Optimal Reconfigurable HW/SW Co-design of Load Flow and Optimal Power Flow Computation [1]

M. Murach, *Member*, *IEEE,* P. Vachranukunkiet, *Member*, *IEEE*, P. Nagvajara, *Member*, *IEEE*,
J. Johnson, *Member*, *IEEE*, and C. Nwankpa, *Senior Member*, *IEEE*

Abstract – Load flow and Optimal Power Flow (OPF) constitute core computations used in energy market operation. We considered different design partitions of computational tasks for a desktop computer equipped with Field Programmable Gate Array (FPGA). Load flow and OPF require Lower-Upper triangular matrix decomposition (LU). The number of clock cycles required for data transfer and floating-point operations were used as performance measures in determining optimal hardware/software partitions for each problem. Optimal partition performance is achieved by assigning the Lower-Upper triangular decomposition (LU) and matrix multiplication operations to custom hardware cores. A comparison between the proposed partition and software implemented using a state-of-the-art sparse matrix package running on a 3.2 GHz Pentium 4 shows a six-fold speedup.

Index Terms **– FPGA, optimal power flow, sparse, LU, linear programming, load flow**

I. INTRODUCTION

The energy management system (EMS) is an industrial-strength software package used in the day-to-day operation of the power grid. The primary goals of EMS are to minimize the cost of power generation for a given set of loads and to ensure stable system operation. This paper examines two aspects of EMS namely load flow and optimal power flow. Load flow solves the steady state powers and voltages for the transmission system. Load flow uses the Newton Raphson method to iteratively solve a set of non-linear equations. The evaluation of these sparse matrices accounts for 85% of the execution time [1]. Load flow provides a viable operating point that can be used as a starting point for other EMS applications. Optimal power flow (OPF) aims at providing the lowest cost of operation given set of system constraints. Generally, the primal-dual interior point method (PDIPM) is used since it demonstrates superior performance in systems with more than a thousand variables [2]. The PDIPM also makes extensive use of large-sparse matrix operations.

In this paper we propose to use field programmable gate array (FPGA) linear algebra cores to reduce the overall time needed to solve the sparse linear systems associated with both load flow and OPF. FPGA technology allows the tailoring of these applications to achieve a higher utilization of the floating point unit (FPU) [1]. FPGAs achieve this by allowing the user to implement deep arithmetic pipelines as well as custom memory management to process and access data. Lower-upper triangular decomposition (LU) designed with these FPUs provides a significant speedup on solving these sparse linear systems. Faster evaluation of load flow and OPF can provide both a smaller window of computation and allow larger power grids to be analyzed.

[1] The research reported in this paper was supported by the United States Department of Energy (DOE) under Grant No. CH11171. The authors are also grateful to PJM Interconnection for data provided.

II. PREVIOUS WORKS

A desktop/cluster computing environment is typically used to run the load flow and EMS operations. Our earlier findings indicate that general purpose processing platforms fail to fully utilize floating-point pipeline architecture. From studies conducted using UMFPACK [10], a high performance LU solver package, on Jacobian matrices taken from systems ranging from 118 to 26829 buses, the sparse LU solver could achieve only 1 to 4% of the advertised peak-performances of the floating-point arithmetic units [1], [4].

Fleuck et al. attempted to use cluster computing to quickly evaluate LU matrices [3]. The performance was hindered however, due to load imbalance, fill-ins (due to non-optimal ordering), and high latency in communications. The speedup obtained in the study saturated on 8 processors achieving only a speedup of 4 to 5 on large test cases. Smaller systems will experience much less performance gain since these systems will have smaller granularity.

III. PROBLEM FORMULATION

A. Load Flow

Power flow solution via Newton method involves iterating the following equation [chapter 10 of 5]:

$$-J \cdot \Delta x = f(x) \qquad (1)$$

Until f(x) = 0 is satisfied. The Jacobian, J, of the power system is a large highly sparse matrix (very few non-zero entries), which while not symmetric has a symmetric pattern of non-zero elements. Δx is a vector of the change in the voltage magnitude and phase angle for the current iteration. And f(x) is a vector representing the real and reactive power mismatch. The above set of equations are of the form Ax = B, which can be solved using a direct linear solver. Direct linear solvers perform Gaussian Elimination, which is performed by decomposing the matrix A into Lower (L) and Upper (U) triangular factors, followed by forward and backward elimination to solve for the unknowns.

