and validated using ***QuickCheck*** [5] that is the properties are written as Haskell functions and can be automatically checked on either random input or with custom test data generators. Two types of properties are tested: the basic properties and specialized properties.

1) Basic Properties: Let P be the set of PEs which are the leaves of the tree representing a given hierarchical architecture. The domain H is the domain of all possible tree hierarchies. The domain P is the domain of all possible processor elements.
   a. Basic property one. For any processing element $p \in P$; the only possible placement of a bounded spark withupper and lower bounds of 0 and 0 is the p itself. Formally, this is written as:

   $$\forall h \in H; \forall p \in P \text{setparDist h 0 0 p} = \{\text{p}\}$$

   b. Basic property two. Let path p be a function that returns the longest path to the PE from the root of the tree hierarchy. Let rLeaf h be a function that returns all processor elements (PEs) in the tree hierarchy. The path and rLeaf functions are described in Section [3.1]. For $p \in P$ and$h \in H$, the set of PEs returned by calling the setparDist h 0 (length (path h p)) p function is equal to the set of all PEs in the hierarchy, as returned rLeaf h.

$$\forall h \in H; \forall p \in P \text{setparDist h 0 (length (path h p)) p = rLeaf h}$$

c. Basic property three. If the upper bound u is less than 0 or lower bound m is greater than the longest path to the PE from root of the tree hierarchy then the set of PEs returned by calling ***setparDist h m u p*** function is an empty set of PEs.

$$\text{setparDist h m u p} = \{\} \quad \forall \text{ h Z}$$

The above three basic properties can be considered sanity checks of the semantics. All basic properties have passed Quickcheck testing using one hundred randomly generated inputs, each with a randomly generated tree hierarchy.

2) Specialised Properties: The proposed architecture-aware model exposes the tree hierarchy to the programmer through the ***parDist*** primitive. The ***parDist*** primitive provides a mechanism to spark tasks that can be executed in certain levels of the architecture hierarchy. We believe, for the implementation, it is important that these sparks do not leave their neighborhood, where neighborhood is the set of PEs specified by ***parDist*** primitive. Otherwise the bounded spark may diffuse to arbitrary locations after several steps of fishing (work stealing, as outlined in Section [2.2]. We define the following property to formally specify and check this property.

a. **Specialised proposed property one**. This property reflects the initial intention of the bounded ***parDist***. For $p \in P$ and $h \in H$, the set of PEs returned by ***setparDist h 0 u p*** is equal to the set of PEs returned by ***setparDist h 0 u p'***, where p' is a possible location after one step of fishing. The aim is to guarantee that if the spark is fished again from p' it will be executed in the same neighborhood specified by the original p.

$\forall h \in H;\ \forall p;\ p' \in P;\ u\ 2\ Z\ ;\ p' \in$ (setparDist h 0 u p))

setparDist h 0 u p' = setparDist h 0 u p

On closer examination, this property fails under the ***quickcheck*** test. In the case of an unbalanced tree hierarchy, the result of ***setparDist h 0 u p*** may return a subset of the set returned by ***setparDist h 0 u p***.

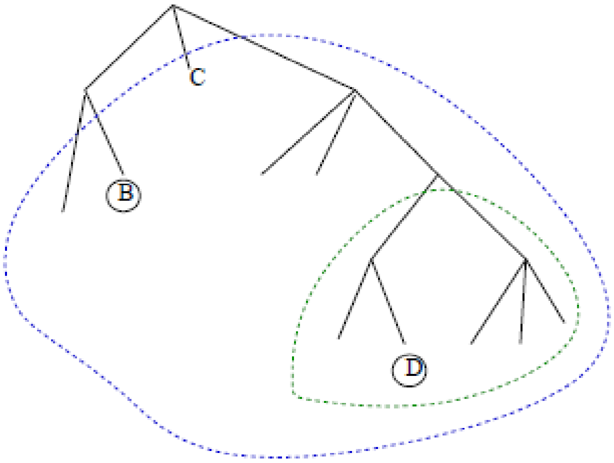


Fig 9: Tree Example of Specialized Proposed Property One

For example, in the tree hierarchy shown in Figure 9, if PE (B) launches a spark with boundaries 0 and 2, then any PE in the outer circle can fish the spark. In particular, it can be fished by PE (D). In second step the spark can be fished only from PEs in the inner circle. That is why the property fails. However, this is not always true, illustrated in the next proposed property.

b. Specialized proposed property two. For $p\ \in P$ and$h\ \in\ H$, the set of PEs returned by setparDist h 0 u p’ is a subset of the set of PEs returned by setparDist h 0 u p.

$$\forall\, h \in H\ ,\forall\, p\,, p' \in P\ , u \ \in\ Z\ \in\ p' \in$$
$$(\,setparDist\ h\ 0\ u\ p)\,)\quad =>$$
$$setparDist\ h\ 0\ u\ p'\ \ \leq\ setparDist\ h\ 0\ u\ p$$

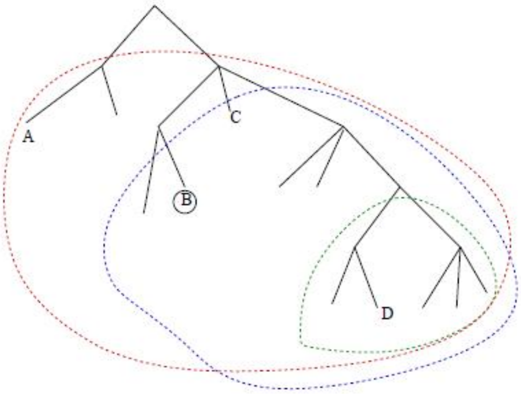


Fig 10: Tree Example of Specialized Proposed Property Two.

The property also fails the ***quickCheck*** test, because of unbalanced tree hierarchies. In the second step, the spark may be fished by a PE which is not one of the elements of the original PE neighborhood. For example, in the tree hierarchy showed in Figure 10, if PE (B) launches a spark with boundaries 0 and 2, then any PE in the blue circle can fish the spark. In particular, it can be fished by PE(C). In a second step, the spark can be fished from PE (C) by any PE in the red circle. In particular, it can be fished by PE (A), which is outside the original neighborhood. We call this behavior of the fishing mechanism diffusion of sparks. In the implementation, we must prevent this scenario from happening. We achieve this by resetting the boundaries of the spark to be 0 and 0, after the first fishing stage. This forces evaluation of the spark on the initial target PE, and thus within the neighborhood specified by the spark.

c. Specialized proposed property two. For p 2 P and h 2 H, after fishing a spark from p to p', which is within the bound u and after resetting the bound u for the spark to 1, this spark can only be fished by a p inside the original neighborhood of p. The set of PEs returned by ***setparDist h 0 1 p'*** is a subset of the set of PEs returned by ***setparDist h 0 u p.***

$$\forall h \in H; \forall p;\ p' \ \in P;\ \ u \ \ \in\ Z;\ \ p' \ \in$$

$$(\,setparDist\, h\, 0\, u\, p)\,)$$
$$setparDist\, h\ 0\ 1\ p' setparDist\, h\ 0\ u\ p$$

This property passes the ***quickCheck*** and guarantees that there is no diffusion of sparks, i.e. sparks always remain in the neighborhood specified by the original **parDist**.

4. Related Work

Many of the semantics primitives prove described in this section are similar to the semantics prove presented in this paper. Berry and el.[3]defined a transitional semantics of a simple language to preserve the expected behavior of sequential programs. Mosses and el. [7] has investigated the sequential behavior of ML action semantics and extended the language with concurrency primitives.

5. Conclusion

We have presented the semantics for the architecture aware constructs, specifying the set of possible locations when providing boundaries to the sparks. We have specified the expected behavior of the constructs by Haskell functions, achieving an executable specification. We have formulated several properties as Haskell predicates and used ***Quickcheck*** to check them on random input. The three basic properties represent sanity checks of the semantics. Two proposed implementation relevant properties did not hold, and counterexamples extracted from ***Quickcheck*** identified ***diffusion of sparks*** to be the problem. In the implementation, we avoided this problem by resetting the boundaries after one fishing stage. The final property, checked with ***Quickcheck***, shows that with this modification, the desired property holds.

References

1. ARCHIBALD, B., MAIER, P., STEWART, R., TRINDER, P., ANDDE BEULE, J. Towards generic scalable parallel combinatorial search .In Proceedings of the International Workshop on Parallel Symbolic Computation (New York, NY, USA, 2017), PASCO 2017, ACM, pp. 6:1–6:10.
2. BARROSO, L. A., GHARACHORLOO, K., MCNAMARA, R., NOWATZYK, A., QADEER, S., SANO, B., SMITH, S., STETS, R., AND VERGHESE, B. Piranha: A scalable architecture based on single-chip multiprocessing. SIGARCH Comput. Archit. News 28, 2(May 2000), 282–293.
3. BERRY, D., MILNER, R., AND TURNER, D. A Semantics for ML Concurrency Primitives. 2 1992, pp. 119–129.
4. CHAPMAN, B., JOST, G., AND RUUD, V. Using OpenMP. Portable Shared Memory Parallel Programming. No. ISBN-13: 978-0-262-53302-7. The MIT Press Cambridge, Massachusetts, London, England,2008.
5. CLAESSEN, K., AND HUGHES, J. Quick Check: A Lightweight Tool for Random Testing of Haskell Programs. In Acmsigplan notices (2000), vol. 35, ACM, pp. 268–279.
6. DAMRON, P., FEDOROVA, A., LEV, Y., LUCHANGCO, V., MOIR, M., AND NUSSBAUM, D. Hybrid transactional memory. SIGPLAN Not. 41,11 (Oct. 2006), 336–346.
7. MOSSES, P., AND MUSICANTE, M. An action semantics for ml concurrency primitives. BRICS Report Series 1, 20 (1994).

8. SKILLICORN, D. B., AND TALIA, D. Models and languages for parallel computation. ACM Comput. Surv. 30, 2 (June 1998), 123–169.
9. SNIR, M., OTTO, S., HUSS-LEDERMAN, S., WALKER, D., AND DONGARR,J. MPI: The Complete Reference , vol. 1. The MIT Press, 1998.MIT, Cambridge.
10. SUTTER, H., AND LARUS, J. Software and the concurrency revolution. Queue 3, 7 (Sept. 2005), 54–62.
11. TARDITI, D., PURI, S., AND OGLESBY, J. Accelerator: Using data parallelism to program gpus for general-purpose uses. SIGOPS Oper. Syst. Rev. 40, 5 (Oct. 2006), 325–335.
12. TRINDER, P. W., HAMMOND, K., MATTSON, JR., J. S., PARTRIDGE,A. S., AND PEYTON JONES, S. L. Gum: A portable parallel implementation of haskell. SIGPLAN Not. 31, 5 (May 1996), 79–88.
13. WALKER, D., AND DONGARRA, J. MPI: a Standard Message Passing Interface. Supercomputer 12 (1996), pp. 56–68. ASFRA BV.

5

Performance Evaluation of solar cells after 31 Years of Work

Performance Evaluation of solar cells after 31 Years of Work

Mohamed A. S. Alshushan,
Sabratha University, Faculty of engineering,
mohammedsaad1318@gmail.com

Ibrahim M. Saleh
Tripoli University
Tripoli, Libya
Ibrahim.saleh@lttnet.net

Abstract

Crystalline silicon photovoltaic modules are being used for long time in many photovoltaic applications. It was not expected that photovoltaic modules of old technology will last for twenty years. In contrast, a photovoltaic system which was installed in 1979 in the Libyan Desert is still running with a little decrease in the output power and small changes in its designed parameters. The study goal is to evaluate the performance of thirty-year old crystalline silicon cells under Standard test condition. Some of the solar cells were dismantled from one of the photovoltaic modules which have been working for more than thirty years in order to test and measure their current-voltage curve. This paper presents the measuring results of indoor measurements on the thirty-year old solar cells.

Index Terms standard test condition, Qualification and Testing, solar cells

1. INTRODUCTION

Crystalline PV cells encapsulated in front glass modules have been used for long time in many applications [1]. Long period of run is one of the major technical strengths of the photovoltaic (PV) modules [2]. Manufactures continue to search for new materials in order to reduce cost and improve performance of both solar cells and modules. Many Crystalline silicon module manufacturers offer warranties that their products will survive for certain duration of time [3]. Even though old technologies of crystalline silicon was not expected to last for more than twenty years, solar modules have still been running more than thirty years and they are expected to run for more. [4].

Isolated PV systems have been used in Libya for more than thirty years in microwave repeater stations which are located in Libyan Desert. In this study, a number of solar cells was removed form one of the thirty-year old PV modules and tested to measure their performance “current-voltage curves (I-V carves)”.

2. APPROACH AND METHODOLOGY

The performance of the solar cell is determined by measuring the I-V curve which can tell a lot about the cell performance. The I-V curve is a plot of the current versus voltage. The big challenge in this part is how to remove the solar cells from its module since there is encapsulation liquid around the cells between the front and the back glass. Any a small wrong movement, it will cause to break the cell. Eventually a solution was made to pass this obstacle. The indoor I-V curves of the cell were measured in the laboratories of the Solar Energy Research Center in Tajoura City using Sun Simulator – Solsim see fig. 1. Solsim (Sun Simulator) enables accurate measurement of solar cells and measures the short circuit current I_{sc}, the open circuit voltage V_{oc}, the fill factor FF, the maximum power P_{mpp} with its voltage V_{mpp} and current I_{mpp}, the current density J_{sc} and the efficiency eta. In addition, it can calculate I_{sc}, V_{oc}, and J_{sc}. Furthermore its xenon lamp produces 1000 W/m light intensity which meets standard test condition. The Solsim is also able to change the temperature of measuring cell between 20°C and 60°C.

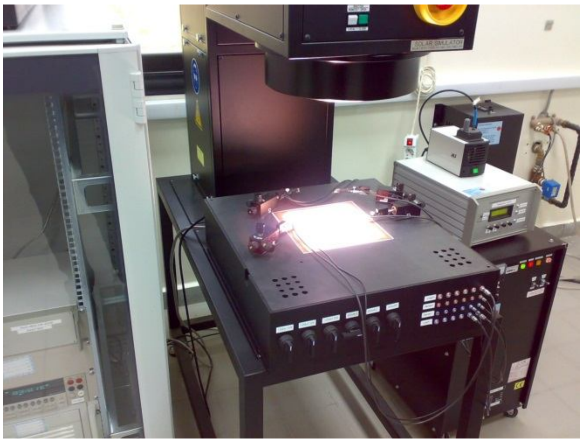


Fig. 1 Sun Simulator – Solsim

3. PREVIOUS WORK

In the paper [2], Visual Inspections and performance measurements were done only for the twenty seven year old PV module. In this paper, the authors indicated that this was one of the longest, if not the longest, exposed and operated arrays ever evaluated in the world, but it will definitely not be the longest one anymore.

4. FIELD OBSERVATIONS

The modules, PQ 10/40/02, were installed in Libya in the late of seventies as stand-alone system as shown in fig. 2. The cells are made of Multicrystalline silicon. In addition, the encapsulation of the PV modules is front and back glass with polyvinyl butyral (PVB), and the frame is made of stainless steel. The dimensions of each module are (459 mm × 1076 mm) with 40 cells (10 cm × 10 cm) each.

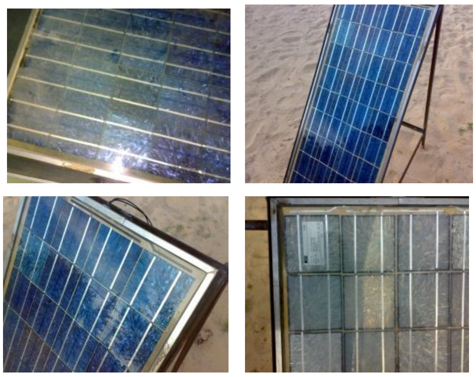


Fig. 2. The module PQ 000876 and 000877

5. MEASUREMENTS AND RESULTS

The table 1 summarizes the indoor measurements for solar cells under standard test condition.

Table (1): performance parameters of solar cell

Temperature [°C]	25
$I_{sc,meas}$ [mA]	2231.940
$J_{sc,meas}$ [mA/cm^2]	22.319
$V_{oc,meas}$ [mV]	541.222
$I_{sc,korr}$ [mA]	2229.600
$J_{sc,korr}$ [mA/cm^2]	22.296
$V_{oc,korr}$ [mV]	540.518
I_{mpp} [mA]	1973.755
J_{mpp} [mA/cm^2]	19.738
V_{mpp} [mV]	422.974
FF [%]:	69.274
P_{mpp} [mW]:	834.848
eta [%]:	8.348

Korr: calculation, Meas: measurement

The fig. 3 shows the I-V carve of solar cell.

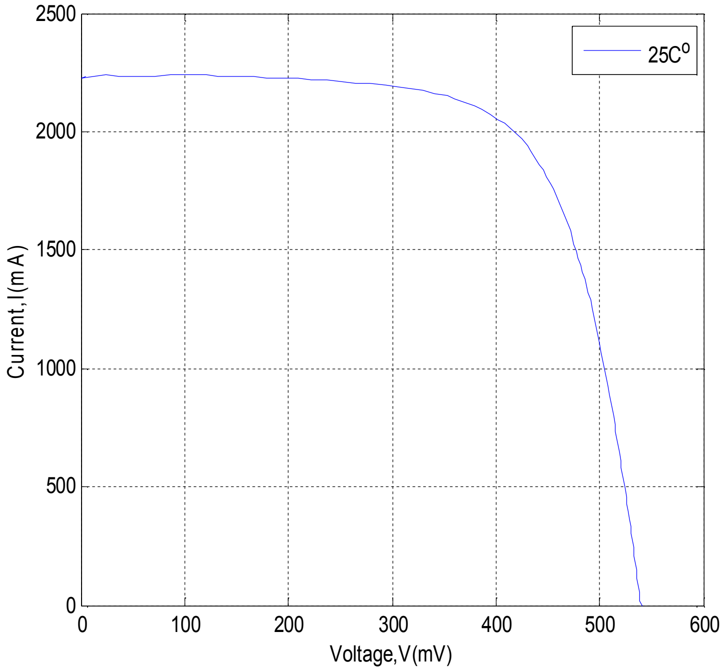


Fig. 3. The I-V carve of solar cell

6. Temperature effect on solar cell performance

During the measurement, the temperature is changed using a temperature controller to get different values of the temperature in order to see the effect on the performance of the solar cell. The table (2) presents the performance parameters for different values of temperature.

