

ACKNOWLEDGEMENT

It is my foremost responsibility to express my profound gratitude and sincere respect to my Research Guide **Dr.S.Subha** for her continuous support throughout my work. Her talents, dedication and enthusiasm for research will be a continuing source of inspiration

I wish to express my heartfelt gratitude to **Dr.G.Viswanathan**, Chancellor, VIT , Vellore, for providing facilities for the research work.

I am highly grateful to our Vice Presidents, **Shri. Sankar Viswanathan, Dr. Sekar Viswanathan and Shri. G.V.Selvam**, Vice Chancellor **Dr. Anand A. Samuel**, Pro-Vice Chancellors **Dr. S. Narayanan** and **Dr. V. Raju** for providing necessary resources.

I wish to express my heart-felt gratitude to the Doctoral Committee members, **Dr.Dhavachelvan & Dr.S.Venkatesan**.

I wish to place on record my gratitude to **Dr.Buvaneswari**, Dean (Academic Research), VIT for her kind words of support and encouragement during my research work.

My sincere gratitude to **Dr. Aswani Kumar Cherukuri**, Dean, School of Information Technology and Engineering for giving me the opportunity to undertake the research work.

I thank the management of VIT for permitting me to use the library resources. I also thank all the faculty members of VIT for giving me the courage and strength I needed to complete my goals. This acknowledgement would be incomplete without expressing my whole hearted thanks to my family and friends who motivated me during the course of the work.

Finally, and most importantly, I would like to thank the Almighty God, for it is under his grace that I live, learn and flourish

Place : Vellore

Date :

MANIKANDAN N

CONTENTS

ABSTRACT	i
LIST OF FIGURES	ii
LIST OF TABLES	iv
LIST OF ACRONYMS	v
1 Introduction	1
1.1 MULTI CORE PROCESSOR	1
1.2 VECTORIZATION	2
1.3 ARCHITECTURE SCHEMES AND PARALLEL ALGORITHMS	4
1.3.1 ARCHITECTURE SCHEMES	4
1.3.2 PARALLEL ALGORITHMS	5
1.3.3 MODEL OF COMPUTATION	6
1.4 ALGORITHM EFFICIENCY	8
1.5 PARALLEL COMPUTING MODELS	11
1.6 DATA PARALLEL PROGRAMMING AND NEED FOR VECTORIZATION	13
1.7 PROS AND CONS OF SIMD	17
1.8 VECTOR ARCHITECTURES	19
1.8.1 GPGPU	19
1.8.2 INTEL PROCESSOR VECTORIZATION	20
1.8.3 CPU CORES PARALLELIZING TECHNIQUES	21
1.8.4 GPU ACCELERATION TECHNIQUE	22
1.8.5 CPU AND GPU CORES ACCELERATING TECHNIQUE	22
1.9 INTRODUCTION TO THE APPLICATIONS DEVELOPED	23
1.9.1 DEPENDENCY ANALYSIS	24
1.9.2 PARALLEL QUERY PROCESSING	26
1.9.3 PARALLEL ARTIFICIAL NEURAL	27

	NETWORK DESIGN	
1.9.4	PARALLEL DECISION MAKING SYSTEM FOR ICU DATA	28
1.9.5	PARALLEL AES ALGORITHM DESIGN	30
1.9.6	VECTORIZATION OF DATA COMPRESSION ALGORITHMS	32
1.9.6.1	Branch Instruction Prediction	33
1.9.6.2	Multiple Branch Prediction	33
2	Literature Survey	35
2.1	VECTORIZATION IN MEDICAL DATA PROCESSING	35
2.2	VECTORIZATION IN DATA COMPRESSION	39
2.3	VECTORIZATION IN AES ALGORITHM APPLIED TO IOT	41
2.4	PARALLEL ARTIFICIAL NEURAL NETWORK FOR TIME SERIOUS PREDICTION	46
3	Dependency Analysis And Parallel Algorithm Design	52
3.1	DYNAMIC DEPENDENCY ANALYSIS	52
3.1.1	SETTING UP SIMD ENVIRONMENT FOR VECTORIZATION	52
3.1.2	ANALYSING CODE FOR MANUAL VECTORIZATION	53
3.1.3	APPLYING DEPENDENCY ANALYSIS	54
3.1.4	QUERY PROCESSING USING IMPROVED OPENCL BASED MULTI CORE PROCESSOR LEVEL VECTORIZATION	55
3.2	OPENCL AND VECTORIZATION	56
3.3	PARALLEL QUERY PROCESSING	57
3.4	RESULT AND DISCUSSION	58
3.4.1	SPEEDUP	61
3.4.2	EFFICIENCY	61
3.4.3	COST	62
3.4.4	SCALABILITY	63