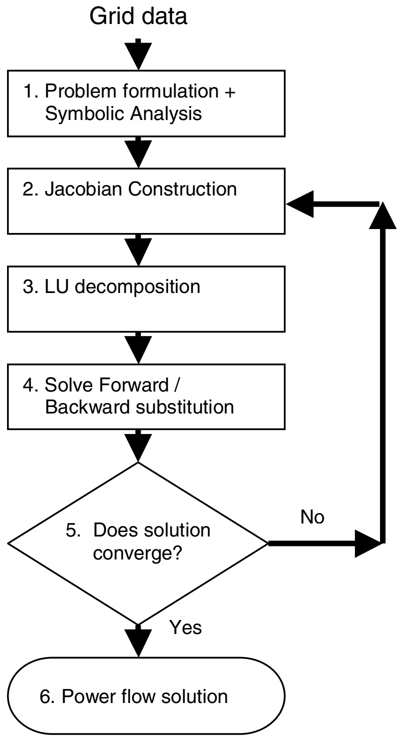

Fig. 1. Load Flow Computation

Fig. 1 shows the process used in computing load flow. This process generally repeats 4 to 10 times until a tolerance (usually 10^{-4} pu) is satisfied.

B. Optimal Power Flow

OPF can be broken into two stages, solving the base case and running contingency analysis. The base solution determines the best generator dispatch for normal system operation. The contingency analysis ensures that the system will continue to operate under all single outage conditions. This is often referred to as Security Constrained OPF (SCOPF). A flow chart of the DC OPF is shown below in Fig. 2. If an optimal dispatch is invalid, a contingency causes additional outages; the generation constraints must be modified to ensure that the additional outages will not occur. The base case is then recomputed and the process is repeated until the contingencies of interest are satisfied.

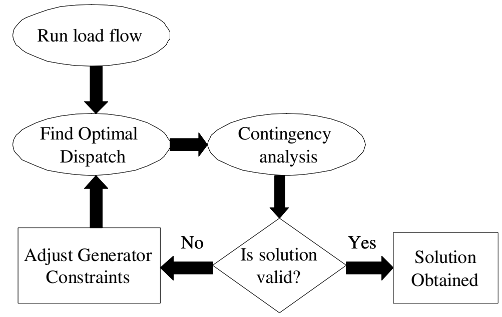

Fig. 2. Optimal Power Flow Computation

Solving the base case in DC optimal power flow (OPF) is generally accomplished by using linear programming (LP) techniques. The primal-dual interior point method (PDIPM) was chosen since it almost exclusively utilizes matrix operations. The base case problem for OPF can be constructed as follows:

List of terms for equations 2 through 5

c: Cost of power per MW at bus *i*
P_{gi}: Power dispatched by generator *i*;
S_{ki}: Shift factor relating the flow in branch *k* to generator *i*
T_k: Active load through branch *k*
P_k,: Flow in branch k due to load profile;
L_i: Load at bus *i*

Objective:

$$\min \sum_{i=1}^{M} \left(c_i * P_i \right)$$

S.T.

$$\sum_{i=1}^{M} \left(S_{ki} * P_i \right) + T_k = P_k \quad (2)$$

$$T_k^{\min} \le T_k \le T_k^{\max} \quad (3)$$

$$P_i^{\min} \le P_i \le P_i^{\max} \quad (4)$$

$$\sum_{i=1}^{M} P_i - \sum_{i=1}^{M} L_i - losses = 0 \quad (5)$$

The objective function is to minimize the cost of running the system. The constraints for this problem are given by (2)-(5). The first constraint represents the line flow constraints, which arise from the admittances of the transmission lines. The second set of constraints represents transmission limitations on lines due to physical limitations such as thermal capacity. Thermal capacity is defined as the maximum allowable load that a line can deliver without undergoing permanent plastic deformation. The third set of constraints defines limits at which each generator can operate. The last constraint is based on the conservation of power in the system. The net flow injected must equal the sum of the power withdrawn and net losses in the transmission system.

IV. BENCHMARK SYSTEMS

A detailed analysis of the systems is provided below. The 118 and 298 cases were provided by the University of Washington's website on test cases: (http://www.ee.washington.edu/research/pstca/). The 1648 and 7917 case files were obtained from the PSS/E data collection [9]. The properties of the following systems were analyzed to obtain data orientation. This information can then be exploited to achieve greater performance. Table I summarizes the number of branches and number of non-zeros (NNZ) in the admittance matrix (Ybus) and resulting Jacobian for the following cases. Formulation for the Ybus is given in chapter 9 of [5].