Table (2): temperature effect on the solar cell

Temperature [°C]	25	30	35	40
Isc,meas [mA]	2231.940	2248.905	2260.097	2276.202
$I_{sc,meas}$ [mA]	22.319	22.489	22.601	22.762
$J_{sc,meas}$ [mA/cm^2]	541.222	528.337	524.127	503.010
$V_{oc,meas}$ [mV]	2229.600	2251.440	2262.432	2277.774
$I_{sc,korr}$ [mA]	22.296	22.514	22.624	22.778
$J_{sc,korr}$ [mA/cm^2]	540.518	527.847	523.507	502.498
$V_{oc,korr}$ [mV]	1973.755	1974.984	1969.569	1948.262
I_{mpp} [mA]	19.738	19.750	19.696	19.483
J_{mpp} [mA/cm^2]	422.974	402.134	390.660	360.492
V_{mpp} [mV]	69.274	66.829	64.964	61.362
P_{mpp} [mW]: [mW]:	834.848	794.208	769.432	702.333
eta [%]:	8.348	7.942	7.694	7.023

Korr: calculation, Meas: measurement

The fig. 4 shows the temperature effect on the I-V carve of the cell. In addition, the fig. 5 illustrates s the temperature effect on Power-voltages carve.

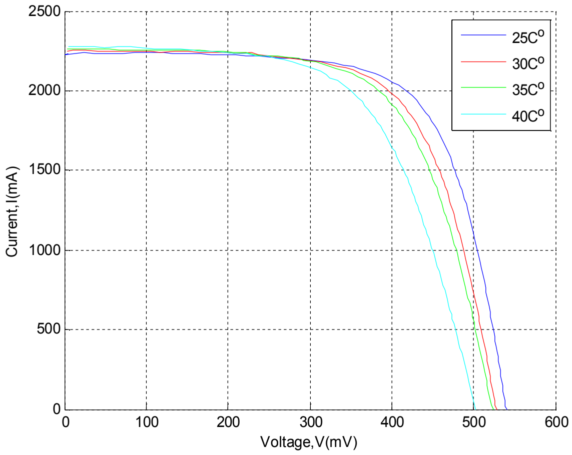


Fig. 4. Temperature effect on I-V carve of solar cell

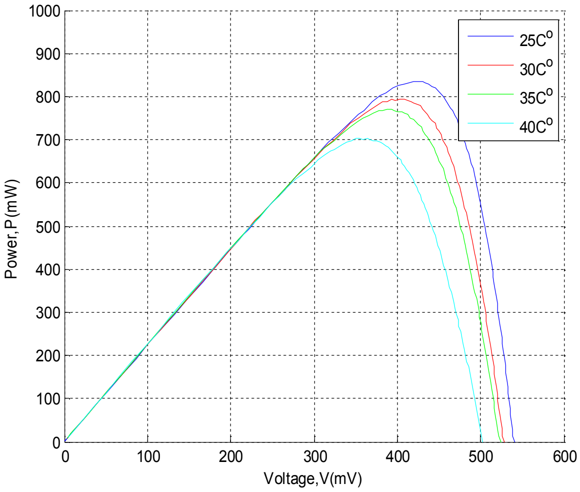


Fig. 5. Temperature effect on power carve of solar cell

7. Visual inspection

The module and cells are still in good condition as shown in Fig. 6. From visual inspection, it is clear that not only the cells seem to be unaffected but also no cells have been turned into a yellowish color [1].

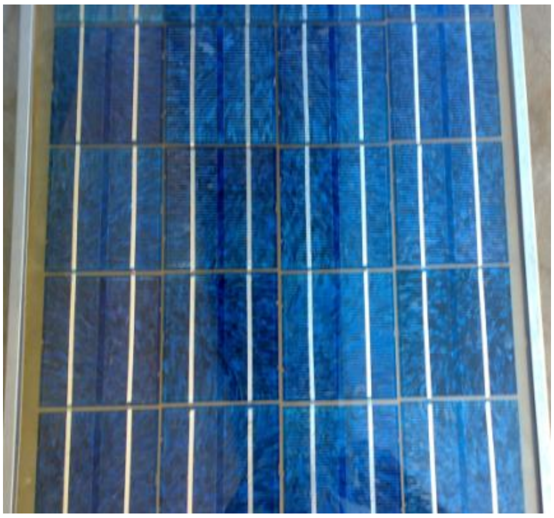


Fig. 6. Visual inspection of solar cell

8. Conclusion

In this paper, the I-V carve of solar cells which were removed from thirty year old PV module was measured under standard test condition.

In conclusion, the results show a peak power of 834.848 mW for thirty-year old solar cells with 8.348% efficiency. According to the temperature effect, the power degrades each time the temperature goes higher. Furthermore, the solar cells is still in good condition and no a yellowish color.

REFERENCE

1. M. A. Alshushan & I. M. Saleh, "Power Degradation and Performance Evaluation of PV Modules after 31 Years of Work" *IEEE*, 2013.
2. Y. Tang, B. Raghuraman, J. Kuitche, G. TamizhMani, C. E. Backus, and C. Osterwald, "An evaluation of 27+ years old photovoltaic modules operated in a hot-desert climatic condition" *IEEE,* vol. 2, pp. 2145 - 2147, 2006.
3. J. H. Wohlgemuth, "Long term photovoltaic module reliability" *NCPV and Solar Program Review Meeting*,
4. M. Saleh, I. Abouhdima, and M. B. Gantrari, "Performance of thirty years stand-alone photovoltaic system" *European Photovoltaic Solar Energy Conference*, pp. 3995 – 3998, 2009.

6

Develop a new method for detailed wheel and rail roughness measurements using replica material and Dektak profilometer

Develop a new method for detailed wheel and rail roughness measurements using replica material and Dektak profilometer

A. Shebani,
College of Computer Technology – Zawya
amerelshibani@yahoo.com

C. Pislaru
Institute of Railway Research,
University of Huddersfield, Huddersfield, UK

Abstract

Wheelsets are one of the most expensive components through the life of a rail vehicle. They require regular maintenance activities such as reprofiling on a wheel lathe, inspection for safety-critical damage to wheel and axle, and renewal of wheelset. There are several reasons for reprofiling such as tread wear, flange wear, and thermal; while, the cost of changing damaged rails is much greater than that of changing any other damaged part of track. The wheel and rail damage has been a concern in railway systems for several decades. The change of wheel profile and rail profile makes a large contribution to track maintenance cost. The develop a new method to measure the wheel/rail surface roughness can assist to improve the design of wheel and rail profiles, where the wheel/rail surface roughness is correlated to wheel/rail safety and economy. Therefore, the main aim of this work is to develop a new method for measuring wheel/rail roughness parameters using Dektak profilometer and replica material. The replica technique is very useful for situations in which it difficult to get to the surface in order to measure it, such as when the specimen is large; it is also useful when the components change due to wear and mechanical actions and the record of the original surface is needed. In this paper, the replica material which was applied to the wheel and rail surfaces of the twin disc test rig to make a copy of wheel and rail; then, the replica samples were measured using Dektak profilometer and the results were processed to establish wheel and rail roughness parameters.

Keywords: wheel and rail roughness, measurements, Dektak profilometer, replica material, twin disc rig.

1. Introduction to replica technique

The replica technique can be used to make a copy of the surface which needs to measure, and then; Dektak profilometer can be used to measure the surface roughness. The replica material has many types such as AccuTrans material; it has been developed for many applications such as wear and roughness measurement; and has many advantages such as: Fast and accurate even on rough surfaces, no hand mixing required, fast and easy to use, accurate dispensing, ideal for rough surfaces, flexible and accurate, horizontal and vertical planes, the setting time of regular AccuTrans is just four minutes at 20°C [1]. The replica technique has many advantages; but the most important advantage is that the replica offers a permanent history of surfaces; this record can be stored to use it for investigations at a later time. Another important advantage of the replica technique is that it can be used in places that are very difficult to access [2]. The replica technique is very useful for situations in which it difficult to get to the surface in order to measure it, such as when the specimen is large; it is also useful when the components change due to wear and the record of the original surface is needed [3]. Surface replication has been widely used in many applications such as examination and assessment of either surfaces difficult to be accessed by measurement tools or parts difficult to be dismantled for measurement. The fidelity and accuracy of the replication is one of the major concerns in actual applications [4]. The replica technique has several significant advantages, such as a permanent record of the surface being obtained, better resolution, hazardous environment is minimized, and scanning electron microscopy can be utilized [5]. The replication technique can be used to overcome many difficulties such as if there is a large wheel and we need to achieve measurements to the wheel surface, the replica technique making these measurements possible. The replication material is pressed onto the area of interest on the wheel. When the replica material is dried; it provides a surface that be identical to the surface of concern. The stylus and microscope techniques can be used for replicas measurements [6]. The replica method can be used in many applications such as hardness measurements, wear examinations, roughness measurements, and profile measurement [7].

2. DektakXT Stylus Profilometer

The measurement of various parameters of interest for a surface, including roughness, step heights or depths, by any metrology method necessarily provides only a representation of the surface details. The power of proper filtering for data analysis, according to recognized ISO standard methods cannot be underestimated when striving to provide the most accurate and reproducible results for a measuring system. Bruker has designed ISO compatibility to the two-dimensional (2D) profile ISO 4287 and 4288 standards into the versatile Vision 64TM software that powers the DektakXT Stylus Profiler. The Dektak XT (Fig 1) is a 2D contact profilometer used to provide quantitative information about step heights and surface roughness for thin and thick films measurements. This information is collected and analyzed in the Vision 64 application software. The advantages of Dektak is compatible with a wide range of materials, measures a wide range of vertical features with high resolution, and easy to use.

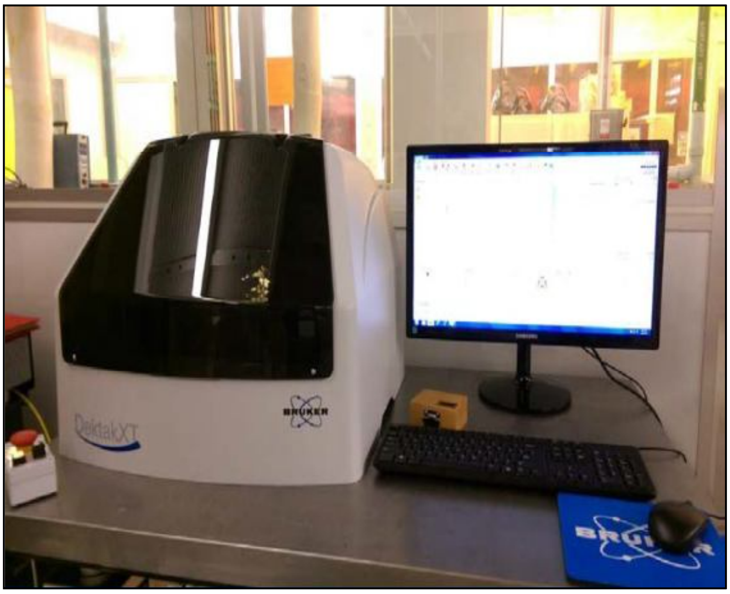


Fig. 1 DektakXT Stylus Profilometer

DektakXT Stylus Profilometer specifications:

Standard Operating Procedure: System Start up, adjusting the stage, taking a measurement, data analysis, and system shut down

System overview: Components (Hardware), and components (Software)

Factors: Scan parameters

Range: Vertical resolution of the scan: When measuring extremely fine geometries, the 6.5 um range provides a vertical bit resolution of 0.1 nm. For general applications, the 1.0 nm vertical resolution of the 65.5 um range is usually adequate. When measuring thick films or very rough or curved samples, select the 524 um range with 8.0 nm resolution.

Resolution: The horizontal resolution for the scan length and scan duration: the scan resolution is expressed in um/sample, indicating the horizontal distance between data points.

Speed: The scan speed in units of um/s. A slower scan generally indicates for more accurate results.

3. The twin disc rig test and replica material

The twin-disc system is simple and efficient; it consists of the use of two rollers pressed into contact, the variation of the relative velocity and of the contact pressure allows performing of the test under different conditions [8]. The twin disc approach possibly provides the best solution, and has been used extensively for wheel wear and rail testing materials [9]. The University of Huddersfield twin disc rig consists of an upper steel wheel of 310mm diameter, and a lower steel wheel with diameter of 290mm. The rollers and shafts are made of EN24T steel. Vertical force of up to 4KN can be applied on the rollers through a jacking mechanism. The rig consists of a rotary table to allow a relative yaw angle between the rollers; this yaw angle is indicated by markings on the handle of the rotary table. [10]. The twin disc rig for the University of Huddersfield (UOH) is shown in Fig (2).

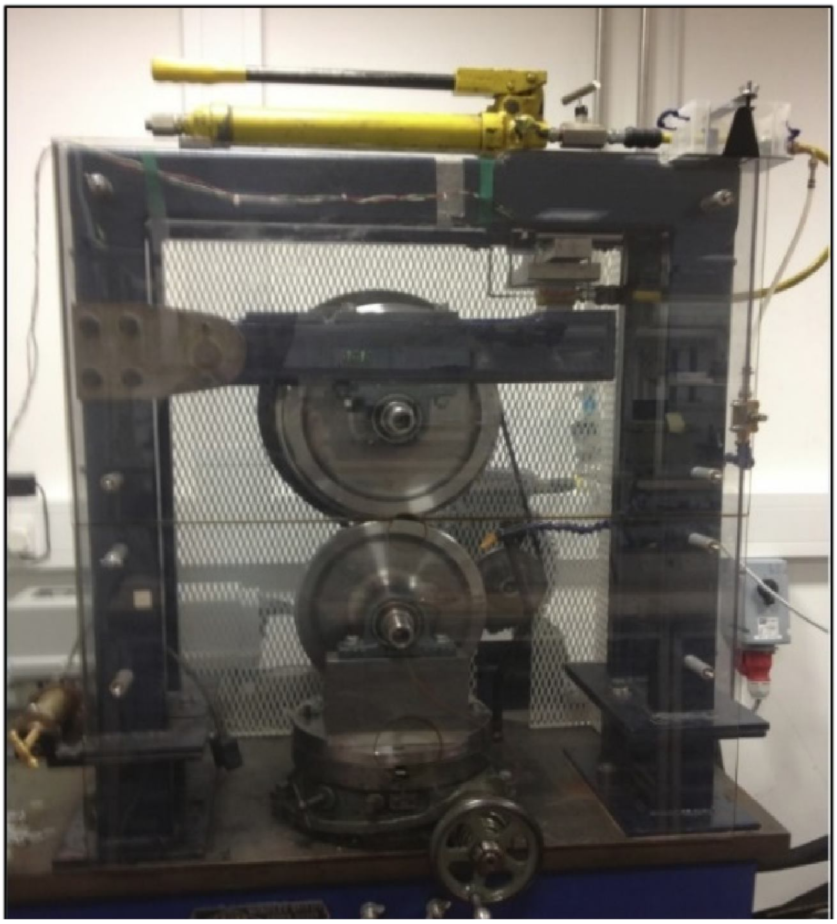


Fig 2 The twin disc rig UOH

In this work, the replica technique used for roughness measurements of the twin disc rollers, where the replica used to make a copy of the surfaces of two rollers before the test and after each test, and then the Dektak profilometer used to measure the rollers roughness. The name of the replica material which is used in this project is AccuTrans. The set of AccuTrans is shown in Fig (3).

Fig 3 The set of AccuTrans [11]

On this paper, the replica technique, twin disc rig test, and Dektak profilometer were developed for wheel and rail roughness measurements using the twin disc rig. Fig (4) shows a sample of replica material on the wheel and rail surfaces; and after the replica was taken off.

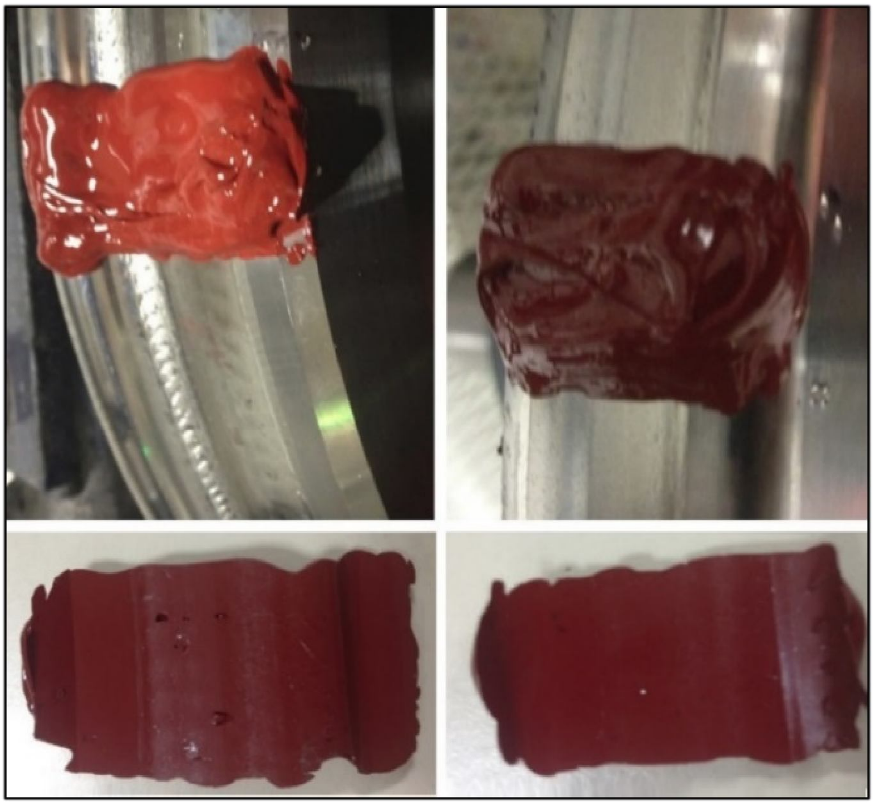


Fig 4 Sample of replica material on the wheel and rail surfaces; and after the replica was taken off

4. Surface roughness

The contact between two rough surfaces occurs at discrete contact points because of surface roughness such as shown in Fig (5). The real area is the sum of the areas of all the contact points [12].