4	Parallel Processing based Patient Monitoring System	64
4.1	MEDICAL DATA COLLECTION	64
4.1.1	CREATION OF PATIENT DATABASE	64
4.2	PERFORMANCE MANAGEMENT	66
4.2.1	OPTIMIZATION SCHEMES	66
4.3	HANDLING DEPENDENCY	69
4.4	IoT IMPLEMENTATION	71
4.4.1	SAMPLE DATA SETS	72
4.4.2	PERFORMANCE BENCHMARKS	74
5	Vectorization Based Software Design For Time Serious Prediction Systems	76
5.1	PARALLEL ANN ALGORITHM DESIGN	76
5.1.1	SYSTEM DESCRIPTION	76
5.2	RESULTS AND DISCUSSION	84
5.2.1	DATA COLLECTION	85
5.2.2	CONSTRUCTION OF PERFORMANCE METRICS	85
5.3	SIMULATION RESULTS	86
5.4	PARALLEL IMPLEMENTATION OF SAME ALGORITHM	90
6	Multi-Level Vectorization for Data Compression	92
6.1	SYSTEM OVERVIEW	92
6.1.1	IMPLEMENTATION OF COMPRESSION ALGORITHMS	93
6.2	IMPLEMENTATION OF MULTI-LEVEL VECTORIZATION	97
7	Parallel AES Algorithm for Performance Improvement in Data Analytics Security	102
7.1	CONSTRUCTION OF COMPUTATIONAL MODEL	102
7.1.1	SECURITY MECHANISM THROUGH IoT	103
7.2	PARALLEL AES ALGORITHM	104
7.2.1	DATA LEVEL PARALLELISM	105

7.2.2	ALGORITHM DESIGN	106
7.3	PROCESS DESCRIPTION	108
7.4	PARALLEL IMPLEMENTATION	109
7.4.1	CHALLENGES AND ASSUMPTIONS	109
7.4.2	PSEUDO CODE FOR MATLAB IMPLEMENTATION	111
7.5	EXPERIMENTAL RESULTS	112
8	Conclusion and Future Work	117
8.1	CONCLUSION	117
8.2	FUTURE WORK	119
	References	120

ABSTRACT

In the evolution of modern processors, multi core based CPUs play a vital role. The current world of computing is getting more and more powered by latest processors. Multimedia and Big Data based applications frequently need series of calculations and features for supporting parallelism. With the help of SIMD functions and instruction set, the research can enhance the speed of the multi core based processing. Tools and techniques used for exploiting SIMD parallelism will have high impact results. Now commercial compilers are coming with automatic vectorization of programs but code performance of such compilers are not common for all type of applications. Compiler level vectorization has some difficulties like pointer reference, data and control dependency. These problems can be addressed only by parallel programmers. There are some parallel programming tools like OpenMP, OpenCL, CUDA, MATLAB and thread based programming which supports vectorization. This thesis has developed a detailed analysis and implementation on vectorization methods for multi core processing. This thesis includes dependency analysis methods, choosing correct parallel algorithm, parallel execution tools and manipulation of data. Implementations focus on remodelling the existing applications and algorithms that act as the performance booster with the support of code analyser, parallel algorithm and tool analyser. To test the performance of methodologies, computer intensive applications like matrix computing, big data processing, data compression and encryption algorithms are used. The results are obtained with Intel based multi core processors. The same work is implemented and compared with multicore processors and graphics processing unit combination. As a result of vectorising different applications with varying scale, minimum of 3X and maximum of 140X performance improvement has been achieved.

