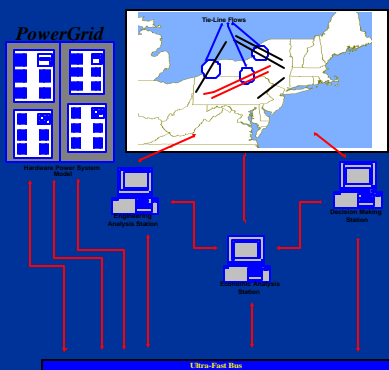


# Linear Algebra Processor using FPGA

Jeremy Johnson, Prawat  
Nagvajara, Chika Nwankpa  
Drexel University



# Goal

- ✍ To design an embedded FPGA-based multiprocessor system to perform high speed Power Flow Analysis.
- ✍ To provide a single desktop environment to solve the entire package of Power Flow Problem (Multiprocessors on the Desktop).
- ✍ Provide a scalable solution to load flow computation.
- ✍ Deliver: Prototype and feasibility analysis.

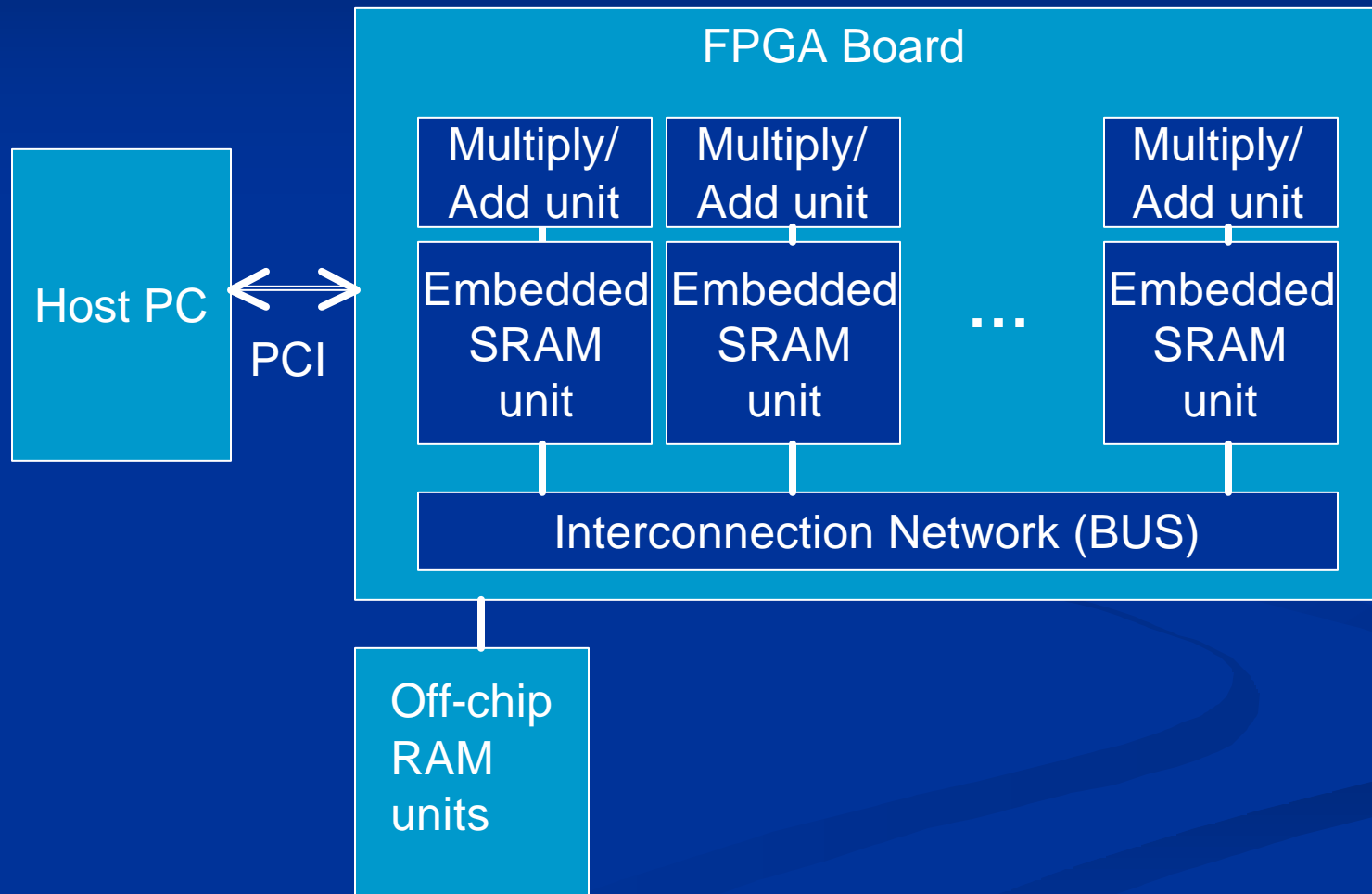
# Approach

- ✍ Utilize parallel algorithms for matrix operations needed in load flow.
- ✍ Utilize sparsity structure across contingencies.
- ✍ Use multiple embedded processors with problem specific instruction and interconnect.
- ✍ Scalable parameterized design.
- ✍ Pipelined solution for contingency analysis.