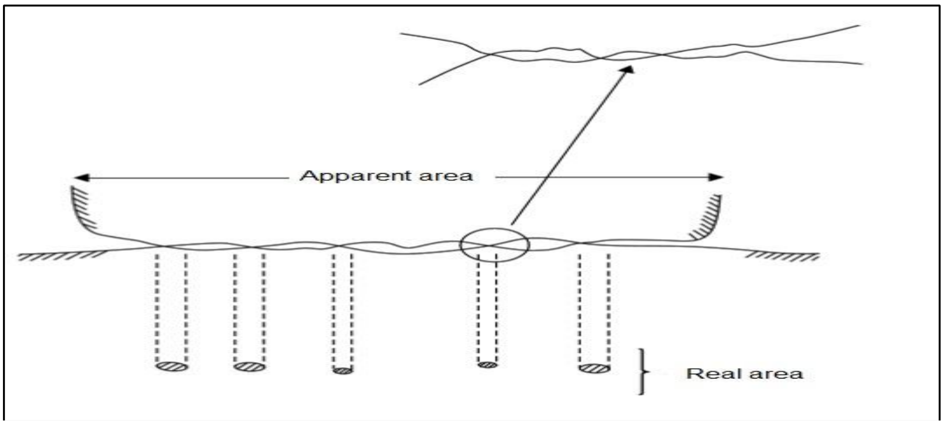


Fig 5 Apparent area and real area [12], [13]

Fig (6) shows the original profile, waviness profile, and roughness profile [14],[15].

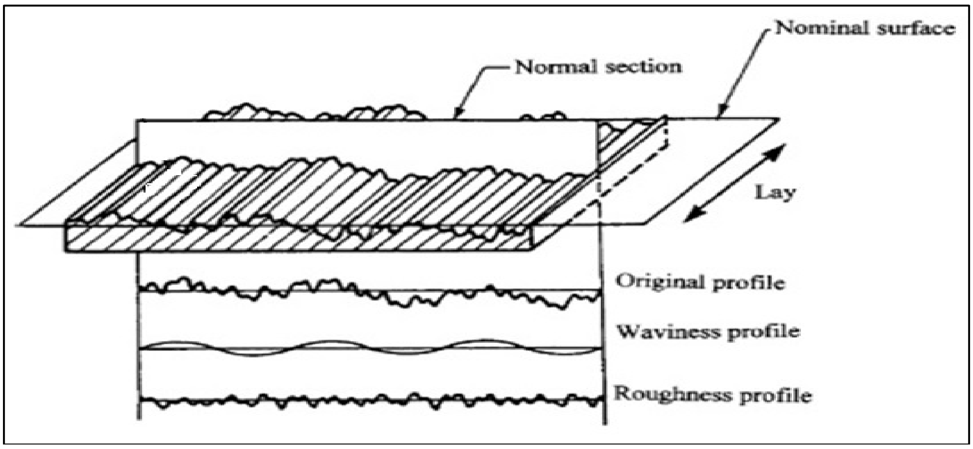


Fig 6 Original profile, waviness profile, and roughness profile [14], [15]

The basic definitions in roughness surface are [16], [17], [18]:

1. The profile can be defined as the line of crossing of a surface with a sectioning plane which is vertical to the surface.
2. Nominal surface is the intended surface.
3. Measured profile can be defined as the profile obtained with some measuring profilometers such as Talysurf profilometer.
4. Primary profile is the sum of all the deviations of the measured profile.
5. Waviness profile includes medium wavelength deviations of the measured profile.
6. Roughness profile includes only the shortest wavelength deviations of the measured profile.

The steps used to extract the roughness profile are shown in Fig (7), where low pass filter was used to extract the primary profile from the measured profile, high pass filter was used to extract the roughness profile from the primary profile, and low pass filter was used to extract the waviness profile from the primary profile. Therefore, the function of the filter is to separate the roughness profile and the waviness profile from the primary profile [15], [19].

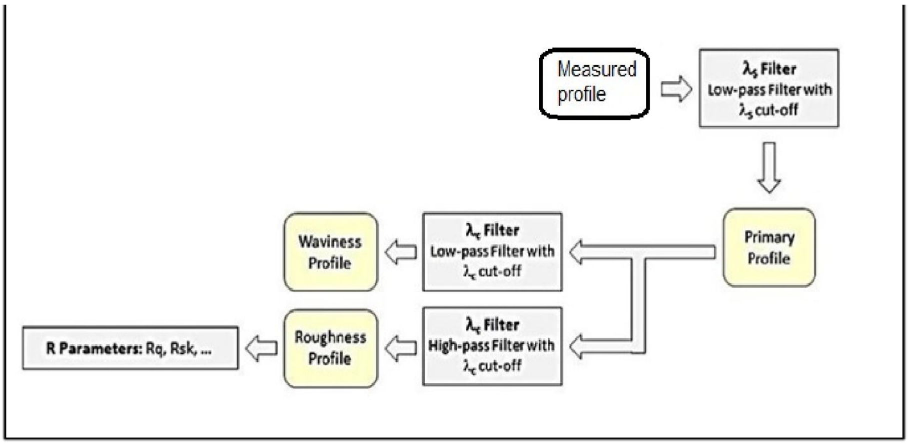

Measured profile
λs Filter Low-pass Filter with λs cut-off
Primary Profile
λc Filter Low-pass Filter with λc cut-off
Waviness Profile
λc Filter High-pass Filter with λc cut-off
Roughness Profile
R Parameters: Rq, Rsk, ...

Fig 7 Extract the roughness profile and waviness profile [20]

The roughness parameters are:

1. The centre line average value (R_a) or arithmetic average roughness; it can be determined from deviations about the center line within the evaluation length such as in Fig (8).

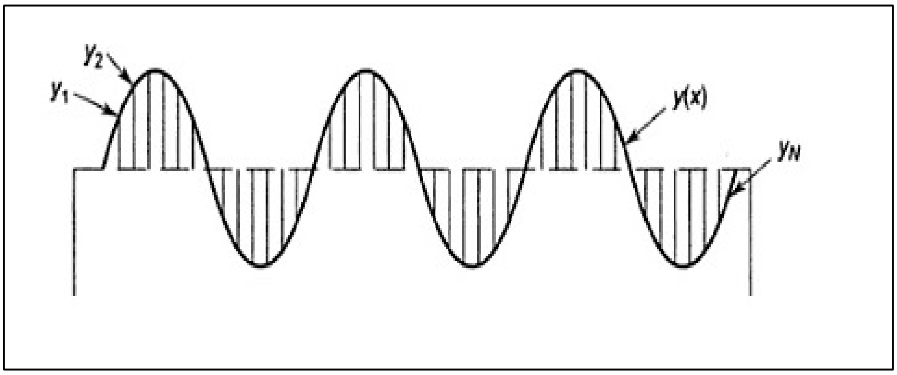


Fig 8 Surface roughness parameters [21]

The arithmetic average roughness can be calculated by the following equation [22]**:**

$$R_a = \frac{1}{N} \sum_{i=1}^{N} |y_i| \qquad (1)$$

Where N is the total number of points and y_i is the surface profile height to the center line average.

2. The maximum deviation of a peak above the centre line (R_p); this is the value of the highest peak measured above the centerline. It is the maximum data point height above the mean line through the entire data set. It can be calculated by using the following equation [23], [24].

$$R_p = \max y(x),\ 0 < x < L \qquad (2)$$

3. The maximum depth of valley below the centreline (R_v); this is the value of the lowest valley measured below the centre line. It is the

maximum data point depth below the mean line through the entire data set, and it can be calculated by using the following equation [23], [24], [25],[26].

$$R_v = |\ miny(x)\ |\ ,\ \ 0 < x < L \qquad (3)$$

4. Maximum vertical distance (R_t), it the maximum peak to valley height of the filtered profile. It can be calculated by using the following equation [23], [24], [25], [26],[27] .

$$R_t = R_p + R_v \qquad (4)$$

5. Investigate the arithmetic average roughness ($\mathbf{R_a}$) for wheel and rail using twin disc rig and Dektak profilometer.

Table (1) shows the values of arithmetic average roughness for wheel and rail after applying different values of load on rollers of the twin disc rig; where, the arithmetic average roughness was measured using Dektak profilometer.

Table 1 Arithmetic average roughness and load

Test No	Load (N)	R_a for wheel (μm)	R_a for rail (μm)
1	1000	3.53μm	2.57μm
2	1400	3.69μm	3.02μm
3	1800	3.96μm	3.24μm
4	2200	4.11μm	3.55μm
5	2600	4.19μm	3.57μm
6	3000	4.23μm	4.11μm
7	3400	4.26μm	6.13μm

Table (2) shows the values of arithmetic average roughness for wheel and rail after applying different values of yaw angle; where, the arithmetic average roughness was measured using Dektak profilometer.

Table 2 Arithmetic average roughness and yaw angle

Test No	**Yaw angle**	$\mathbf{R_a}$ **for wheel** (µm)	$\mathbf{R_a}$ **for rail** (µm)
1		2.64µm	2.85µm
2		3.15µm	3.08µm
3		3.20µm	3.45µm
4		3.29µm	3.60µm
5		3.44µm	3.95µm
6		3.94µm	5.89µm
7		4.90 µm	6.70µm

6. Discussion

A new method was developed for measuring wheel and rail surface roughness for the twin disc test rig. A replica material was used to make a copy of the surfaces of the two rollers before and after each test, and then, the Dektak profilometer used to measure the wheel and rail surface roughness parametrs. The twin disc test rig experiments were carried out to investigate the effect of key parameters such as load, and yaw angle on wheel/rail surface roughness parametrs for the twin disc test rig. Table (1) shows the values of arithmetic average roughness (R_a) for the wheel and rail after applying different values of load on rollers of the twin disc rig. For wheel surface, the R_a was equal $3.53\mu m$ at load of 1000N, and it increased to $4.26\mu m$ after applied load of 3400N. For rail surface, the R_a was equal $2.57\mu m$ at load of 1000N, and it increased to $6.13\mu m$ after applied load 0f 3400N. These results indicate that the load influence on wheel/rail roughness. These results show that the values of arithmetic average roughness were affected by increasing of load. Table (2) shows the values of arithmetic average roughness (R_a) for the wheel and rail after applying different values of yaw angle on rollers of the twin disc rig. For wheel surface, the R_a was equal $2.64\mu m$ at yaw angle of 0.1^0, and it increased to $4.90\mu m$ at yaw angle of 0.7^0. For rail surface, the R_a was equal $2.85\mu m$ at yaw angle of 0.1^0, and it increased to $6.70\mu m$ at yaw angle of 0.7^0. Therefore, there are a significant influence of yaw angle on wheel and rail surface roughness.

7. Conclusion

The University of Huddersfield twin disc test rig together with a replica technique and Dektak profilometer were developed for wheel/rail surface roughness parametrs measurements. The arithmetic average roughness (R_a) was measured for wheel and rail surfaces after each test using Dektak profilometer. Tests results show that the roughness parameter influenced by changing of load and yaw angle. The wheel and rail surface roughness can be measured using Dektak profilometer. It can have concluded that the replica material and Dektak profilometer are effective tools for the wheel/rail surface roughness parametrs measurements. The advantage of use replica material, that it is a permanent record to the wheel and rail surface roughness measurements.

References

1. Alicona, "Alicona Profilometer," Alicona UK Ltd, UK2015.
2. G. Lütjering and J. C. Williams, *Titanium* vol. 2: Springer, 2003.
3. D. J. Whitehouse, *Handbook of surface and nanometrology*: CRC press, 2010.
4. C. Y. L. Y. C. Liu, A. A. Malcolm, Z. G. Dong, "Accuracy of replication for non-destructive surface finish measurement," *Singapore International NDT Conference & Exhibition,* 2011.

5. A. Marder, "Replication microscopy techniques for NDE," *ASM Handbook.,* vol. 17, pp. 52-56, 1989.
6. W. B. Rowe, B. Dimitrov, and H. Ohmori, *Tribology of abrasive machining processes*: William Andrew, 2012.
7. K. G. Boving, *NDE handbook: Non-destructive examination methods for condition monitoring*: Elsevier, 2014.
8. N. Bosso and N. Zampieri, "Experimental and numerical simulation of wheel-rail adhesion and wear using a scaled roller rig and a real-time contact code," *Shock and Vibration,* vol. 2014, 2014.
9. E. Gallardo-Hernandez and R. Lewis, "Twin disc assessment of wheel/rail adhesion," *Wear,* vol. 265, pp. 1309-1316, 2008.
10. S. S. Hsu, Z. Huang, S. D. Iwnicki, D. J. Thompson, C. J. Jones, G. Xie*, et al.*, "Experimental and theoretical investigation of railway wheel squeal," *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit,* vol. 221, pp. 59-73, 2007.
11. Alicona, "AccuTrans ", http://www.forensicmag.com/articles/2014/12/best-forensic-products-2014.
12. B. Bhushan, *Modern Tribology Handbook, Two Volume Set*: Crc Press, 2000.
13. B. Bhushan, *Principles and applications of tribology*: John Wiley & Sons, 2013.
14. B. Muralikrishnan and J. Raja, *Computational surface and roundness metrology*: Springer Science & Business Media, 2008.
15. J. Raja, B. Muralikrishnan, and S. Fu, "Recent advances in separation of roughness, waviness and form," *Precision Engineering,* vol. 26, pp. 222-235, 2002.
16. U. Khandey, "Optimization of surface roughness, material removal rate and cutting tool flank wear in turning using extended taguchi approach," 2009.
17. Z. Dimkovski, "Characterization of a Cylinder Liner Surface by Roughness Parameters Analysis," *Blekinge Institute of Technology, Karlskrona, Sweden,* 2006.

18. I. Standards, "Surface Metrology Guide - Surfaces and Profiles," http://www.htskorea.com/tech/spm/profile.pdf 2015.
19. A. Boryczko, "Distribution of roughness and waviness components of turned surface profiles," *Metrology and measurement systems,* vol. 17, pp. 611-620, 2010.
20. E. Mainsah, J. A. Greenwood, and D. Chetwynd, *Metrology and properties of engineering surfaces*: Springer Science & Business Media, 2001.
21. Rao, *Manufacturing Technology,* India, McGraw-Hill publishing, 2009
22. R. Chattopadhyay, *Surface wear: analysis, treatment, and prevention*: ASM international, 2001.
23. H. Hocheng, *Machining technology for composite materials: Principles and practice*: Elsevier, 2011.
24. V. Bellitto, "Atomic Force Microscopy-Imaging, Measuring and Manipulating Surfaces at the Atomic Scale," *Published online: Intech,* 2012.
25. S. Srirattayawong, "CFD study of surface roughness effects on the thermo-elastohydrodynamic lubrication line contact problem," Department of Engineering, 2014.
26. D. A. Stephenson and J. S. Agapiou, *Metal cutting theory and practice* vol. 68: CRC press, 2005.
27. S. Tavares, "Analysis of surface roughness and models of mechanical contacts," University of Porto, Italy, 2005.

7

Implementation of Substitution Cipher on Field Programmable Gate Arrays

Implementation of Substitution Cipher on Field Programmable Gate Arrays

Ali F. Kaeib
University of Sabratah
Alikaeib@gmail.com

Osama A. S. Abourodes
Engineering Technology College
Zawia.
Abourodes.osa@gmail.com

Abstract

This paper presents algorithms implementation of substitution cipher on a FPGA “Field Programming Gate Arrays”, FPGA provides faster data rate, more flexibility to make changes to the programme and better physical security than other hardware.

The design was coded, simulated and tested by MATLAB /Simulink and Xilinx system generator and implemented on Xilinx Spartan 3A DSP XC3SD3400A -4CSG84C hardware implanted and test by using Xilinx ISE 12.4.

Index Terms Decryption, Encryption, FPGA, MATLAB, Substitution Cipher, Xilinx.

1. INTRODUCTION

Communications in the past were dependent mainly on traditional letters, payments were mainly made in cash or by checks, and confidential documents were sealed and stored in boxes. Traditional paper systems have been subject to development for a long time in parallel with suitable regulations that maintain their security and reliability.

Nowadays, all this has been changed dramatically. Large numbers of people have switched to new and modern methods of communications such as emails; this is due to their speed and cost efficiency in the communication process. “More people prefer to make their payments in electronically via the internet” [1].

Such trends save time and effort for users; however, they tend to cause security risks to their users at the same time. Therefore, a secure infrastructure is essential to deal with changes in the rapid development of technology of electronic communication systems [2].

The cryptography technology is as old as written language itself. The term “cryptography” originated from Greek roots, meaning "hidden word"; it is used in describing the earliest science of secret communications.

Until recently, such technology was mostly used by governments to protect diplomatic and military communications. Today, cryptography plays an important role in ensuring information and communications systems are secure [3].

Cryptography is a mathematical tool which security engineers use to protect data from manipulation or illegal access. Cryptography provides security personnel with the necessary utility to cover data, control access to it, authenticate its integrity, and estimate the time and cost of breaching security. Like any other services, security comes with a cost; hence it requires time, money and effort to implement cryptography tasks.

Key (asymmetric) and private (symmetric) key protocols are the most popular types of cryptographic protocols. Both communication partners use a common key in private key protocols, the encryption and decryption purposes both using this same key. Among them are the AES “Advanced Encryption Standard”, IDEA “International Data Encryption Algorithm”, and DES “Data Encryption Standard”.

The systems mentioned provide a high speed, which have a big disadvantage; this being that, for each participant, a common key has to be created. Public key protocols contain two keys; the first is left confidential (private key) and is used for either decryption (confidentiality) or encryption (signature) of messages. The reverse operation done by the other key, (public key), is published.

RSA (which stands for Rivest, Shamir and Adleman, who were the first to describe it publicly), Elliptic curve cryptography (ECC) are examples of public key and ElGamal, DSS (Digital Signature Standard) systems. Although these systems are not as fast as symmetric systems, they offer very high security levels and do not require the presence of the initial private key exchange. The two are used in real life applications. The algorithm of the public key establishes first a common private key over an insecure channel. Then the system of symmetric formed can be used for a more secure communication with high throughput [2].

2. Introduction to Cryptography

Encryption is a technique used to transform data, referred to as clear-text or plaintext, into a structure which appears unreadable and unsystematic, with this form known by the name of cipher-text. Plaintext comes in an understandable form for either a computer (executable code) or a person (a document). As soon as it is not changed into cipher-text, neither the machine nor the person can correctly process it unless it is decrypted. This provides the ability to transmit classified information through insecure channels with no unauthorized exposure.

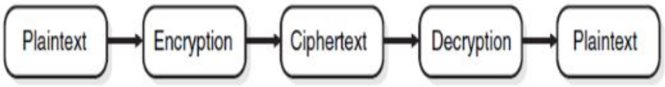


Figure 1: Encryption and Decryption

Fig 1 illustrates the process of transforming plaintext into cipher-text by encryption and vice versa, (cipher-text into plaintext) by the decryption process.