LIST OF FIGURES

1.1	SIMD Computers	7
1.2	Master/Slave Task Assignment	12
3.1	Parallel Query Processing Architecture	56
3.2	Report Generation Process	58
3.3	OpenCL Execution Vs other methods	60
3.4	Execution Time of Scalar and Vectorized Queries	60
3.5	Parallel Speedup Evaluation	61
3.6	Efficiency of the System	62
3.7	Cost Evaluation of the Proposed System	62
3.8	Performance Analysis of Processor Scaling	63
4.1	Data Collection and Processing System	65
4.2	First Level Decision Tree With Parallel Search	67
4.3	IoT Implementation	71
4.4	Parallel Vs Normal Query Execution	74
4.5	Performance Without Parallelisation	74
4.6	Performance With Parallelisation	75
5.1	First Level Design Decisions	76
5.2	Design for Input Data Generation	77
5.3	Design of Output Prediction Using NARX	78
5.4	Dependency Network	79
5.5	Hybrid Architectural Designs for Training	82
5.6	Implementation Results of Predictions	89
5.7	Performance Improvement Through Parallel Execution	90
6.1	Multi Level Vctorization of SIMD and SIMT	92
6.2	Performance Improvement of Static Parallelism in Two Dimensional DCT	96
6.3	ILP Speedup of Multi-Level Vectorization	100
6.4	Performance Comparison	100
7.1	Requirements for IoT Based Applications	103
7.2	Constraint Based Encryption for IoT	104

7.3	Parallellizing AES Algorithm	105
7.4	Parallel AES Execution in GPU	109
7.5	Comparison Graph Between core i5 and GPU With Basic Optimization	115

LIST OF TABLES

1.1	GPU Configuration	23
2.1	Research Gaps in Surveys made	31
2.2	Comparative Analysis on IoT and AES Algorithm Implementations	45
3.1	Performance of OpenCL	59
4.1	Dependency in Data Management	69
4.2	Handling Control Dependence	70
4.3	Sample Blood Record Stored for Patients	73
5.1	Quality Attributes of Software Design	84
5.2	Data Collected for Analysis	85
5.3	Measurement of Prediction Performance Over 60 Day Prediction	87
6.1	MATLAB Implementation Results Without Parallelism-1D DCT	94
6.2	Implementation Results in MATLAB without Parallelism-2D DCT	95
6.3	Implementation Results without Parallelism- Wavelet transform	97
7.1	Feature Selection in Intel Processors	112
7.2	Serial Code Execution Times	113
7.3	Parallel Code Execution Times	113
8.1	Summary of research	118

LIST OF ACRONYMS

AES	Advanced Encryption Standard
ALU	Arithmetic and Logical Unit
ANN	Artificial Neural Network
CCUMA	Cache Coherent UMA
CDB	Common Data Bus
CPU	Central Processing Unit
CUDA	Compute Unified Device
DCT	Discrete Cosine Transform
DES	Data Encryption Standard
DSP	Digital Signal Processing
EBPN	Error Back Propagation Network
GPGPU	General Purpose Graphics Processing Units
GPU	Graphics Processing Units
ICU	Intensive Care Unit
ILP	Instruction Level Parallelism
IoT	Internet of Things
IPC	Instructions per cycle
MEDUSA	Medical Distributed Utilization of Services and Applications
MIMD	Multiple Instruction stream and Multiple Data stream
MISD	Multiple Instruction stream and Single Data stream
NUMA	Non Uniform Memory Access
OLAP	Online Analytical Processing
PANN	Parallel Artificial Neural Networks
PRAM	Programmable Random Access Memory
SIMD	Single Instruction stream and Multiple Data stream
SIMT	Single Instruction Multiple Threads
SISD	Single Instruction stream and Single Data stream
SMP	Symmetric Multiprocessor
UMA	Uniform Memory Access
IOT	Internet of Things