TABLE I
SUMMARY OF POWER SYSTEM MATRICES

System	Branches	NNZ YBUS	NNZ Jacobian	Jacobian Size
118 Bus	186	490	1,065	181
298 Bus	410	1,118	3,732	526
1648 Bus	2,602	6,852	21,196	2,982
7917 Bus	13,014	33,945	105,522	14,508
10279 Bus	14,571	37,755	134,621	19,285
26829 Bus	38,238	99,225	351,200	50,092

The important result obtained in Table I is the linear growth rate of the number of non-zero elements in the Ybus and resulting Jacobian matrices. This justifies the use of sparse solvers and iterative techniques for solving load flow. Sparse methods will grow linearly with system size whereas direct methods using the Z-bus will grow exponentially with system size and quickly become impractical for large systems chapter 7 of [6].

V. PARTITION ANALYSIS

To assess the potential for hardware acceleration, the computational demands associated with each stage of the algorithm must be taken into account. The impact on the data transfer, memory hierarchy, and arithmetical operation counts must be considered.

The bulk of the computation time for the load flow computation is dedicated for LU decomposition, which lends itself to hardware applications as depicted in Fig. 3. The hardware model can be further simplified since forward and backward substitution can be overlapped using the host computer. Jacobian construction, symbolic computation, and iteration updates account for a relatively small portion of the execution time [1].

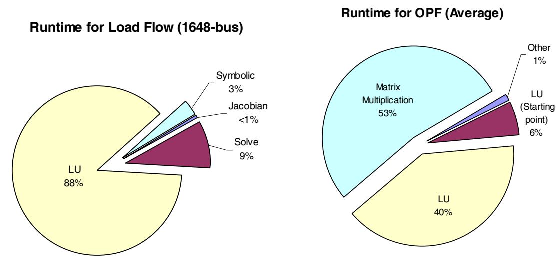

Fig. 3. Percentage Runtime for Load Flow (1648-bus) and Percentage Runtime for OPF (Average)

The potential for hardware acceleration in OPF is somewhat different. Significant computation time is spent on matrix manipulations, which are used in the creation of a fully determined system. However, LU factorization is still used extensively in the OPF evaluation both for starting point generation and for the iterative scheme presented in [2]. Using the data gathered from Fig. 3, a valid partitioning scheme can be chosen that effectively utilizes FGPA resources. LU and matrix multiplication operations are good candidates for hardware implementation since they are comprised mostly of floating point arithmetic. Other operations such as symbolic computation and the Jacobian construction fail to justify hardware implementation. FPGA size limitations, development time required, and low return on investment deters further work into implementation of non-matrix operations on FPGA. For these operations, general processors are ideal.

VI. PERFORMANCE MODEL

Several performance models were constructed to assess the potential speedup over software only applications (SW) considering factors such as partitioning and communication overhead. The impact of memory latency and PCI bus communications must be considered. Fig. 4 shows the LU hardware's

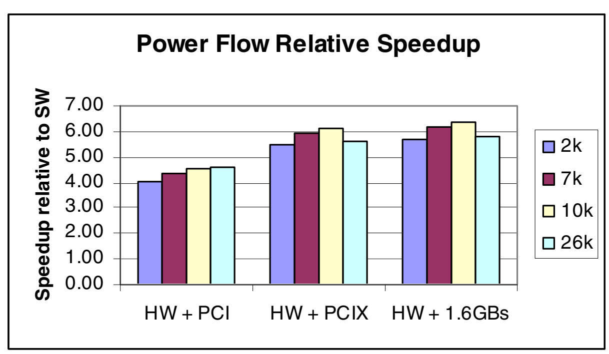

Fig. 4. Load Flow Speedup over software including communication costs

speedup for power flow applications utilizing different PCI bus architectures. Most notably, it can be seen that a PCI-X bus provides necessary bandwidth for our current implementation. Further decreases in communication costs do not significantly improve the overall performance. Next, results were scaled with increasing amounts of FPGA arithmetic units to estimate the optimum amount of logic necessary to achieve a satisfactory speedup.