A cryptosystem is a system that can provide decryption; it can be created either by program codes or via hardware. Encryption algorithms are used in a cryptosystem and control the complexity of the process. The majority of algorithms come as complex mathematical formulas which are functional in a particular cycle and applied to the plaintext. In most encryption methods a key is used which is a secret value (commonly a long series of bits), which operates alongside the algorithm in encrypting and decrypting the text. Confidentiality, authenticity and integrity services can be provided by cryptosystems. They do not provide the availableness of systems or data. Confidentiality denotes that only authorised parties have access to information. Authenticity refers to certifying the message's source to maintain proper identification of the sender. Integrity ensures that no modification has been done to the message at any point of transmission, deliberately or by accident. [5].

Symmetric and Asymmetric Cryptography (asymmetric) Key and private (symmetric) key protocols are the most popular types of cryptographic protocols. Both communication partners use a common key in private key protocols, this key being used for both decryption and encryption purposes. Among them are DES “Data Encryption Standard” and IDEA “International Data Encryption Algorithm”.

3. Symmetric Cryptography

In a cryptosystem that employs symmetric cryptography, the same key of encryption will be used for decryption and encryption by both parties, as shown in Figure 2. This provides a functionality mode which is dual. Symmetric keys can also be referred to as secret keys because this type of encryption depends on both users keeping the key a secret and also protected. Once an intruder lays hands on this key, he will have the ability to decrypt all messages that have been encrypted with this key which he intercepted. Any two users who intend to use symmetric key encryption for exchanging data must have a restricted set of keys for only both of them. For example if persons (A) and (B) want to communicate using a symmetric encryption, both of them must obtain of a copy of the similar keys. Also assuming person A wants to communicate with another person (person C) using symmetric, a second key which is different from the first key has to be obtained.

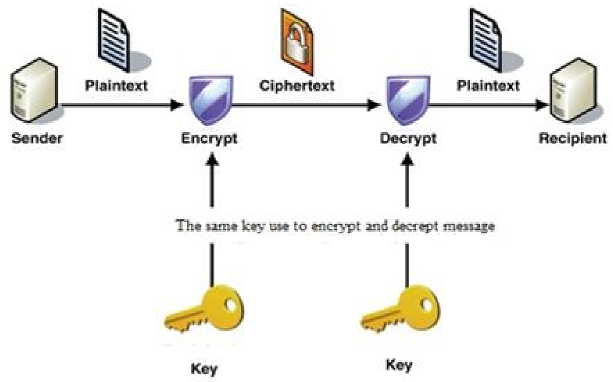


Figure خطأ! لا يوجد نص من النمط المعين في المستند.2: Symmetric cryptography system

This may not seem a complicated matter until person (A) needs to communicate for a longer period with a larger number of people: the task of keeping track of each correspondent and linking him with the correct key will become a complicated task [1].

The symmetric method's security depends completely on the user's ability in protecting the key. If a key is compromised, the intruder can easily decrypt and read all messages that have been encrypted by this key.

Symmetric systems can only provide confidentiality; they cannot provide authentication. If two people are using the same key, it is not possible to prove which one of them sent the message [5].

One may ask, are symmetric cryptosystems used if they have all these disadvantages?

It can be argued that symmetric algorithms are much faster than asymmetric algorithms. Large amounts of data can be encrypted and decrypted in a short time, while if an asymmetric algorithm is used to encrypt or decrypt the same amount of data, it will take an unacceptable amount of time. Additionally, uncovering data encrypted by a large key size in a symmetric algorithm is a very difficult task. [1]. The strengths and weaknesses of symmetric key systems can be summarized as follows:

Strengths:

1. Symmetric systems are faster than asymmetric systems
2. They are usually difficult to work out when using a large key size

Weaknesses:

1. Key distribution: in order to properly deliver keys, it requires a secure mechanism.
2. Scalability: a unique pair of keys is required by each pair of users, which allows for an exponential growth in the number of keys.
3. Limited security: confidentiality is provided, but not authenticity or non-repudiation.

The following are some examples of symmetric key cryptography algorithms:

- IDEA “International Data Encryption Algorithm”
- RC4, RC5 and RC6
- Blowfish
- DES “Data Encryption Standard”
- 3 DES “Triple DES”

4. Asymmetric Cryptography

One single key is used between entities in symmetric key cryptography, while a different key is used for each entity in public key systems: these are called asymmetric keys. Both keys are related mathematically. Assuming one key is used in a message to encrypt it, the other key is necessary for decrypting. In public key systems a pair of keys consists of a private key and a public key. Only the owner can know the private key, whereas the public key can be known by everybody.

In many cases the public keys can be listed in email databases and directories in order to be available for anyone who wishes to use them for encrypting or decrypting messages while communicating with another person. Figure 3 gives a clear illustration of an asymmetric cryptosystem.

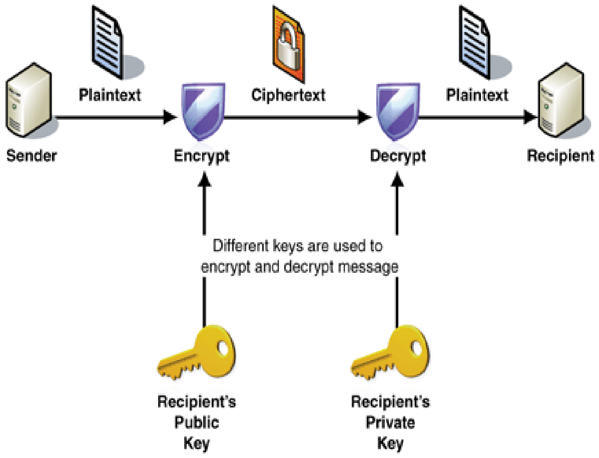


Figure 3 Asymmetric cryptosystem

In the figure above two people are involved in the asymmetric cryptosystem (person A is female and person B is male). In such a system both keys are mathematically related, but neither of them can be derived from the other. This means that once an intruder gains a copy of person B's public key he cannot know B's private key to read messages encrypted by using B' private key: the intruder can only decrypt this message if he has a copy of the private key [5].

The receiver can decrypt B's message with an intention to reply to B in a form which is encrypted. All that needs to be done is for her to encrypt the reply with B's public key, to which B can decrypt the message with his private key. In the case where asymmetric key card encryption technology is used, encrypting and decrypting are not possible when the exact same key is used. A message can be encrypted by B using his private key while the receiver can then use B’s public key to decrypt it. In order for the receiver to be sure that the message was sent from B, the message has to be decrypted with B’s public key. Encrypting a message with the corresponding private key is the only way it can be decrypted on the other end with a public key. This also helps in providing authentication, the reason being that B is the only person who should have his private key. Encrypting her response with B’s public key is probably the only way in which she can guarantee that B would be the only person reading the reply from her. B would be the only person who can decrypt the message because the necessary private key which is required to decrypt the message is known only by him. On the other hand, the receiver using her own private key can encrypt her response instead of using B’s public key. What is the point of her doing this? This makes it known to B that the message being sent was from her. Encrypting her response using B’s public key does not guarantee the authenticity, because anyone could have gotten hold of B’s public key.

Using her private key to encrypt the message allows for B to be sure that no one else but her sent the message. There is no authenticity produced by symmetric keys because at both ends, the same key is normally used. In the case where one of the secret keys is used, this does not specify that the message came from a specific entity. Encrypting the file with the receiver’s public key is the only means to guarantee confidentiality if this is the most important factor for the sender [1].

When a message can only be decrypted by the person who has the matching private key, this is often called a secure message format. An open message format is when the message is encrypted with the sender’s private key for which confidentiality is not ensured.

Encrypting a message with a sender’s private key and also encrypting it once more with the receiver’s public key would guarantee a message to be secure and in a signed format. In this case, the receiver would have to use his own private key to decrypt the message and also use the sender's public key to decrypt it again. This ensures that the message delivered is confidential and authentic.

It should be noted here that the public key is not used solely for encryption purposes while the private key's role is data decryption. Both keys have the capability of either encrypting or decrypting data. After all, data encrypted with a private key cannot be decrypted with the same private key. Any data encrypted with a private key has to be in all cases decrypted with a public key corresponding to it.

Also vice versa, data encrypted with a public key requires a corresponding private key to decrypt the data. A cryptosystem which is asymmetric normally works slower than a system which is symmetric, but this provides for authentication, confidentiality and non-repudiation depending on how the configuration is used. Asymmetric systems do not have scalability problems that symmetric systems have and they provide easier and more manageable key distribution [5].

The following are the strengths and weaknesses of asymmetric key systems:

Strengths:

1. Its key distribution is better than the symmetric system.
2. The scalability is much better than the symmetric system.
3. There is provision of non-repudiation, authentication and confidentiality.

Weaknesses:

The asymmetric key systems are not as fast as the symmetric system. Systems listed below are some examples of asymmetric key algorithms:

1. El Gamma
2. Digital Signature Standard (DSS)
3. Diffie-Hellman
4. RSA

5. Elliptic Curve Cryptosystem (ECC)

5. FPGA

Field programmable gate arrays (FPGAs) are valuable tools and can help in a number of stages of design. A prototyping model, on the other hand can be developed using a FPGA module. Developing an Application-specific integrated circuit (ASIC) chip is quite expensive due to the fact that in order to change its structure, a completely new chip is required once a design is completed. Opportunities are given to FPGAs designers in order to test the complete hardware (limitations to timing are inherent) for possible bugs and problems. There are also large and inexpensive FPGAs which have been developed recently: the design of a complete system is now possible in a single chip (SoC, or a system on chip) [6]. The figure 4 below shows the structure of the FPGA which is a structure of logic elements (LEs), which through neighboring LEs can be linked. This link is possible through the programmable interconnect which the grid of rows and columns would form throughout the chip [8].

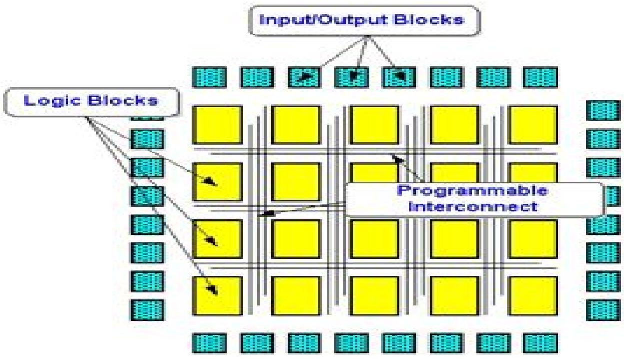


Figure 4: FPGA structure

In addition, the FPGA contains several embedded memory blocks collected in columns with each block ("MK4 block") holding 4 kbits. A number of columns of the M4K blocks might be available for each FPGA. A 4-bit look up table (LUT) is contained in each LE having a single output as shown in Fig 5, and this store 16 values which are programmable. This is similar to storing a 16 row truth table and can be used as a 4 bit input/1 bit output SRAM memory. Bypassing the flip flop (“FF”), the LUT output can give rise to an ‘unregistered’ value, or give a ‘registered’ value by entering the FF. This routing can be achieved by the multiplexers (MUXs) [7].

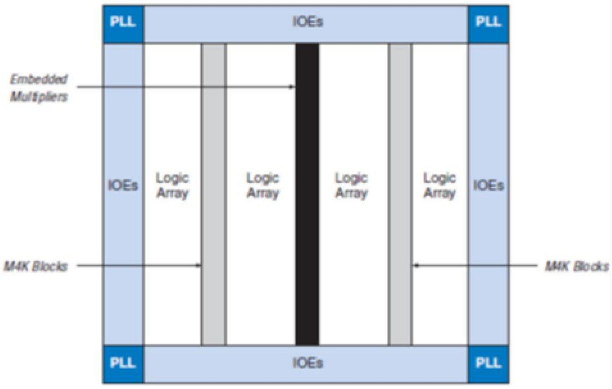


Figure 5: Embedded memory blocks on FPGA

6. Benefits of FPGA

1. Performance. FPGAs in attractive advantage of hardware parallel with exceed the computing power of DSPs “digital signal processors” by breaking the concept of sequential execution and accomplishes more per clock cycle.
2. Input and output (I/O) that are controlled at the hardware level, closely match the applications requirements by providing a faster response time and specialized functionality [8].
3. Time to market – The technology offered by FPGAs gives flexible and rapid prototyping capabilities when faced with increased time-to-market concerns. Without necessarily going through the custom ASIC designs, which is a long fabrication process, an idea or concept can be tested and also verified in hardware. This makes it possible to implement incremental

changes within hours on an FPGA design which normally would take weeks. Commercial off-the-shelf (COTS) hardware with different types of I/O is available which are already connected to a user programmable FPGA chip. The decrease in the learning curve is as a result of the growing high level software availability tools that have layers of abstraction, and do often include valuable IP cores (pre-built functions) for the control of advanced signal processing [7].

4. Reliability – Program execution of the FPGA circuitry is said to be 'hard' implementing since software tools provide for the programming environment. Processor based systems in order to share resources and schedule tasks frequently involve several layers of concept among multiple processes. The operating system manages the processors bandwidth and memory, while the hardware resource is controlled by the driver layer. At any particular time, only one instruction can be executed for any given processor core. The processor based systems by pre-empting one another are at risk of time critical tasks continually. There is minimal concern on reliability of FPGAs that do not use operating systems causing true parallel execution and deterministic hardware dedicated to every task [6].
5. Long-term maintenance – FPGA chips which are field upgradeable as mentioned earlier, compared to ASIC redesigns, are not time consuming nor expensive. For example, digital communication protocols can change overtime due to some specifications, causing maintenance and forward compatibility problems to an ASIC based interface. FPGA chips, being reconfigurable, can keep up with necessary future modifications that require to be performed. Functional enhancements can be made as a system or product matures, without time being spent in redesigning the hardware or the board layout being modified [7].

7. Substitution Cipher

This chapter explains substitution cipher and implementation on FPGA. Substitution cipher is a very old and simple cipher, and was used by Julius Caesar; the so-called “Caesar Cipher”. The algorithm used a simple shift of letters by three letters in plain text message to produce unintelligible messages. To obtain the original message simply needing to put back each letter in the unreadable message “ coded” by the letter three places to the left as shown table below [2].

Original alphabet:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Substitution cipher alphabet:

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

If the characters of the plain text alphabet and cipher-text alphabet are numbered and denoted by i and j respectively, then in the above example, for all i = 1,...,26: j = i +3 (mod 26). Mod 26 implies that the left part and right part of the equation may only differ by a multiple of 26. In a more general form, j = i + t (mod 26), in which t represents the number of characters the two alphabets are shifted.

An important characteristic of the Caesar substitution is the fact that the order of the characters of the substitution alphabet remains unchanged. The total number of keys is no more than 26, so this cipher can very easily be cracked; once a single letter of the cipher text can be related to a letter of the plaintext, the system breaks down. If the message is sufficiently large, it is all the more straightforward to find such a relation; simply note the most frequently occurring letter and the chances are that this is equal to the letter of the original plain text [9].

8. Methodology

In this paper MATLAB Simulink (containing Xilinx blocks) was used for the implementation.

8.1 Xilinx ISE

The final step is to place and route the synthesized code with Xilinx ISE. ISE means Integrated Software Environment. It provides many tools to accomplish each step of the design process from design entry to downloading the design to the FPGA. A synthesizer (XST) is part of these tools. One of the other ISE's tools is Impact, used to download the bit generated file directly to the FPGA.

8.2 Hardware

The FPGA used in this project is a Spartan (3A DSP XC3SD3400A - 4CSG84C) as shown in Figure 5 below.

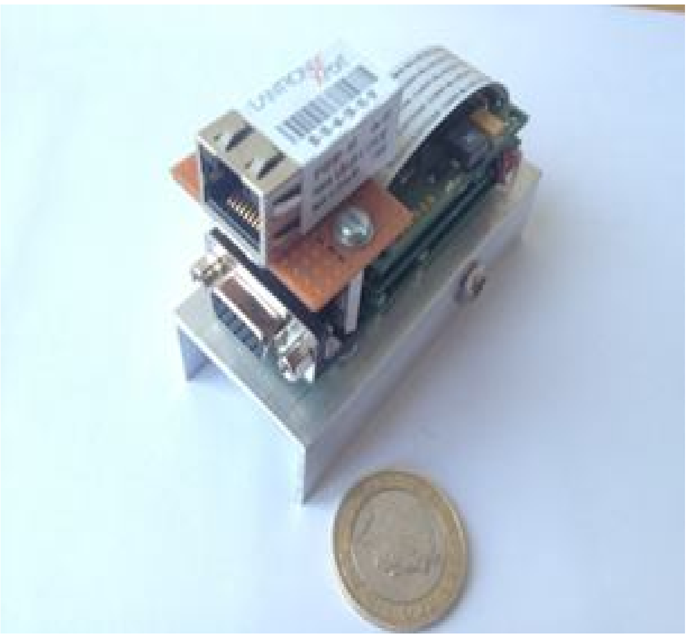


Figure 5: FPGA board

TABLE 1: Main characteristics of FPGA

Characteristic	Value
System Gates	3400.000
Logic Cells	53712
Slices	23842
Block RAM	2268 Kbit
DCM (Digital Clock Manager)	8

8.3 Design Flow

The next steps are summarized in the design flow chart below.

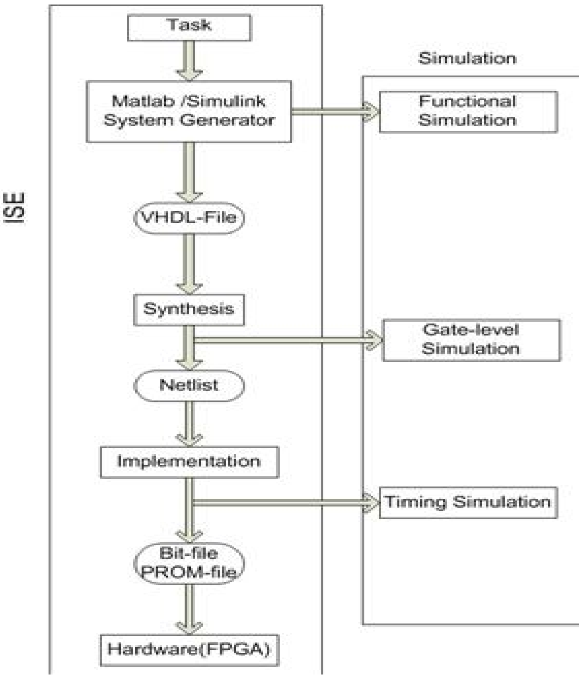


Figure 6: Design flow chart

9. Implementation of Substitution Cipher

In Substitution Cipher the design was coded and tested by MATLAB/Simulink and Xilinx system generator. For implementation on FPGA Xilinx ISE was used. MATLAB /Simulink provides a high-level view over the test environment using the system generator for (DSP) toolbox from Xilinx. In order to test the substitution cipher in Simulink text was converted to ASCII code because one cannot enter text as input in MATLAB Simulink at the end we get the result every single character shifted lift by three letters.

