

Superpipelined CORDIC Unit to Aid in Power System Analysis Using FPGA Technology

Michael Fitzharris, Servesh Tiwari, Jeremy Johnson, Prawat Nagvajara
Center for Electric Power Engineering (CEPE) <http://power.ece.drexel.edu>

Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA. 19104



Abstract

Some applications, such as power system analysis, require many trigonometric computations. In power system analysis an iterative solver using Newton-Raphson is commonly used to solve the loadflow equations. The Jacobian matrix, which is used each iteration, is constructed using numerous sine and cosine computations. In such applications, substantial speedup can be obtained by utilizing specialized hardware to compute sine and cosine.

Jacobian Matrix

$$J = \begin{bmatrix} \frac{\partial P_1}{\partial \theta_1} & \frac{\partial P_1}{\partial \theta_2} & \dots & \frac{\partial P_1}{\partial \theta_n} \\ \frac{\partial Q_1}{\partial \theta_1} & \frac{\partial Q_1}{\partial \theta_2} & \dots & \frac{\partial Q_1}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_n}{\partial \theta_1} & \frac{\partial P_n}{\partial \theta_2} & \dots & \frac{\partial P_n}{\partial \theta_n} \\ \frac{\partial Q_n}{\partial \theta_1} & \frac{\partial Q_n}{\partial \theta_2} & \dots & \frac{\partial Q_n}{\partial \theta_n} \end{bmatrix}$$

Loadflow Equations

$$P_i = \sum_{j=1}^n V_i V_j [G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j)] \quad i = 1, 2, 3, \dots, n$$

$$Q_i = \sum_{j=1}^n V_i V_j [G_{ij} \sin(\theta_i - \theta_j) - B_{ij} \cos(\theta_i - \theta_j)] \quad i = 1, 2, 3, \dots, n$$

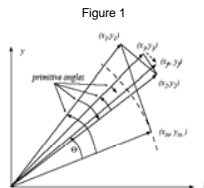
The CORDIC method provides an iterative scheme, consisting of simple addition and binary shift operations, to compute trigonometric values to any desired precision. The iterations in the CORDIC method can be pipelined to devise an efficient hardware unit that is capable of computing one sine and cosine every clock cycle. The number of stages in the pipeline depends on the number of iterations and hence the desired accuracy. This leads to a very long pipeline, which provides efficient computation provided there is a stream of sine and cosine computations. In such cases, an order of magnitude improvement in performance was obtained using CORDIC unit implemented in FPGA (field programmable gate array) compared to a software solution using a Pentium IV processor.

CORDIC Background

A 2-dimensional vector rotation by an angle θ can be achieved with the matrix equation seen to the right which requires four floating point multiplies.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

The same 2-D vector rotation can also occur iteratively with a number of micro-rotations as seen in figure 1.



By choosing the angle of rotation so that $\tan(\theta)$ results in a fractional power of 2, the vector multiplications can be accomplished in custom hardware using simple shift and add operations.

CORDIC Background

Any angle in the first or fourth quadrant can be iteratively constructed as a series of micro-angles with tangent values as powers of 2.

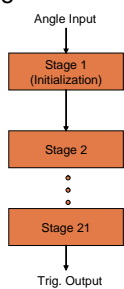
The desired precision can be extended to any degree by simply adding more rotations, and therefore pipe stages.

Each rotation adds a gain to the vectors length, which can be accounted for by initializing the x value to a constant number, depending on the number of stages in the pipe.

Example: 37 degrees can be obtained with the combination seen below.
 $45 - 26.56 + 14.04 + 7.13 - 3.58 + 1.79 - 0.89 = 36.93$

CORDIC Pipeline

- The X vector is initialized to 0.607 to avoid a multiplication at the end of the pipe that would account for the gain.
- Each pipe stage adds a single clock cycle to CORDIC's overall latency, as well as precision to the desired solution.
- The Super-pipeline stretches each stage into 2 clock cycles, which essentially doubles the latency, while still providing a sine and cosine on each clock.



Super-pipelined CORDIC

Allowing the CORDIC processing element to perform both a shift and an add operation in a single clock cycle proves to be costly to the FPGA's maximum frequency.