Fig. 5 shows the predicted performance ratio of performance of the LU hardware at 200 MHz to the Pentium 4 benchmark system. This data predicts a speedup over a Pentium 4 processor by a factor of 10 with only 4 update units and approximately 15 with 16 units. Update units refers to the number of rows that can be reduced concurrently with respect to the current pivot selection. The amount of parallelism that can be achieved is limited by the average number of row updates needed.

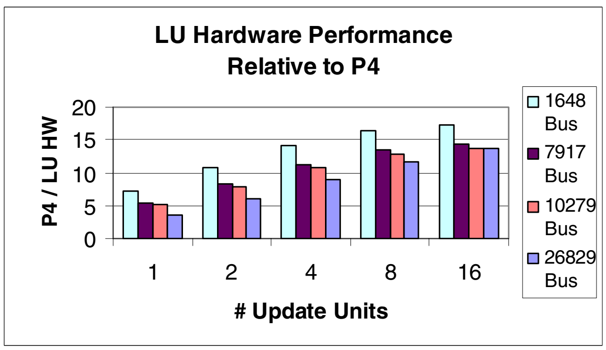

Fig. 5. Load Flow Speedup including communication

A C program was developed in order to evaluate the number of iterations, floating point operations, and projected hardware performance on LU decomposition. Validity of the linear solver portion was checked using lp_solve [9], an open source package that uses the simplex method. Fig. 6. shows the ratio of a Pentium 4 benchmark system to LU hardware on an ADA^T matrix used in DC OPF for the three largest systems. A is a rectangular matrix constructed from (2)-(5). D is a diagonal matrix comprised of the current solution. More detail on the primal dual interior point method can be found in [7].

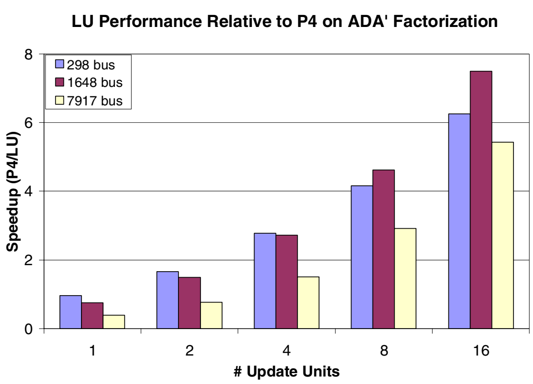

Fig. 6. LU Hardware Performance Ratio (OPF)

This data predicts that speedup of 2.5 is achievable with 4 update units and a speedup of 6 is possible with 16 units. The smaller systems do not saturate as before since the matrices used in LP are less sparse than those used in load flow. This allows more arithmetic units to be used since significantly more computations are needed. The results indicate that the LU hardware is useful for smaller systems in linear programming, since these matrices are considerably less sparse than the Jacobian matrices used in load flow.

A combined performance model for the largest solved system is provided in Fig. 7. This indicates the net speedup that is achievable with the 1648-bus case assuming an 85% achievable enhancement in both the load flow and base case stages (LP solver). No speedup is taken from the contingency analysis. The percentage of time given for each stage given in Fig. 3 is then used with the relative speedup in each stage. The trend given in Fig. 7 indicates that the speedup is logarithmically dependent on the number of update units. The results indicate that a speedup of 5 is possible using an FPGA with four update units.

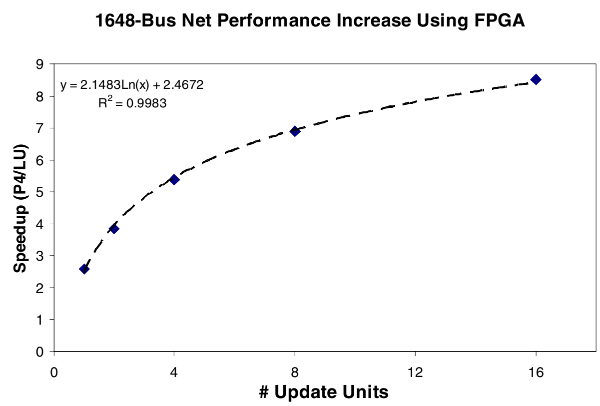

Fig. 7. Net Speedup for OPF on 1648-Bus Case

VII. CONCLUSION

A system-on-chip design is suitable for EMS since the large sparse matrices computation constitutes a net speedup of at least 5 utilizing a 200 MHz hardware LU with four parallel update units versus a standard desktop computer. Results that have been obtained indicate that a 10x performance is achievable over existing state-of-the-art sparse linear solvers, for the LU decompositions used in load flow calculations. This speedup is achieved by employing custom hardware to reduce pivot search times. Custom hardware also provides data structures for sparse matrices that allow the use of fine-grained parallelism, updating many rows in parallel. This reduction in overhead greatly increases the utilization of floating point units compared to software only solutions on general-purpose architectures. Further work will aim at the development of other sparse matrix operations to further speedup power computations. Sparse matrix-

matrix multiplication, matrix-vector multiplication, and QR factorization (used in state estimation) will benefit from the use of FPGA technologies.