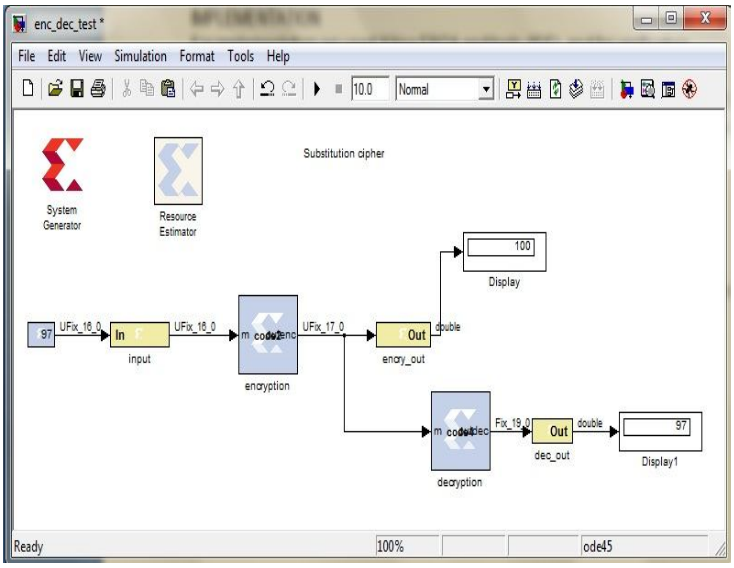
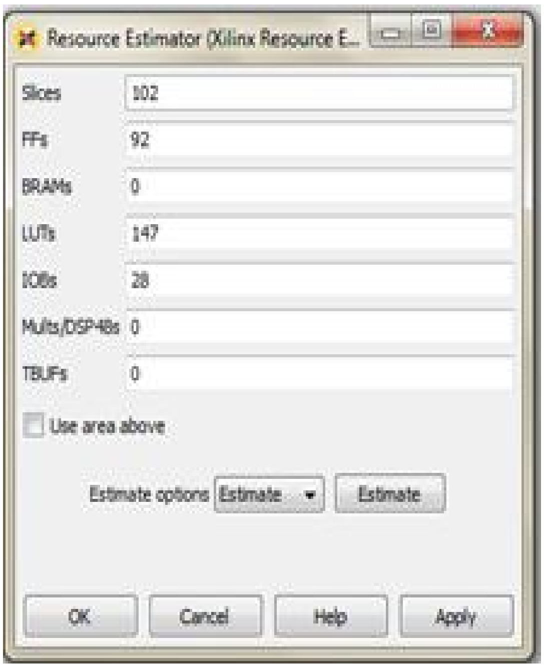


Figure 7 : MATLAB Simulink Substitution cipher

9.1 Resources Estimator

Xilinx tools provided resource estimator which is used to know how much hardware is needed for implementation, the number of hardware used depending on FPGA type.

The figure below shows us the requirement of hardware that is needed for implementing the design on FPGA.



Resource Estimator (Xilinx Resource E...

Slices	102
FFs	92
BRAMs	0
LUTs	147
IOBs	28
Mults/DSP48s	0
TBUFs	0

□ Use area above

Estimate options: Estimate

Figure 8: Resource Estimator

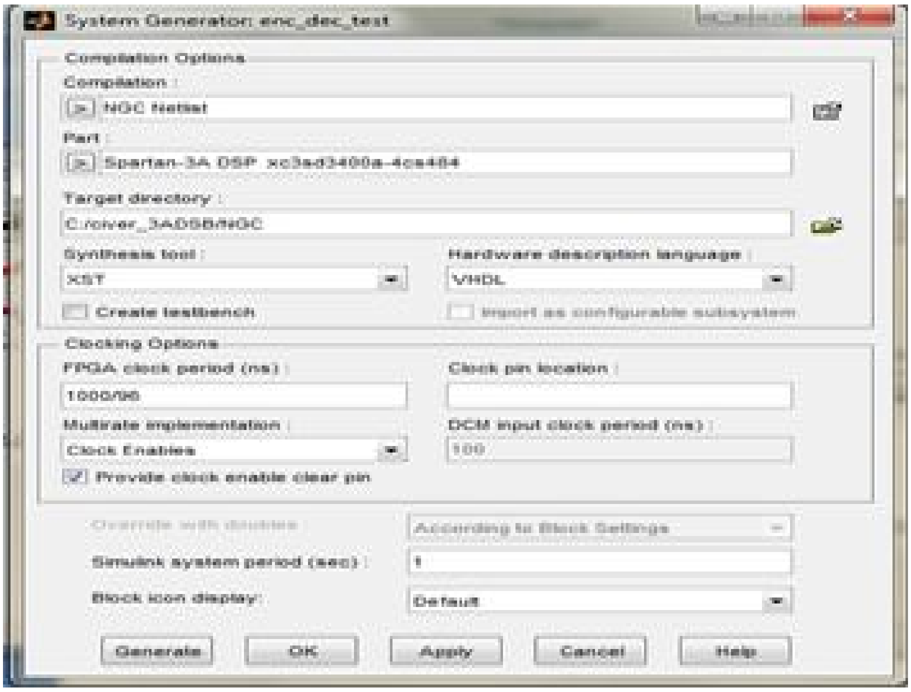
9.2 System Generator

The System Generator block provides control of the system and simulation parameters, and is used to invoke the code generator. The System Generator block is also referred to as the System Generator "token" because of its unique role in the design. Every Simulink model containing any element from the Xilinx Blockset must contain at least one System Generator block (token). "Once a System Generator block is added to a model, it is possible to specify how code generation and simulation should be handled" [10].

9.3 Xilinx ISE Process

After testing in MATLAB Simulink the system generator was run to generate NGCNitlist which converts MATLAB /Simulink design to HDL -"Hardware Description Language"- also known as synthesis presses.

Synthesis is one of the most essential steps in our design methodology because it takes the conceptual Hardware Description Language (HDL) design definition and generates the physical or logical symbol for the targeted FPGA. Synthesis is the optimisation process of adapting a logic design to the logic resources available on the chip. Then the schematic symbol shown in figure 9 was created. In this step one has to define the inputs and outputs.



System Generator: enc_dec_test

Compilation Options

Compilation : NGC Netlist

Part : Spartan-3A DSP xc3sd3400a-4cs484

Target directory : C:/oiver_3ADSB/NGC

Synthesis tool : XST

Hardware description language : VHDL

□ Create testbench

□ Import as configurable subsystem

Clocking Options

FPGA clock period (ns) : 1000/96

Clock pin location :

Multirate implementation : Clock Enables

DCM input clock period (ns) : 100

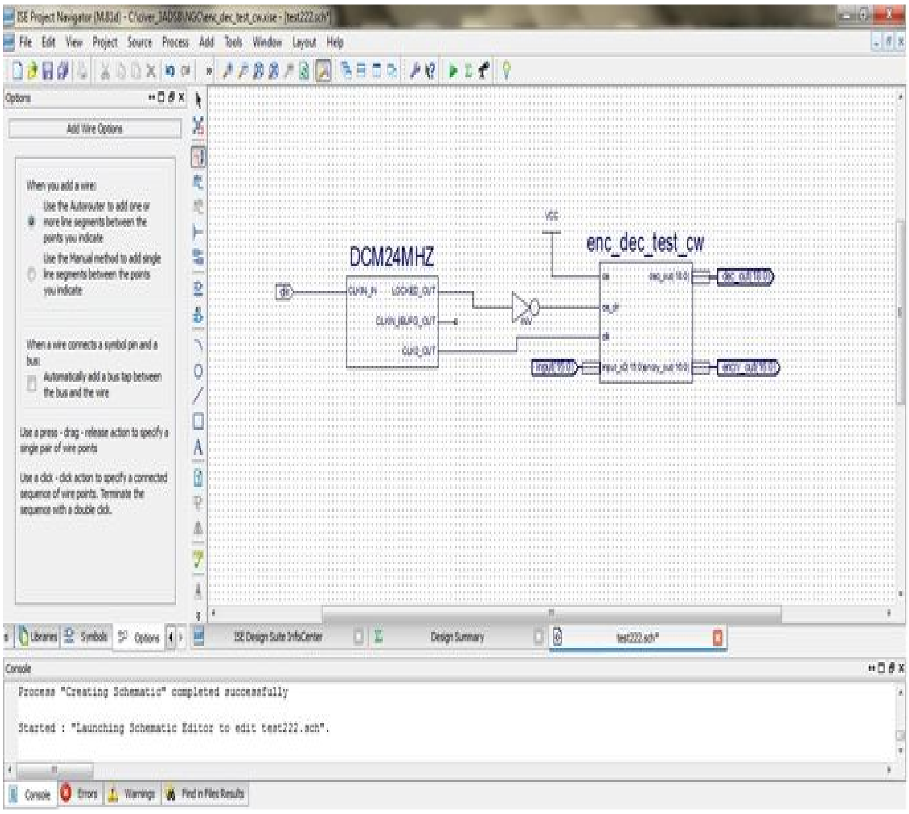
☑ Provide clock enable clear pin

Override with doubles : According to Block Settings

Simulink system period (sec) : 1

Block icon display: Default

Figure 9: System Generator (A)



When you add a wire:

● Use the Autorouter to add one or more line segments between the points you indicate

○ Use the Manual method to add single line segments between the points you indicate

When a wire connects a symbol pin and a bus:

□ Automatically add a bus tap between the bus and the wire

Use a press - drag - release action to specify a single pair of wire points

Use a click - click action to specify a connected sequence of wire points. Terminate the sequence with a double click.

```
Process "Creating Schematic" completed successfully

Started : "Launching Schematic Editor to edit test222.sch".
```

Figure 9: Schematic (B)

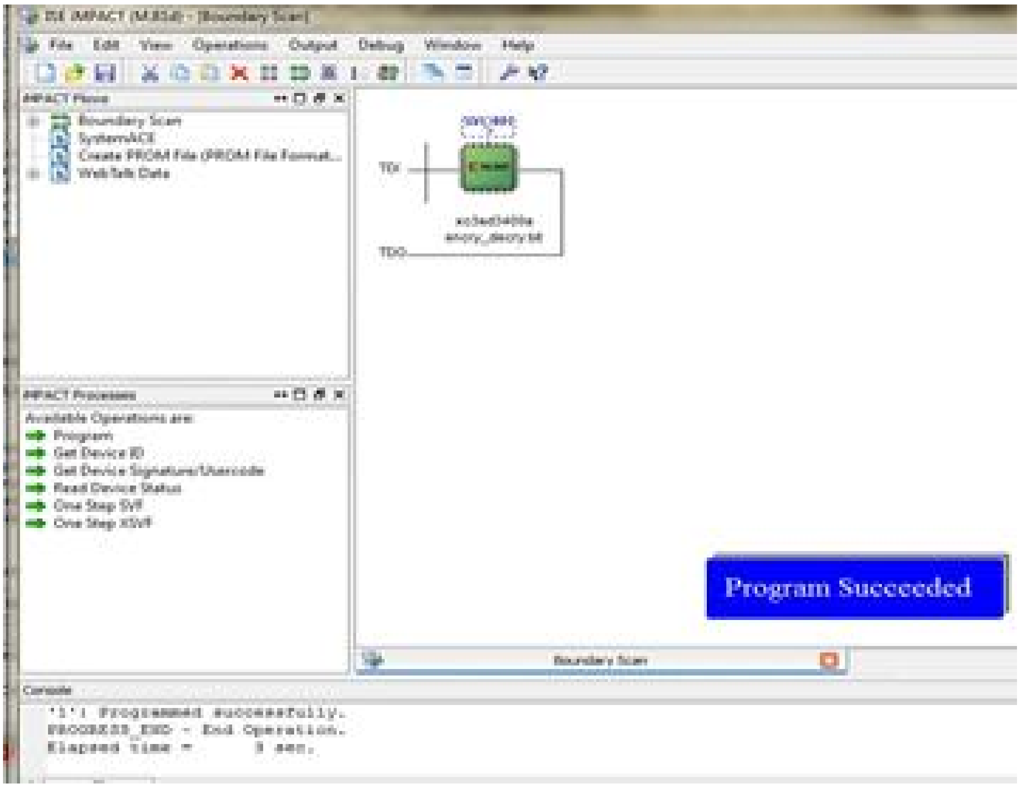
9.4 Configuration Process

This is the implementation step summarized in the steps below. These steps are the same in all implementation of FPGA:

1) Create the program file/bit file, conversion of our design in loadable bit file
2) A bitstream is generated from the physical place and route information and is transferred through cables to the target device
3) Under the start up options tab, the start-up clock can adjust in JTAG clock or CCLK.

The last step to configuration is Impact process:

1) Programming FPGA respective PROM
2) ISE IMPACT is a robust configuration tool that automatically manages everything from bitstream generation to the device download as shown in the figure below.



Program Succeeded

```
'1': Programmed successfully.
PROGRESS_END - End Operation.
Elapsed time =       3 sec.
```

Figure 10: Shows download program on FPGA succeeded (Impact process)

9.5 Results

This table below shows the results from ISE tools after implementation on FPGA.

Table 2 shows the results of implementation of the substitution cipher

enc_dec_test_cw Project Status			
Project File:	enc_dec_test_cw.xise	**Parser Errors:**	No Errors
Module Name:	encry_decry	**Implementation State:**	Programming File Generated
Target Device:	xc3sd3400a-4cs484	**Errors:**	
Product Version:	ISE 12.4	**Warnings:**	
Design Goal:	Balanced	**Routing Results:**	All Signals Completely Routed

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of4 input LUTs	**79**	**47,744**	**1%**	
Number of occupied Slices	**51**	**23,872**	**1%**	
Number of Slices containing only related logic	**51**	**51**	**100%**	
Number of Slices containing unrelated logic	**0**	**51**	**0%**	
Total Number of 4 input LUTs	**93**	**47,744**	**1%**	
Number used as logic	**79**			
Number used as a route-thru	**14**			
Number of bonded IOBs	**52**	**309**	**16%**	
Average Fanout of Non-Clock Nets	**2.24**			

The table above has been taken from ISE tools that show the type of FPGA used errors and warning in our design: no error found. The second part in the table shows the device utilization summary that means the number of hardware used in the design, and which are available in FPGA.

10. Conclusion

In this paper, we present an algorithm implementation of Substitution Cipheron on FPGAs "Field Programming Gate Arrays", because FPGA provides faster data rate, more flexibility to make changes to the program and better physical security than other hardware. Therefore the implemented design used a total number of 79 LUTs on a product version of Xilinx ISE 12.4.

References

1. Shokrollahi, J., Efficient Implementation of Elliptic Curve Cryptography 2006.
2. Buchmann, J. Introduction to cryptography. New York: Springer 2004.
3. Salomaa, A., Public Key Cryptography. Berlin: Springer 1996..
4. Schneir, B.,. Applied cryptography: protocols, algorithms, and source code in C. New York: Wiley 1996 .
5. Meyers, M. & Harris, S.. CISSP All-in-One Certification Exam2001
6. El-Maleh, A., n.d. Introduction to field Programmable Gate Array
7. National Instruments, 2012. Introduction to FPGA Technology
8. OptiMagic™, Inc, Frequently-Asked Questions (FAQ) About Programmable Logic, 2000.
9. Lubbe, J. C. A.. Basic method of cryptography. Cambridge: Cambridge University Press 1994, pp 62-84
10. Xilinx Inc. Design Tools, 2012 URL: http://www.xilinx.com/products/design-tools/ise-design-suite/index.htm

8

A Comparative Study of Lossless Data Compression Techniques

A Comparative Study of Lossless Data Compression Techniques

Elham Yakhlef Abushwashi
Technical electronic department
Technical faculty engineering, Zwara
Elham_abuelshwashi@yahoo.com

Hamida Aboulqasim Oushah
Software development department
College of Computer Technology, Zawia
e_hamida@yahoo.com

Abstract

Data compression is a process that reduces the data size, removing the excessive information and redundancy. It is a common and important requirement for most of the computerized applications, it can shorter the data size, which lead to cost reduction. The main purpose of data compression is to remove data redundancy from the store or transmitting data, it is also an important application in file storage field and distributed system. Data compression techniques are can be used in different data formats such as text, audio, video and image files. The aim of the study is to compare between many of the Lossless data compression techniques and compare their performance. There are many techniques of data compression and they can be categorized as Lossy and Lossless Compression methods. In this study a Run-length encoding, Huffman Coding, Shannon-Fano Coding, and LZW Encoding algorithm were used, their performance were compared by using data compression in the text format, the compression ratio, compression factor and saving percentage were calculated. Compression ratio in Huffman coding and Shannon-Fano coding where less than Run-length encoding and LZW encoding (38%, 40%, %81, 74%) respectively, while compression factor where higher than Run-length encoding and LZW encoding (2.63, 2.48, 1.23, 1.35) respectively, the results of saving percentage by using Huffman coding and Shannon-Fano coding where higher than Run-length encoding and LZW encoding (62%,60%, 19%, 26%) respectively. The study pointed to the effectiveness of Huffman and Shannon-Fano coding reducing the size of the files compare to other algorithms.

Key Keywords: Data compression, Lossless data compression technique, Huffman Coding, Run length encoding, Shannon-Fano coding, LZW encoding.

1. INTRODUCTION

Data compression is a way to reduce storage cost by eliminating redundancies that happen in most files. There are two types of compression, lossy and lossless. Lossy compression reduced file size by eliminating some unneeded data that won't be recognize by human after decoding, this often used by video and audio compression[1]. Lossless compression on the other hand, manipulates each bit of data inside file to minimize the size without losing any data after decoding. This is important because if file lost even a single bit after decoding, that mean the file is corrupted. Lossless data compression is a technique that allows the use of data compression algorithms to compress the text data and also allows the exact original data to be reconstructed from the compressed data. This is in contrary to the lossy data compression in which the exact original data cannot be reconstructed from the compressed data. The popular ZIP file format that is being used for the compression of data files is also an application of lossless data compression approach. Lossless compression is used when it is important that the original data and the decompressed data be identical. Lossless text data compression algorithms usually exploit statistical redundancy in such a way so as to represent the sender's data more concisely without any error or any sort of loss of important information contained within the text input data. Since most of the real-world data has statistical redundancy, therefore lossless data compression is possible. Lossless compression methods may be categorized according to the type of data they are designed to compress. Compression algorithms are basically used for the compression of text, images and sound. Most lossless compression programs use two different kinds of algorithms: one which generates a statistical model for the input data and another which maps the input data to bit strings using this model in such a way that frequently encountered data will produce shorter output than improbable(less frequent) data. The advantage of lossless methods over lossy methods is that Lossless compression results are in a closer representation of the original input data. The performance of algorithms can be compared using the parameters such as Compression Ratio and Saving Percentage. In a lossless data compression file the original message can be exactly decoded. Lossless data compression works by finding repeated patterns in a message and encoding those patterns in an efficient manner. For this reason, lossless data compression is also referred to as redundancy reduction[2]. Because redundancy reduction is dependent on patterns in the message, it does not work well on random messages. Lossless data compression is ideal for text.