By stretching the pipeline out, and thus separating the most comprehensive operations, our custom hardware can achieve more parallelism and therefore run at higher frequencies.

Tsunami's Stratix FPGA displays a maximum frequency of 122 MHz when traditional CORDIC is used, and 166 MHz when a super-pipelined version of CORDIC is used, which is approximately a 36% increase in frequency.

Problem Set Analysis

Name	Source	Bus Size	Ymin/NZ	Jac/NZ	Trig Func.	Fixed Func.	Fact/Func.	Mem Size
2k Bus	PSS/E	1646	6952	2232	6952	6534	51422	5010kbits
7k Bus	PSS/E	7917	33945	11316	33945	43136	26237	1.8Mbits

- The two power system data sets that were analyzed are provided by PSS/E.
- Each Jacobian Matrix is currently computed utilizing host software capabilities.
- The values seen above are obtained using various counters throughout the software.
- A hardware implementation would require the memory displayed based on the data structures created by host software.

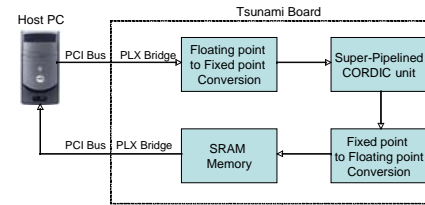
Current Architecture Limitations

The PCI bus is 64 bits wide and is limited to a maximum frequency of 66 MHz which can vary depending on the number of peripherals utilizing the bus.

Data transfers must be multiplexed onto Tsunami's 32 bit wide Avalon system bus using the PLX Bridge which further hinders DMA performance.

The timing analysis proves that for CORDIC to obtain valuable speedup, data and power flow calculations most reside locally to bypass any data transfer limitations.

Current Implementation



- DMA transfers occur between the host PC and FPGA in both directions.
- Software running on the host PC handles all other calculations required for the Jacobian formulation.

Future Implementations

Tsunami Platform

- All power system data can be stored on chip in one of many 64Mbyte single ported SDRAMs to greatly reduce communication expenses.
- Hardware floating point units can be used along with the existing CORDIC to further speed up calculation time.
- Due to the random data structure caused by a power system's network configuration, a very sophisticated cache controller must be developed in order to avoid cache misses.

VFP1 Platform

- The VFP1 board is equipped with Power PC embedded processor support which can easily handle all control required for the Jacobian formulation.
- There is also 2 banks of 64Mbyte dual ported SDRAM which will allow for quicker storage and retrieval of information.
- Although the cache controller would still be a challenge, the increased chip size, and more efficient memories will allow for a better overall solution.



Timing Analysis

	Software(clocks)	Hardware(clocks)	SW Time (msec)	HW Time (msec)	HW/SW(%)
Trig. VS. Trig.	140000	8874	2.1912	0.1042	0.5691
2k Bus	475000	29957	15.1079	0.5147	0.5955
Trig. VS. Trig. + DMA	140000	11100	2.1912	1.0759	0.5891
2k Bus	475000	34800	15.1079	5.2727	0.5771
Jac. VS. Jac.	850000	78000	12.6788	11.6162	0.9176
2k Bus	4400000	406200	67.4242	62.0076	0.9197

Specifications

The analysis seen above was measured using an onboard timer running at 66 MHz.

Observations

- Trigonometric calculations alone display a 20x speedup from Pentium IV software libraries to the custom CORDIC FPGA.
- Once DMA transfers are included, the speedup on trig. calculations reduces to approximately 2x.
- The full Jacobian calculation time is reduced by less than 10% utilizing this architecture.

Future Research

Super-pipelined CORDIC units can be created to handle a large variety of mathematical equations required for power flow analysis.

CORDIC processing elements to calculate arctangent and to rotate any vector in a 2-dimensional space can be combined in order to perform QR factorization for use in matrix inversion, or state estimation.

The square root function can also be achieved iteratively with CORDIC, which can aid in the complex multiplication used in the Gauss-Jacobi method for power flow.

Each method provides for a faster solution to the loadflow problem by achieving higher frequencies and better throughput than both software and other hardware cores.