VIII. REFERENCES

[1] P. Vachranukunkiet, J. Johnson, P. Nagvajara, S. Tiwari, and C. Nwankpa "Performance Analysis of Load Flow on FPGA", 15th Power Systems Computational Conference August 22, 2005.

[2] M. Murach, P. Nagvajara, J. Johnson, C. Nwankpa "Optimal Power Flow Utilizing FPGA Technology", 37th North American Power Symposium (NAPS) 2005.

[3] Feng Tu and A. J. Flueck. "A Message-Passing Distributed-Memory Parallel Powerflow Algorithm", Power Engineering Society Winter Meeting, 2002. IEEE, Volume 1 (2002), pp 211 – 216.

[4] Davis, T. and Duff, I. "An Unsymmetric-pattern Multifrontal method for Sparse LU Factorization", SIAM Journal on Matrix Analysis Applications., 18 (1997), pp. 140-158.

[5] A. Bergen, V. Vittal, Power Systems Analysis, 2nd Edition, Prentice Hall, 2000.

[6] Grainger, J.J. and Stevenson, W.D., Jr., Power System Analysis, Mcgraw-Hill, New York, 1994.

[7] G. Astfalk et al. "The Interior-Point Method for Linear Programming", IEEE Software July 1992. pp 61-68.

[8] PSS/E. The Shaw Power Group Inc. 2005. http://www.shawgrp.com

[9] M. Berkelaar, Lp_solve, Available: ftp://ftp.es.ele.tue.nl/pub/lp_solve/lp_solve

[10] T. A. Davis and I. S. Duff, An unsymmetric-pattern multifrontal method for sparse lu factorization, SIAM J. Matrix Anal. Applic., 18 (1997), pp. 140-158.

IX. BIOGRAPHIES

Matthew Murach is currently a BS/MS student in Biomedical Engineering and Computer Engineering at Drexel University. He is currently a teaching assistant for Drexel University in the department of Electrical and Computer Engineering. His research interests include embedded microcontrollers, reconfigurable hardware design, and optimizations research.

Petya Vachranukunkiet received a BS in Microelectronic Engineering from Rochester Institute of Technology in 1999, MS in Electrical Engineering from Drexel University in 2003. He is currently a PhD candidate and teaching assistant at Drexel University in the department of Computer Engineering. His current research interests include reconfigurable hardware design, and computer architecture.

Prawat Nagvajara is a Associate Professor in Electrical and Computer Engineering at Drexel University. He received his Ph.D. in Computer Engineering form Boston University. His research interests include high-performance embedded computing and information assurance. He is co-director, with Jeremy Johnson of the Department of Computer Science, of the Application Specific Computing Research Group and is a member of the DOE Powergrid projects.

Jeremy R. Johnson is Professor and Department Head for Computer Science at Drexel University. He received his Ph.D. in Computer and Information Science from the Ohio State University. His research interests include algebraic algorithms, computer algebra systems, problem solving environments, programming languages and compilers, high performance computing, hardware generation, and automated performance tuning. He is co-director, with Prawat Nagvajara of the Department of Electrical and Computer Engineering, of the Application Specific Computing Research Group and is a member of the SPIRAL and Powergrid projects, which are sponsored by NSF, DARPA, and DOE.

Chika O. Nwankpa (SM'87) received the Magistr Diploma in electric power systems from Leningrad Polytechnical Institute, USSR, in 1986, and the Ph.D. degree in the electrical and computer engineering from the Illinois Institute of Technology, Chicago, in 1990. He is currently a Professor of Electrical and Computer Engineering and director of the Center for Electric Power Engineering (CEPE) at Drexel University, Philadelphia, PA. His research interests are in the areas of power systems and power electronics. Dr. Nwankpa is a recipient of the 1994 Presidential Faculty Fellow Award and the 1991 NSF Engineering Research Initiation Award.