2. COMPRESSION TECHNIQUES

There are two categories of compression techniques, lossy and lossless. Whilst each uses different techniques to compress files, both have the same aim: To look for duplicate data in the data.

2.1 Lossless Compression

Lossless data compression is a class of data compression algorithms that allows the original data to be perfectly reconstructed from the compressed data. In lossless data compression, the integrity of the data is preserved. Redundant data is removed in compression and added during decompression. Lossless compression methods are normally used in cases where it is important that the original and the decompressed data be identical [3].

2.2 Lossy Compression

In lossy data compression original data is not exactly restored after decompression and accuracy of reconstruction is traded with efficiency of compression. This type of compression used for image data compression. The decompression ratio is high compare to lossless data compression technique. Sometimes some loss of quality is acceptable. For example the human ear cannot hear all frequencies, people can't hear may end up with a smaller file, but it is not possible to get back to how exactly the original music sounded [2]. In such cases, we can use a lossy data compression methods. These methods are cheaper, they take less time and space when it comes to sending millions of bits per second for images and video.

3. LOSSLESS COMPRESSION TECHNIQUES

3.1 Run-length encoding(RLE)

Run Length Encoding (RLE) is the simplest of the data compression algorithms. It is created especially for data with strings of repeated symbols [3]. The consecutive sequences of symbols are identified as runs and the others are identified as non runs in this algorithm [4]. The general idea behind this algorithm is to replace consecutive repeating occurrences of a symbol by one occurrence of the symbol followed by the number of occurrences. The RLE algorithm uses those runs to compress the original source while keeping all the non-runs without using for the compression process [3].
For example, consider the following text string:

eelhhhaaammhhhaaammmiiddaaa = 27 characters
= 27 * 8 bits for each character
= 216 bits

Compressed string:

2l1h3a3m2h3a3m3i2d2a3 = 22 characters
= 22 * 8 bits for each character
= 176 bits

3.2 Huffman Coding

One of the most popular techniques for removing coding redundancy is due to Huffman [3]. Huffman coding was developed by Dr. David A. Huffman in 1952. Huffman coding assigns shorter codes to symbols that occur more frequently and longer codes to those that occur less frequently [5].

There are mainly two major parts in Huffman Coding

- Build a Huffman Tree from input characters.
- Traverse the Huffman Tree and assign codes to characters.

Huffman uses bottom-up approach, it is simple and can be described in terms of creating a Huffman code tree [3].
Example for Huffman coding:

Elhamabushwashihamidaoushah

Count of symbols stream as shown in Table 1:

Table 1: Huffman symbols stream

Symbol	**H**	**a**	**s**	**m**	**u**	**i**	**e**	**l**	**B**	**w**	**d**	**O**
Count	6	6	3	2	2	2	1	1	1	1	1	1

The tree of Huffman example is shown below in Fig.1:

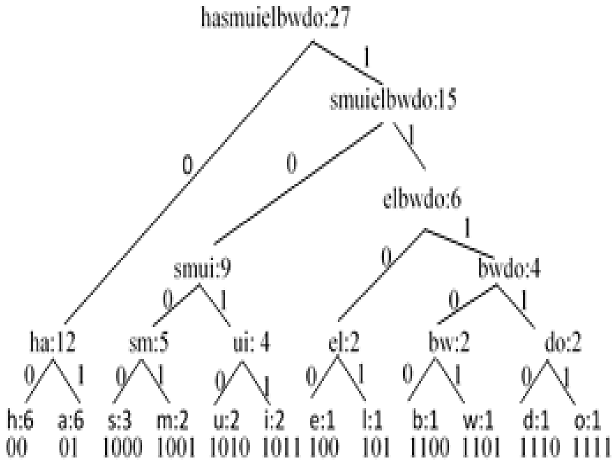


Fig. 1: Huffman Tree

The Table 2 illustrates the total length of compression output.

Table 2: Huffman total length

Symbol	Freq.	code	code length	total length
H	6	00	2	12
A	6	01	2	12
S	3	100	3	9
M	2	1001	4	8
U	2	1010	4	8
I	2	1011	4	8
E	1	100	3	3
L	1	101	3	3
B	1	1100	4	4
W	1	1101	4	4
D	1	1110	4	4
O	1	1111	4	4
Total				79 bits

Input:

elhamabushwashihamidaoushah = 27 characters

= 27 * 8 bits for each character

= 216 bits

Output:

100101000110010111001010100000110101100000101100011001101 11110011111101010000000100 = 82 bits

3.3 Shannon-Fano Coding

This is one of an earliest technique for data compression that was invented by Claude Shannon and Robert Fano in 1949.In this technique, Shannon-Fano Algorithm is a top-down approach, 1 sort symbols according to their frequencies [4].

A simple example will be used to illustrate the algorithm:

Input stream:

elhamabushwashihamidaoushah

Frequency symbols in stream as shown in Table 3:

Table 3: Frequency of character

Symbol	**h**	**A**	**s**	**M**	**u**	**i**	**e**	**l**	**b**	**w**	**d**	**O**
Count	6	6	3	2	2	2	1	1	1	1	1	1

Recursively divide into two parts, each with approximately same number of counts, i.e. split in two so as to minimize difference in counts. Left group gets 0, right group gets 1 [6]. Fig. 2 shows the Shannon coding tree.

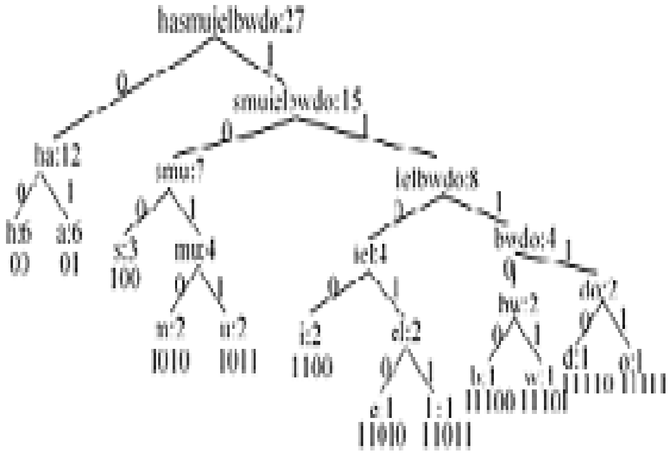


Fig. 2: Shannon-Fano Tree

The following table illustrates the complete operation for this algorithm.

Table 4: Shannnon-Fano total length

Symbol	Freq.	Code	code length	total length
H	6	00	2	12
A	6	01	2	12
S	3	100	3	9
M	2	1010	4	8
U	2	1011	4	8
I	2	1100	4	8
E	1	11010	5	5
L	1	11011	5	5
B	1	11100	5	5
W	1	11101	5	5
D	1	11110	5	5
O	1	11111	5	5
Total				87 bits

Input:

elhamabushwashihamidaoushah = 27 characters

= 27 * 8 bits for each character

= 216 bits

Output:

110101101100011010011110010111000011101001000011000001101
01100111100111111011100000100 = 87 bits

3.4 LZW Encoding Algorithm

LZW is the first letter of the names of the scientists Abraham Lempel, Jakob Ziv, and Terry Welch, who developed this algorithm. LZW compression is a lossless compression algorithm.

LZW algorithm is just like a greedy approach and divides text into substrings. LZW compression algorithm is dictionary based algorithm which always output a code for a character. Each character has a code and index number in dictionary. Input data which we want to compress is read from file. Initially data is entered in buffer for searching in dictionary to generate its code. If there is no matching character found in dictionary. Then it will be entered as new character in dictionary and assign a code. If character is in dictionary then its code will be generate. Output codes have less number of bits than input data. This technique is useful for both graphics images and digitized voice [7]. Table 4 shows the complete operation of this algorithm.

Table 5: LZW operations

Input string= elhamabushwashihamidaoushah			
Character input	*Code output*	*New code value*	*New string*
E	E	None	
E	L	256	EL
L	H	257	LH
H	A	258	HA
A	M	259	AM
M	A	260	MA
A	B	261	AB
B	U	262	BU
U	S	263	US
S	H	264	SH
H	W	265	HW
W	A	266	WA
A	S	267	AS
S	H	264	SH
SH	I	268	SHI
I	H	269	IH
H	A	258	HA
HA	M	270	HAM
M	I	271	MI
I	D	272	ID
D	A	273	DA
A	O	274	AO
O	U	275	OU
U	S	263	US
US	H	276	USH
H	A	258	HA
HA	H	277	HAH

Input: elhamabushwashihamidaoushah = 27 characters

= 27 * 8 bits for each character

= 216 bits

Output: ellhhaammaabbuusshhwwaasshiihhammiiddaauushhah = 87 bits

4. MEASURING COMPRESSION PERFORMANCES

Performance measure is use to find which technique is good according to some criteria. There are various criteria to measure the performance of compression algorithm. Since the compression behavior depends on the redundancy of symbols in the source file, it is difficult to measure performance of compression algorithm in general. The performance of data compression depends on the type of data and structure of input source. The compression behavior depends on the category of the compression algorithm: lossy or lossless [8]. Following are some measurements used to evaluate the performances of lossless algorithms.

Compression Ratio: is the ratio between the size of the source file and the size of the compressed file.

$$compression\ ratio = \frac{Amount\ data\ bits\ compression}{Amount\ data\ bits\ before\ compression}$$

Compression Factor: is the inverse of the compression ratio. That is the ratio between the size of the source file and the size of the compressed file.

$$compression\ factor = \frac{Amount\ data\ bits\ before\ compression}{Amount\ data\ bits\ after\ compression}$$

Saving Percentage: calculates the shrinkage of the source file as a percentage.

$$saving\ percentage = \frac{Amount\ data\ bits\ before\ compression - Amount\ data\ bits\ after\ compression}{Amount\ data\ bits\ before\ compression}\%$$

5. Results

In this work mainly focused on performance of four lossless compression algorithms.

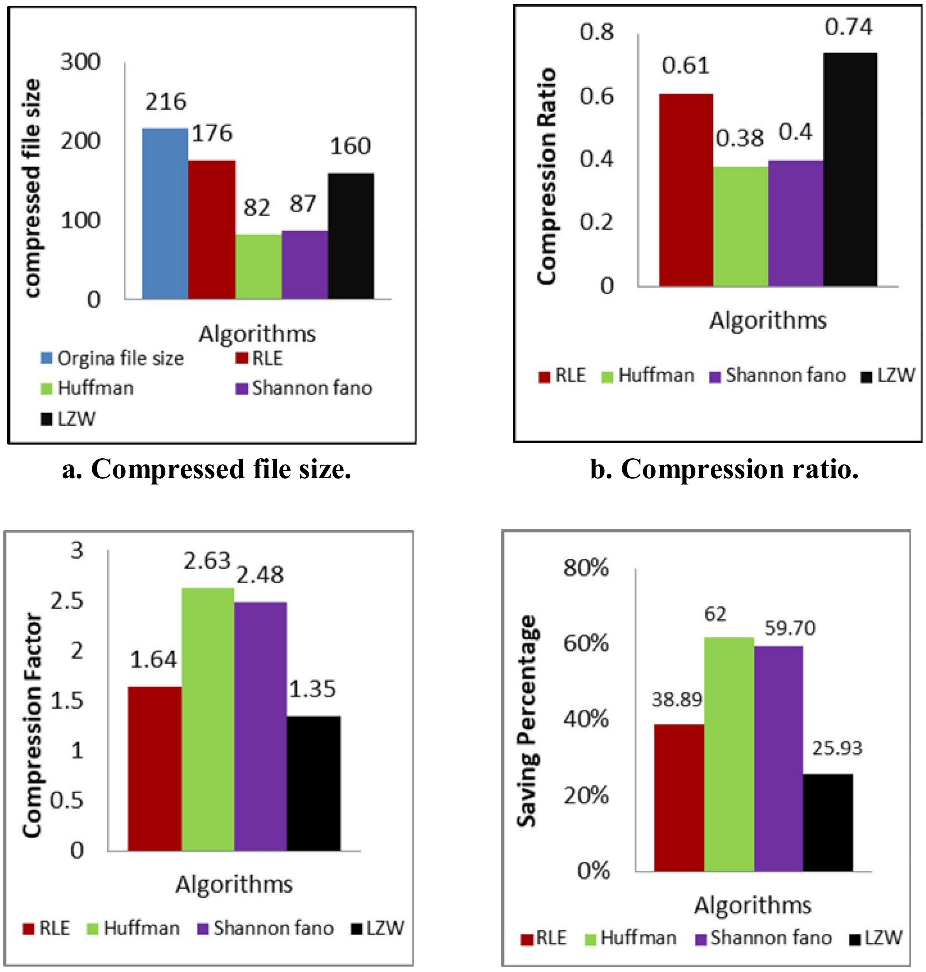
The table 6 shows the comparative results (output size, compression ratio, compression factor, and saving percentage) between our selected algorithms.

Table 6: Comparative results.

	Algorithm			
	RLE	***Huffman***	***Shannon-Fano***	***LZW***
Input size (bits)	216	216	216	216
Output size (bits)	176	82	87	160
Compression Ratio (%)	81	38	40	74
Compression Factor	1.23	2.63	2.48	1.35
Saving Percentage(%)	19	62	60	26

According to the results shown in table 6, Huffman and Shannon-Fano coding can reduce the file size around 50% of original file size, while the Shannon-Fano Coding, and LZW Encoding algorithm can reduce approximately (1.2%), This finding was consistent with study conducted Achinta,2016 [1]. Study, the compression ratio of Huffman and Shannon-Fano coding were more effective than the Shannon-Fano Coding, and LZW Encoding, this result compatible with study done by K.A. Ramya.2006 [2], compression factor is the inverse of compression ratio, and saving percentage of the Huffman and Shannon-Fano coding is better while compared to others.

The bar chart shows the original file size before and after compression, comparison between compressed file size, compression ratio, compression factor, and saving percentage, by using Huffman, Shannon-Fano coding, Run Length encoding, and LZW encoding in Fig. 3. In Fig. 3a. the graph shows original file size before compression (216 bits), and the size after compression using Huffman (82 bits), Shannon-Fano (87 bits), Run Length (176 bits), and LZW (160 bits), it is clear that the file compressed near to the half by using by Huffman and Shannon-Fano coding.



a. Compressed file size. **b. Compression ratio.**

c. Compression factor. **d. Saving percentage.**

Fig. 3. Compartive Results

In Fig. 3b compression ratio calculated by Huffman and Shannon-Fano coding are almost half the compression ratio calculated by Run Length encoding, and LZW encoding, in fig. 3c the compression factor calculated by Huffman and Shannon-Fano coding is almost double the compression factor calculated by the Run Length encoding, and LZW encoding.

In Fig. 3d the saving percentage is almost 60% by using Huffman and Shannon-Fano coding with and around 26% by using Run Length encoding, and LZW encoding.

6. CONCLUSION

Data text compression using Huffman and Shannon-Fano coding algorithms give small size of data in compare to Run-length and LZW encoding.

Compression Huffman and Shannon-Fano coding algorithms in data text resulting a smaller size of data than the size of the data by Run-length and LZW encoding

Huffman and Shannon-Fano coding are very powerful over Run-length and LZW encoding, they provide better results and reduce the size of the text. Run-length encoding more effective when data text with strings of continuous repeated symbols.

REFERENCES

1. Achinta Roy, Dr. Lakshmi Prasad Saikia, "A Comparative Study of Lossless Data Compression Techniques on Text Data" International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 2, February 2016.
2. K.A. Ramya1, M.Pushpa, ,M.Phil Student, Assistant Professor, "Comparative Study on Different Lossless Data Compression Methods", International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-2, Issue-1,January 2016 ISSN: 2395-3470 www.ijseas.com.

3. P.RAVI1, Dr.A.ASHOKKUMAR, “A Study of Various Data Compression Techniques”, International journal of computer scinece & communication, Volume 6, Issue 2, April-September 2015, ISSN:0973-7391.
4. Amit Jain a * Kamaljit I. Lakhtariab, Prateek Srivastava, “A Comparative Study of Lossless Compression Algorithm on Text Data”, Proc. of Int. Conf. on Advances in Computer Science, AETACS, 2013.
5. Dr. AMIN MUBARK ALAMIN IBRAHIM, Dr. MUSTAFA ELGILI MUSTAFA, “Comparison Between (RLE And Huffman) Algorithmsfor Lossless Data Compression”, (IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH, Volume No.3, Issue No.1, December – January 2015, 1808 – 1812.
6. B.A. Al-hmeary, “Role of Run Length Encoding on Increasing Huffman Effect in Text Compression”, Journal of Kerbala University , Vol. 6 No.2 Scientific. 2008.
7. Mamta Sharma, S.L. Bawa D.A.V. college, “Compression Using Huffman Coding”, IJCSNS International Journal of Computer Science and Network Security”, VOL.10 No.5, May 2010.
8. Pooja Singh, “LOSSLESS DATA COMPRESSION TECHNIQUES AND COMPARISON BETWEEN THE ALGORITHMS”, International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056, p-ISSN: 2395-0072, Volume: 02 Issue: 02 May-2015. www.irjet.net

9

A CIS Framework for Libyan District

A CIS Framework for Libyan District

Amal O. Abdulghader
College of Computer Technology
Zawia, Libya
amal_osaad2003@yahoo.com

Khari A. Armih
College of Computer Technology
Zawia, Libya
khari.armih@gmail.com

Rabia Omer A. Saad
Libyan Center For Remote Sensing & Space

Abstract

In the last decades, the proliferation of digital data and the availability of digital map, and the use of geographic information system (GIS) has become the best technique to develop the cadastral information system (CIS). The digital cadastral database (DCDB) that shows real coordinates for cadastral maps is hampered by many land laws in the country. We present a new Framework for developing CIS applications to assist real estate registration. The new framework can be used as a guide to developers helping them in creating a plan of development and defining the system requirements. A cadastral web mapping solution for a Libyan district (CWMSLD) is developed using the proposed framework. CW-MSLD System based on a pilot case study in the capital city of Libya. The prototype is developed using modern GIS techniques (Web Mapping). Web Mapping techniques make maps and geo-information available to groups of end users through a web page. The prototype tool triggers the map server software which integrates the map data stored in DCDB with the land register data stored in the database. The information derived from the system can be used to register or transfer ownership for the cadastral map and further issue a cadastral certificate for the registered cadastral (real estate).

General Terms: Algorithms, Design
Keywords: GIS, Cadastral, Digital Mapping, Framework, Land Registration

1. Introduction

A GIS is a computerised system for storage, retrieval, manipulation, analysis, and display of geographically referenced data [2].
GIS technology integrates common database operations such as query and statistical analysis with the unique visualization and geographic analysis benefits offered by maps. As an information system (IS), GIS is designed to work with data referenced by spatial or geographic coordinates. In other words, a GIS is both a database system with specific capabilities for spatially-referenced data, as well as a set of operations for working with the data [10]. The development of GIS is older than you may think. It started from 1960s in Canada [11]. GIS developed layers, overlay calculations, data structure, scanning as data entry and so on for land resources. Furthermore during the same year, U.S. Census developed digital enumeration districts and geo-coding for address matching in commercial area, and ESRI created in 1969 as one of the most successes software company in the world. As seen by today, the use of GIS grew strongly and fast in different applications like business, government, and academic.
Cadastral system is one of the GIS applications that can be successfully developed by using GIS techniques. Cadastral refers to a map or survey showing administrative boundaries and property lines [9]. Cadastral information system (CIS) is a system that consists of two sub-system i.e cadastral map system and land register system. A cadaster administration is very important for the owners to register and get a certificate for their own specific property or cadastral. The government would use this information and use the outputs from

cadastral and land registry system as input for other projects like water supply and highway projects.

The possibility of using geographic information system (GIS) techniques have generated wide interest in cadastral system, and many countries have developed digital cadastral maps by using GIS techniques known as digital cadastral databases (DCDB). However, this DCDB needs application software to add, update, delete, and search query in this database. This mean the application software will enable the map modification work to be carried out in more easily and efficient manner using DCDB.

Current cadastral work in Libya is carried out manually and this has led to a number of serious problems resulting in legal battles for ownership of land titles. Title/deeds reported as being lost have been re-created resulting in having duplicate files for a signal cadastral (real estate). Furthermore, the current manual system does not support and track updates.

In this paper, we propose a new framework for Cadastral information system. We also present a practical and functional prototype e-tools based on a Cadastral Web-Mapping Solution for a Libyan district (CW-MSLD) to assist real estate ownership registration.

2. Related Work

Several frameworks and techniques have been proposed to develop CIS systems and applications. Here we compare and describe a number of existing CIS projects.

2.1 LR&CIS

Turkey has developed the Turkish cadastral automation system [5]. The most important goal for developing Land Registry and Cadaster Information System (LR&CIS) was to improve the services by computerizing the system and make land information system (LIS) to be multipurpose in order to help other organizations by given accurate and reliable data and information. Turkish Cadastral Automation System had been developed by using ArcIMS 2 public web services from ESRI. That system was used by the General Directorate, Ankara Regional Directorate in Turkey. In conclusion, Turkey has a complete LR&CIS integrated system. This system was bought with high cost but gives a lot of services and saves time in different organization in the country. ESRI technology has been used very successfully in cadastre side, ArcSDE for managing the cadastre data at the center of system, and ArcEditor 5 for all cadastre activities in harmony with the land registry side. ArcIMS is used for serving data via Internet for external user. Thereby, the most important LR&CIS benefits are having a central database have a backup for data, easy and faster to access data via the Internet and no damages for original documents in the digital archives. External users can access the system and get services like zoom in, and zoom out the map, pan and access and display data. LR&CIS system is one of most important part of Turkey's e-government structure. This system is not complete yet for covering all Turkey's parcels.

2.2 ECIM

The Egyptians had been well known for their maps in ancient times. They had drawn maps on parchment to show the gold mines at Coptes during the period of 1292 B.C. - 1225 B.C [3]. Egypt is one of the first countries which tried to manage cadastral by enforcing rules and regulations for that. The Egyptian Survey Authority (ESA) had started in 2002 to develop The Egyptian Cadastral Information Management (ECIM) project supported by the Ministry for Foreign Affairs of Finland [6]. The most important aim for ESA/ ECIM was to have a complete computerized system which included digital LIS based on cadastral data. The ECIM project developed a system which integrated the existing ESA, Real Estate Publicity Department (REPD), and Real Estate Taxation Department (RETD). ECIM project is an automated system which enabled the monitoring of various day-to-day activities between various offices. The ArcSDE technology incorporated into ArcGIS 6 server is used to access multi-user geographic databases. The area was chosen Damanhour district in Beheira province, and it is a rural area which is approximately 160 km2. Oracle, ArcSDE, ArcCadastre 7, and MapObjects 8 and Visual Basic were used to develop the system. After the pilot study was successfully implemented, the ESA continued to develop the system that it could be applied nationwide. However, the cadastral and land registration system is in fact more complex and has unclear procedures. The ECIM project had highlighted a number of problems in the existing systems during the long analysis phase. ECIM project has worked hard to understand cadastral procedures, land registration legalities, and how to calculate tax for parcels or other objects. The biggest problem was how to integrate and connect four organizational levels and how data is transferred between them. The most important ECIM project output is the Unified Cadastral Database (UCD), which combines the map data with the attributes. UCD is designed based on user requirements. There are many functions that had been included in the system like continuous and automatic updating cadastral work, monitoring and printing out different map outputs, reports and statistics, converting data from analogue to digital, de-centralizing the system and achieving synchronization which co-ordinates between four organizational levels. In conclusion, ESA has developed a web application which handles the publishing of geographic and tabular information via the Internet using ArcIMS (Internet Mapping Server) from ESRI and published information could be accessed by ESA's regional offices or other stakeholders. This project brought many benefits, not only for ESA but also for the society and government. It had been tried to solve informal registration problems in Egypt. ECIM project will increase the social security by using a digital database and it is possible to standardize products in a more efficient way with lower cost. This could attract more customers and hence increase the income, and provide faster delivery services to customers. Other benefits of the ECIM project is the improved office environment, the digital storing media takes less space than the analogue environment of storing.

2.3 GIS-Sofia

GIS-Sofia Ltd [4] in Bulgaria had developed a new cadastre and property register for Sofia Municipality. The development of an Information System of the Cadastre (ISC) was used for the territory of the Sofia capital city. This project had integrated the registry and cadastre offices. A prototype of the ISC project had been developed by using ArcView 9, AutoCAD Map R 4.0, and SQL Server. In order to develop any system, we first need to understand the laws and regulations in the system environment. Bulgaria had updated the rules and data exchange between registry and cadaster offices in April 2000. Bulgaria started working on developing the ISC from 1987, with digitization cadastral maps. Gradually the integration between attribute database and geographic data was initiated. Sofia completed its digitization by using Arc/Info 10 3.5 and established its attribute database in 1996. Cadastral mapping in Sofia is maintained in two scales: 1: 1 000 for urban area, and 1: 500 for the compact city only. In 1998, GIS-Sofia started updating its digital map base starting from nationwide Sofia city and finished the task in the summer of 2001 by giving the unique identifier for each cadastre object. The project team started work on a pilot study first where property was identified. The property verification work was identified and some techniques and procedures were developed to solve these problems. Finally, the GIS-Sofia completed digitizing about 133000 properties and 242000 buildings. The next step was to link cadastral data (graphic data) to relevant data from ownership (attribute data). Input correction and carrying out information searches was done by using registration software. GIS-Sofia developed the ISC web-based system and the ISC prototype consists of three parts namely ISC center, Library of function and User Interface. SQL Server at ISC center integrated and managed the different parts. GIS-Sofia created an efficient environment to protect information which is not only important to cadaster and registry office but useful to a lot of customers. The digital information was used to develop and build the city. The digital cadaster used in design works for underground project, water supply and sewerage project, and the Bulgarian telecommunication company. However, GIS-Sofia had been updating urban cadastral information system by adding 3D Visualization using remote sensing and GIS. GIS-Sofia used ArcView software for collecting data needed to obtain 3D buildings. They used specific tools to represent 3D buildings to reach the most geometrically correct and real looking three dimensional objects. The geometrical accuracy is very important in cadastral application. The use of aerial- photo provides higher completeness of information, but is considerably more expensive [1]. In conclusion, the information and outcomes from this project is very important to a lot of organization in Sofia city. GIS-Sofia did not stop at this point but they are trying to design technology and applications for furnishing information from the database via Internet and thereby increase the quality of their services.

2.4 DSMM

The department of survey and mapping Malaysia (DSMM) has been looking for other countries experience and experimentation in developing cadastral system. DSMM started strong with cadastral reform and coordinate cadastre [8]. In 1986, the first pilot project was done in Johor state and it created for the DCDB at a scale of 1:4,000 and the first pilot project was successful. The next project was implemented in Penang State in 1993. Based on both these projects, cadastral reform had been done for Peninsular Malaysia in 1995 where a connection network was implemented. A survey accurate DCDB and coordinated cadastre should not be created without fully understanding the place it holds and the effects it has on the operation of the cadastral system [12]. All reforms to introduce a survey accurate DCDB and an improved cadastral surveying system (collectively a coordinated cadastral system /coordinated cadastre) go hand in hand with reforms to the wider cadastral system including reforms to the title registration system. The model proposed for coordinated cadastre is based on a complete DCDB. The accuracy of data acquisition is very important for cadastral survey. Many devices helped the surveyor in their work like Digital Theodolite, Total Station, Digital Level, Global Positioning System (GPS) and Digital Photogrammetric. Nowadays, the District Survey offices are using GPS and it increases survey accuracy, productivity, and reduces costs. The DSMM is collaboration with University of Technology Malaysia (UTM) to determine the feasibility of introducing a Coordinated Cadastral System for Malaysia, and find best techniques for integrating cadastral data with attribute data. In conclusion, it can be deduced that acquisitions, collection, and conversion of analog data into digital format are though very important, it is a time consuming job in building a computerized information system, DSMM has been restructured in 1994 to face new challenges more effectively by using new technology in the field of survey and mapping. It will have a strong base to develop cadastral information system by using GIS technology. To look more closely, DSMM is now better set to meet the challenges of the nation's Vision 2020 which is in the process of establishing an ”Electronic Government”.

2.5 Summary of the Cadastral Projects Reviewed

Table 1 shows the main properties for four projects which the author has investigated.

Table1: Summaries of the Cadastral Projects Reviewed

Country properties	Turkey	Egypt	Bulgaria	Malaysia
Land registration	Deeds	Registration of deeds until 1975 Registration of title since 1976 (80% has been done)	Deeds	Title
Projects services	Regional Directorates land registry offices cadastre offices	Egyptian landowners and relevant authorities	Registry Office Cadastre Office	Department of Survey and Mapping (DSMM)
Survey coverage Completeness	cadastral survey of 97 % of urban areas and 77 % of rural areas end of 2005	Governorate of Beheira, Egypt 54 village	Complete digitizing about 133000 properties & 242000 buildings for Sofia city	two cities were surveyed: Johor, Penang
Technology used & Tools	ESRI technology (ArcEdit, ArcSDE ArcIMS, and ArcCatalog) SQL server VB,C++ and Delphi	Oracle, ArcSDE, ArcCadastre, and MapObjects	Visual Studio environment , Avenue language ArcSDE ,Arc/Info AutoCAD MAP SQL Server Windows NT 4.0 environment Linux Red Hat 7.1 Environment	Global Positioning System (GPS)
Method or approach	"Cadastre 2014" approach	object-relational data model	Object-relational data model	"Cadastre 2014 " approach
Significant improvement	Website design	Full integrated project, Multipurpose Connected 3 applications	Website design	Land reform & established DCDB Coordinated cadastre GPS control network
Other comments	Not complete yet	Completed in August 2006 with Project financing :1 million euros	3D Cadastre under study	The project still under development

3. Proposed Framework

The first step in investigating trends and proposing framework, we looked at the existing four cadastral systems project and attempts to define and organize the system requirements for developing a cadastral system. A general framework diagram is shown in Figure 1. It is proposed by defining the eight elements that all cadastral system shares and it further defines the recent management tools and techniques used for design and development of cadastral information system. Basically, the framework identifies the relationships between the eight elements of the framework with one another.

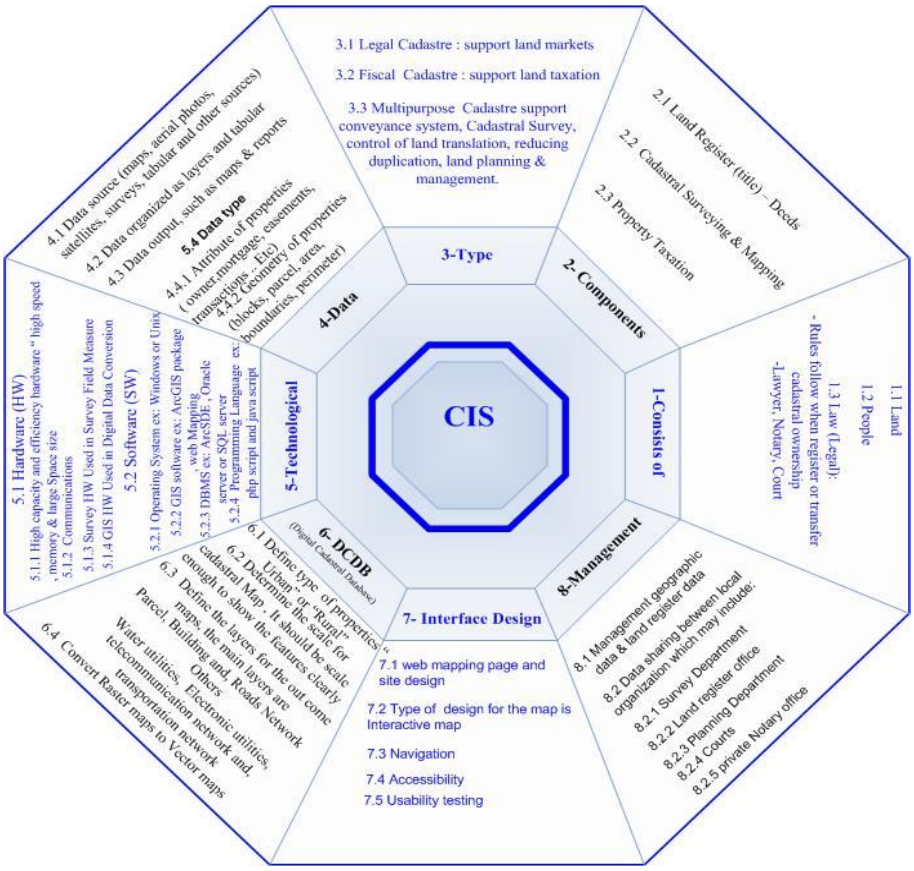

CIS
1-Consists of
1.1 Land
1.2 People
1.3 Law (Legal):
- Rules follow when register or transfer cadastral ownership
-Lawyer, Notary, Court
2- Components
2.1 Land Register (title) – Deeds
2.2 Cadastral Surveying & Mapping
2.3 Property Taxation
3-Type
3.1 Legal Cadastre : support land markets
3.2 Fiscal Cadastre : support land taxation
3.3 Multipurpose Cadastre support conveyance system, Cadastral Survey, control of land translation, reducing duplication, land planning & management.
4-Data
4.1 Data source (maps, aerial photos, satellites, surveys, tabular and other sources)
4.2 Data organized as layers and tabular
4.3 Data output, such as maps & reports
5.4 Data type
4.4.1 Attribute of properties (owner,mortgage, easements, transactions .. Etc)
4.4.2 Geometry of properties (blocks, parcel, area, boundaries, perimeter)
5-Technological
5.1 Hardware (HW)
5.1.1 High capacity and efficiency hardware " high speed ,memory & large Space size
5.1.2 Communications
5.1.3 Survey HW Used in Survey Field Measure
5.1.4 GIS HW Used in Digital Data Conversion
5.2 Software (SW)
5.2.1 Operating System ex: Windows or Unix
5.2.2 GIS software ex: ArcGIS package , web Mapping
5.2.3 DBMS ex: ArcSDE , Oracle server or SQL server
5.2.4 Programming Language ex: php script and java script
6- DCDB
(Digital Cadastral Database)
6.1 Define type of properties " Urban" or "Rural"
6.2 Determine the scale for cadastral Map , It should be scale enough to show the features clearly
6.3 Define the layers for the out come maps, the main layers are Parcel, Building and, Roads Network Others
Water utilities, Electronic utilities, telecommunication network and, transportation network
6.4 Convert Raster maps to Vector maps
7- Interface Design
7.1 web mapping page and site design
7.2 Type of design for the map is Interactive map
7.3 Navigation
7.4 Accessibility
7.5 Usability testing
8-Management
8.1 Management geographic data & land register data
8.2 Data sharing between local organization which may include:
8.2.1 Survey Department
8.2.2 Land register office
8.2.3 Planning Department
8.2.4 Courts
8.2.5 private Notary office

Figure 1: Cadastral Information System (CIS) Framework

First, the developer needs to define the type of CIS based on user requirements and feasibility study for the project where each cadastral system mostly consists of land, people, and legal procedures that need to be followed.

Second, the developer needs to select the components based on the type of CIS. For example, if the developer wants to develop the legal cadastral system, then the land register, cadastral surveying and mapping components ought to be selected.

Third, the developer needs to capture the necessary data and subsequently select the appropriate software to create the DCDB by digitizing the maps data. For interface design element, the developer needs to use web mapping to fulfill sound GUI requirement. Next, management element needs to organized and next you need to establish the land register, cadastral surveying and mapping data between different departments inside the cadastral office and outside the organization.

3.1 Elements of proposed CIS Framework

The eight elements for the proposed GIS framework as listed below:

- *CIS Consists of*: CIS consists of land (location), people (landowner) and law (rules to register and certify the ownership for owner). Landowners users are key components in providing data for CIS. Lawyers and notary users are responsible for certifying the ownership and survey users support the cadastral system with cadastral survey data.
- *Components of CIS*: CIS components are land register (title/deeds), cadastral surveying and mapping, and property tax-ation. Each component is a subsystem in CIS and it provides the CIS with specific data. The combination between the three components is used to carry out all registration operations and contracts on land rights.

- *Type of CIS*: There are three types of CIS namely legal cadastre, fiscal cadastre and multipurpose cadastre. The type of CIS used in the early stage of development defines the system domain and services.
- *Data*: Data requirements must be defined - thinking ahead to future policy developments. CIS includes two types of data which are land register data and survey cadastral data (spatial data). The spatial data can be captured from different resources (i.e. field survey, remote sensing, GPS, aerial photos). The output data for CIS can be maps, tables and reports.
- *Technology*: CIS require high capacity hardware and network infrastructure (ICT) to enable CIS to be accessed by different users on related organizations. The required development software for CIS is GIS software (i.e ArcGIS,Web Mapping, ArcIMS ..etc). The GIS software can be integrated with some programming language such as PHP and Java and the database used includes SQL Server and MySQL. This integration between GIS software and programming language helps to develop all user requirements. The technology should support:
  - Security, reliability, continuity of service.
  - Distribution, publication of data.
  - Use of remote access (public or private).
  - Data convergence issues.
- *Digital Cadastral Database (DCDB)*: Analogue cadastral data is computerized and store as DCDB. DCDB can be created by digitizing the cadastral map (convert raster maps to vector maps). The map scale should be defined based on the features required for digital map. DCDB allows to store the maps in different layers (i.e. parcel layer, building layer ... etc).
- *Interface Design*: GIS techniques support great interface design to CIS. The users can interact with cadastral map to get cadastral information. Web mapping is a new technique that

supports web map design. System accessibility and usability is tested interface design for the CIS.

- *Management*: Management data is most important issue for CIS. DCDB would improve the management capabilities of cadastral data. The data can be shared between survey departments, land register offices and others organization based on type of CIS that is developed.

3.2 Methodology

The CW-MSLD System prototype is actually a legal cadastre system and the component for this system includes land register (title/deeds), cadastral surveying and mapping. The GIS technology will be used to develop this system and the interface design where web mapping occurs by using open source software. In addition to these is Arc/Info software that is used for creating DCDB. There are two important benefits for using CIS framework. First, it is considered as a guide to developers helping them in creating a plan of development and defining the system requirements. This can be seen on design of the CIS framework as eight elements with specific properties. The developer can implement the eight element sequence to reach the desired goal. The second benefit is the ability to define data types. As mentioned before, the cadastral system integrates between attribute data (owner data, mortgage... etc) and spatial data (parcel, area ...etc). The CIS framework defines in shortly data type, data source, and data organized in data element (Element 4) and also points the steps for creating DCDB for the spatial data in DCDB element (Element 6).

3.2.1 Prototype Development Tools

Tool is a computer-based application which supports the use of a modeling technique. Tool-supported modeling functionality includes abstraction of the object system into models, checking that models are consistent, converting results from one form of model and representation to another, and providing specifications for review.

- **Arc/Info** is the complete GIS product to build a comprehensive desktop GIS. As a de facto standard for GIS professionals, ArcInfo provides tools for data integration and management, visualization, spatial modeling and analysis, and high-end cartography. It supports single-user and multi-user editing and automates complex workflows. This software is from ESRI and it used to gather, build, manage data, and analyze geographic relationships, discover new information, and produce publication quality maps for cadastral office which can be used as cadastral index map.
- **MapServer** is a CGI program that sits inactive on your web server [7]. When a request is sent to the MapServer, it uses information passed in the request URL and the map file to create an image of the requested map. The request may also return images for legends, scale bars, reference maps, and values passed as CGI variables. It is open source software. It can be greatly extended and customized, and it can be built to support many different input data formats and output types.

3. CW-MSLD implementation

System implementation describes the development tools that have been used in developing this system. The guidelines for using the CW-MSLD System will be given. The system developed as web mapping page makes use of the map server and PHP as the core programming language techniques. Besides that, the JavaScript and Rosa Applet have been used to help the system interfaces become more interesting and easy to use. For example the Rosa Applet tools used image button tools that help the user interact with the map in easy ways such as ZOOM IN, ZOOM OUT etc. The RDBMS used to build attribute data for this system is MySQL and for the map data (geographic data) are stored in three types of files that relate with each other to produce the vector map such as parcel layer. The merge between the attribute data and map data is done by using map server techniques. Table 2 shows all available features in CWMSLD.

Table 2: CW-MSLD System Features.

No	Statement	Available
1	Administrator able to add new user to login to the system	√
2	Access control for authorized user to login to the system	√
3	Administrator able to monitor the login users to system	√
4	Working with real coordinates map	√
5	Use the map to get requested parcel	√
6	The system able to register the parcel not included on the map	
7	Updating certificate information before issuing certificate	√
8	The system able to register a whole cadastral (parcel/building)	√
9	The system able to register subdivides cadastral (parcel/building)	
10	The system able to transfer ownership for registered real estate	√

4.1 System Security

The system security is very essential to this system especially when the system is running on the internet environment network. Users are grouped into three groups. At the highest level is the system administrator and second level or group is the manager and the last group members are the registered staff users. Valid user ID and password are required whenever a user access the system. This is to prevent unauthorized users from using the system.

4.2 CW-MSLD System Implementation

The system actually has three types of users which are manager, staff and administrator. The Figure 2 shows the system modules tree based on users type.

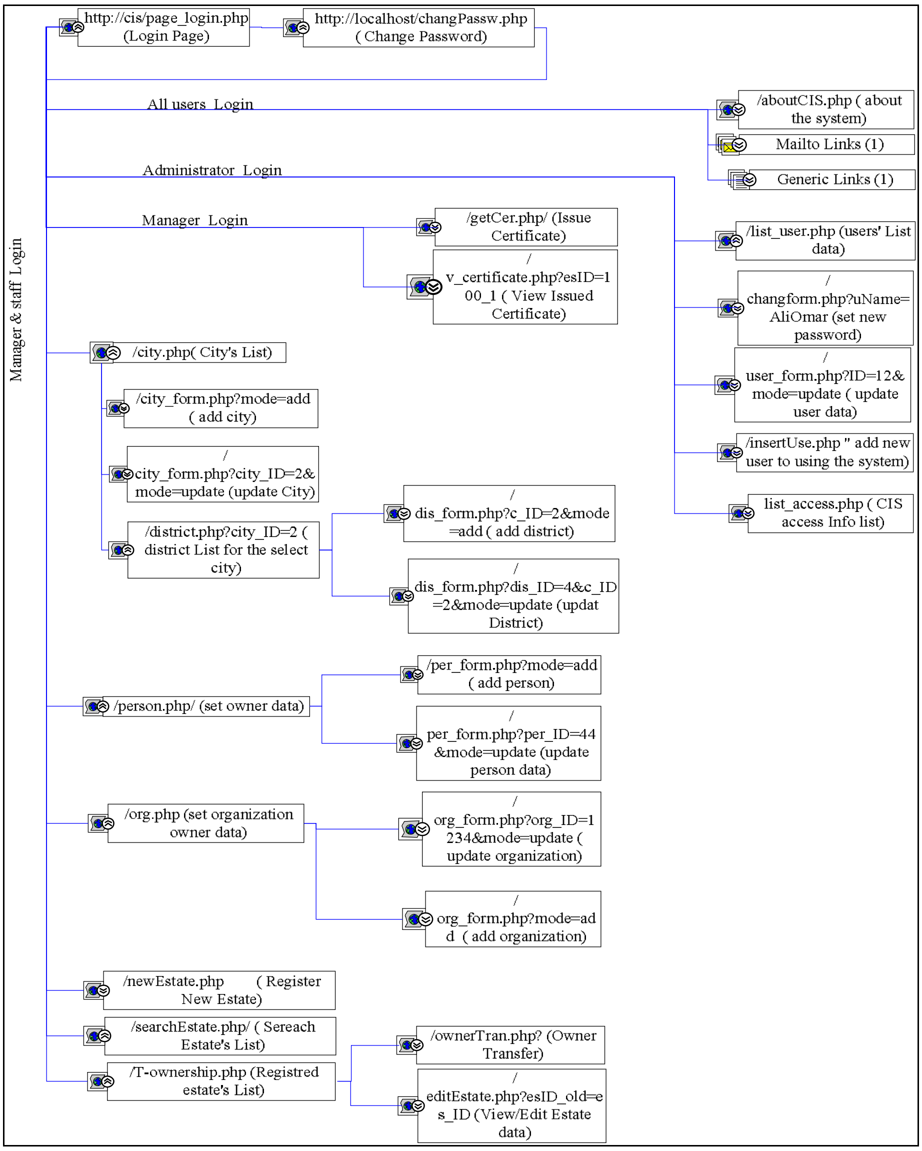

Manager & staff Login
http://cis/page_login.php (Login Page)
http://localhost/changPassw.php (Change Password)
All users Login
/aboutCIS.php (about the system)
Mailto Links (1)
Generic Links (1)
Administrator Login
Manager Login
/getCer.php/ (Issue Certificate)
/ v_certificate.php?esID=100_1 (View Issued Certificate)
/list_user.php (users' List data)
/ changform.php?uName=AliOmar (set new password)
/ user_form.php?ID=12&mode=update (update user data)
/insertUse.php " add new user to using the system)
list_access.php (CIS access Info list)
/city.php(City's List)
/city_form.php?mode=add (add city)
/ city_form.php?city_ID=2&mode=update (update City)
/district.php?city_ID=2 (district List for the select city)
/ dis_form.php?c_ID=2&mode=add (add district)
/ dis_form.php?dis_ID=4&c_ID=2&mode=update (updat District)
/person.php/ (set owner data)
/per_form.php?mode=add (add person)
/ per_form.php?per_ID=44&mode=update (update person data)
/org.php (set organization owner data)
/ org_form.php?org_ID=1234&mode=update (update organization)
/ org_form.php?mode=add (add organization)
/newEstate.php (Register New Estate)
/searchEstate.php/ (Sereach Estate's List)
/T-ownership.php (Registred estate's List)
/ownerTran.php? (Owner Transfer)
/ editEstate.php?esID_old=es_ID (View/Edit Estate data)

Figure 2: System Modules Tree

4. CW-MSLD Evaluation

Here we illustrate the system testing and implementation phases. The testing phase involves some modification to the pervious design phase and system testing has been done to minimize the programming and system error. At the implementation phase, system requirements such as hardware and software will be defined. Besides that, the system prototype interfaces and functionalities (module) will be fully demonstrated to users.

5.1 System Testing

Testing the system is a very important stage to ensure that all system requirements have been developed without errors. System testing can be done through some stages. The first stage is called unit testing or component testing and this testing done during the development of the system. Each component, script or module test isolates from other component or unit by checking the input and output for it. The second stage is called integration testing. The integration between components will be tested and in case there are any errors the components will be tested again. The third stage is called user acceptance testing and this testing done by users who request to develop the system. The third stage is called security testing. The final stage is called user acceptance testing and this testing done by users who request to develop the system.

5.1.1 Unit Testing

Unit testing focuses on testing module, script or component that has been designed by PHP, JavaScript, or Rosa Applet. For example, the developer tested the map tools button functionality such as Zoom in on a map or obtain information when clicking on the map by using Identify button that is designed by using Rosa Applet.

5.1.2 Integration Testing

After the unit testing has been done with satisfaction for each component or script, the integration testing is started to ensure the CW-MSLD System components worked together smoothly. The functional and non-functional requirements were tested in this stage. One example for integration testing is to search the parcel model by entering the parcel ID and if the GIS database has the parcel requested, the system will display it and it can use the data given to register new real estate.

5.2 Analysis of User Interface Evaluation

Working with user interface any system is dependent on users computer background and understanding of the system environment. Based on the evaluation, the system was found to be easy to use. The highest rating mean of 4.2 indicates that searching on the map to get information is easy. The results were converted into a bar chart in Figure 3 to show more clearly.

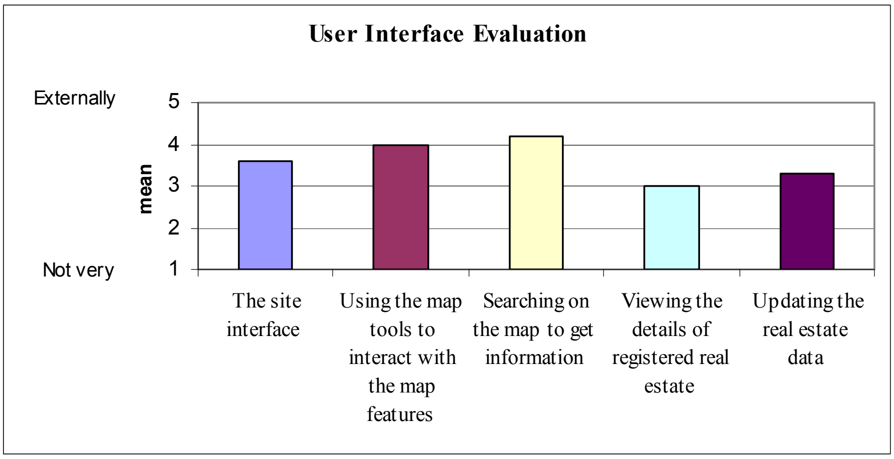


Figure 3: User interface evaluation bar chart

The bar chart in Figure 4 shows the evaluation for user interface satisfaction. The bar chart clearly indicates that the users are satisfied by with using the help tools.

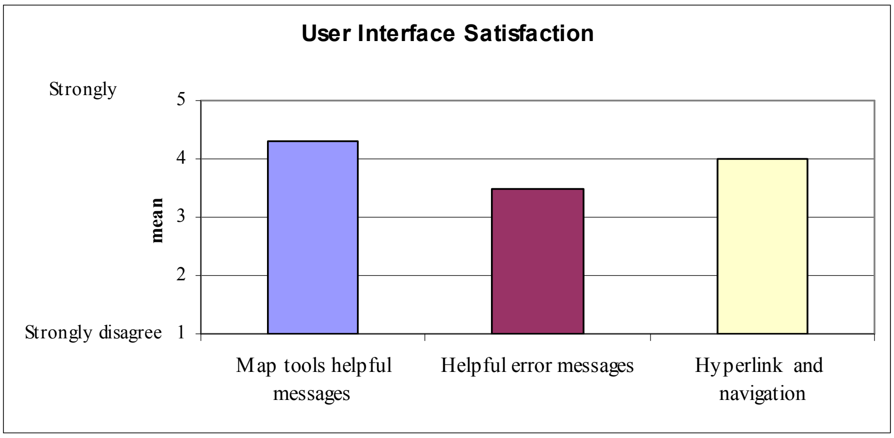


Figure 4: User interface satisfaction bar chart

5.3 Analysis of Evaluation Pertaining to Features

Figure 5 shows the evaluation of testing the accuracy of geographic data (map) as the data accuracy is the most important part of a successful GIS application. The bar chart indicates a good frequency for testing the accuracy for the parcels area, boundary, and location.

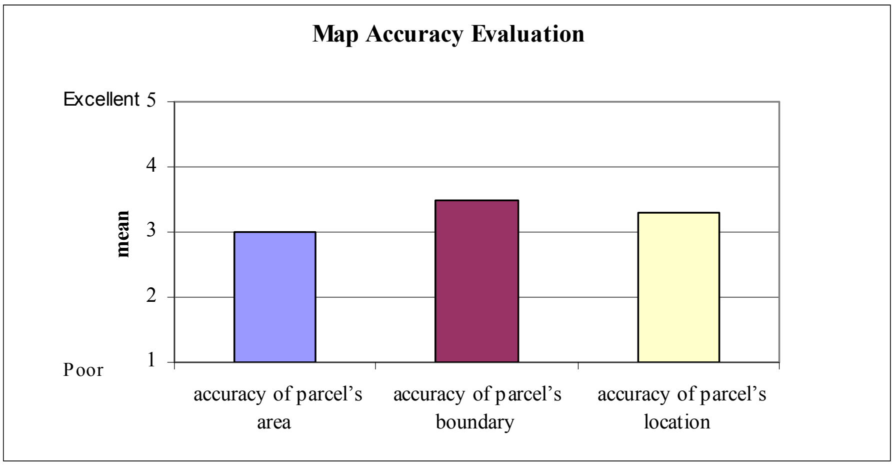


Figure 5: Map accuracy evaluation bar chart

5. Conclusions and Future Work

In this paper, we have presented a new framework for cadastral information system (CIS). The purpose of the new framework is to develop a cadastral web-mapping system, which assist real estate ownership registration. The proposed framework is obtained by studying the cadastral features, cadastral reform and cadastre 2014 vision which gives a general framework for cadastral. The proposed framework provides many options to computerize the day to day cadastral work at cadastral office and it further provides a secure way of keeping the database up-to-date. CW-MSLD System prototype was developed to increase the efficiency and effectiveness of the daily work on Tripoli Cadastral Office by using the modern GIS techniques (Web Mapping).

In future research we hope to link our system with other GIS software such as ESRI package like Arc/Info for editing the cadastral map.

References

1. Alexandrov A et al. Application of quickbird satellite imagery for updating cadastral information. In The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2004.
2. Joseph L. Awange and John B. Kyalo Kiema. Fundamentals of GIS, pages 191–200. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
3. T. Bernhardsen. Geographic Information Systems: An Introduction. Wiley, 2002.
4. Lazarov A Dechev, H. Cadastral information system of sofia. In Proceeding of FIG XXII International Congress Washington, D.C. USA, 2002.
5. Dr. Orhan Ercan and Dr. Emin Bank. Turkish cadastre automation system with esri technology, 2004.

6. Samir Elrouby Harju, K. Developing an automated cadastral information system in egypt. In Proceeding of Pharaohs to Geoinformatics FIG Working Week, Cairo, Egypt, 2005.
7. Bill Kropla. Beginning MapServer: Open Source GIS Development (Expert’s Voice in Open Source). Apress, Berkely, CA, USA, 2005.
8. Abdul Majid Bin Mohamed. Cadastral reforms in malaysia. In Proceeding of FIG, Commision7 meeting in Penang, Penang, Malaysia, 2007.
9. International Federation of Surveyors. The FIG Statement on the Cadastre. Publication (International Federation of Surveyors). International Federation of Surveyors (FIG), 1995.
10. J. Star and J.E. Estes. Geographic Information Systems: An Introduction. Prentice Hall series in geographic information science. Prentice Hall, 1990.
11. R. F. Tomlinson, Duane F. Marble, and H.W. Calkins. Computer Handling of Geographical Data: An Examination of Selected Geographic Information Systems. Bernan Associates, 1979.
12. I.P. Williamson. ”why cadastral reform?” In proceedings of National Conference on Cadastral Reform 1990, Melbourne, Australia, 1990.