# Dense Matrix Multiplier

- ✍ Distributed memory implementation
  - ✍ Hardwired control for communication
  - ✍ Parameterized number of processing elements (multiply and accumulate hardware)
  - ✍ Overlap computation and communication
  - ✍ Use block matrix algorithms with calls to FPGA for large problems
- ✍ Processor  $i$  stores  $A_i$ , the  $i$ th block of rows of  $A$  and  $B_j$ , the  $j$ th block of columns of  $B$ 
  - ✍ Compute  $C_{ij} = A_i * B_j$
  - ✍ Rotate: send  $B_j$  to processor  $(j+1) \bmod \text{Number of processors}$

# Processor Architecture



# Performance

## ✍ Performance estimate

- ✍ Xilinx Virtex 2 (XC2V8000)

## ✍ 168 built-in multipliers and on chip memories (SRAM) ? support for 42 single-precision processing elements

- ✍ 4.11 ns pipelined 18 X 18 bit multiplier

- ✍ 2.39 ns memory access time

- ✍ 7.26 ns for multiply accumulate

## Time for $n \times n$ matrix multiply with $p$ processors

- ✍  $7.26n^3/p$  ns

- ✍ 11 ms for  $n=400$ ,  $p = 42$  ? 11,570 MFLOPS

# APPLICATION, ALGORITHM & SCHEDULE

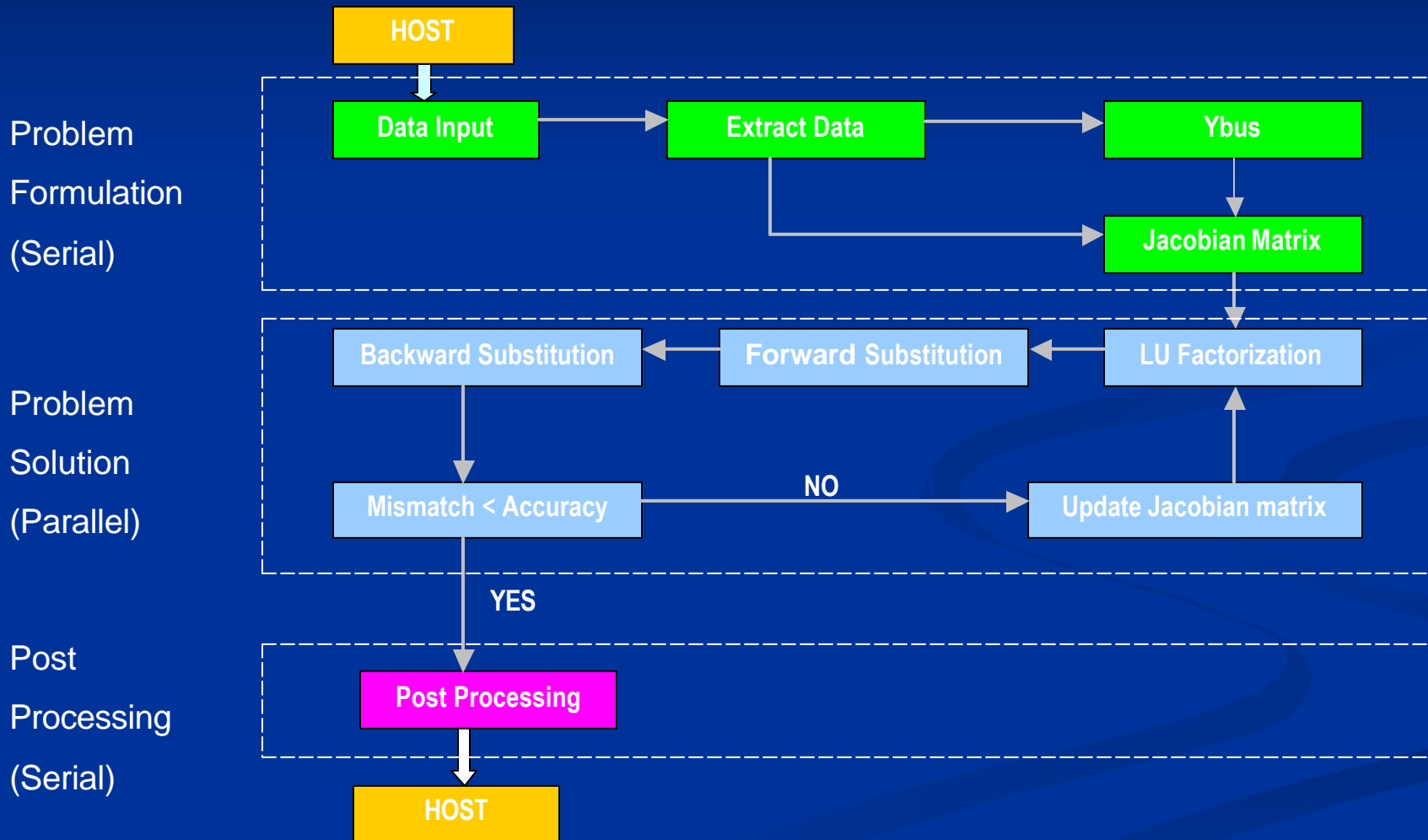
## ✍ Application

- ✍ Linear solver in Power-flow solution for large systems
- ✍ Newton-Raphson loops for convergence, Jacobian matrix

## ✍ Algorithm & Schedule

- ✍ Pre-permute columns
- ✍ Sparse LU factorization, Forward and Backward Substitutions
- ✍ Schedule: Round Robin distribution of rows of Jacobian matrix according to the pattern of column permutation

# Breakdown of Power-Flow Solution Implementation in Hardware

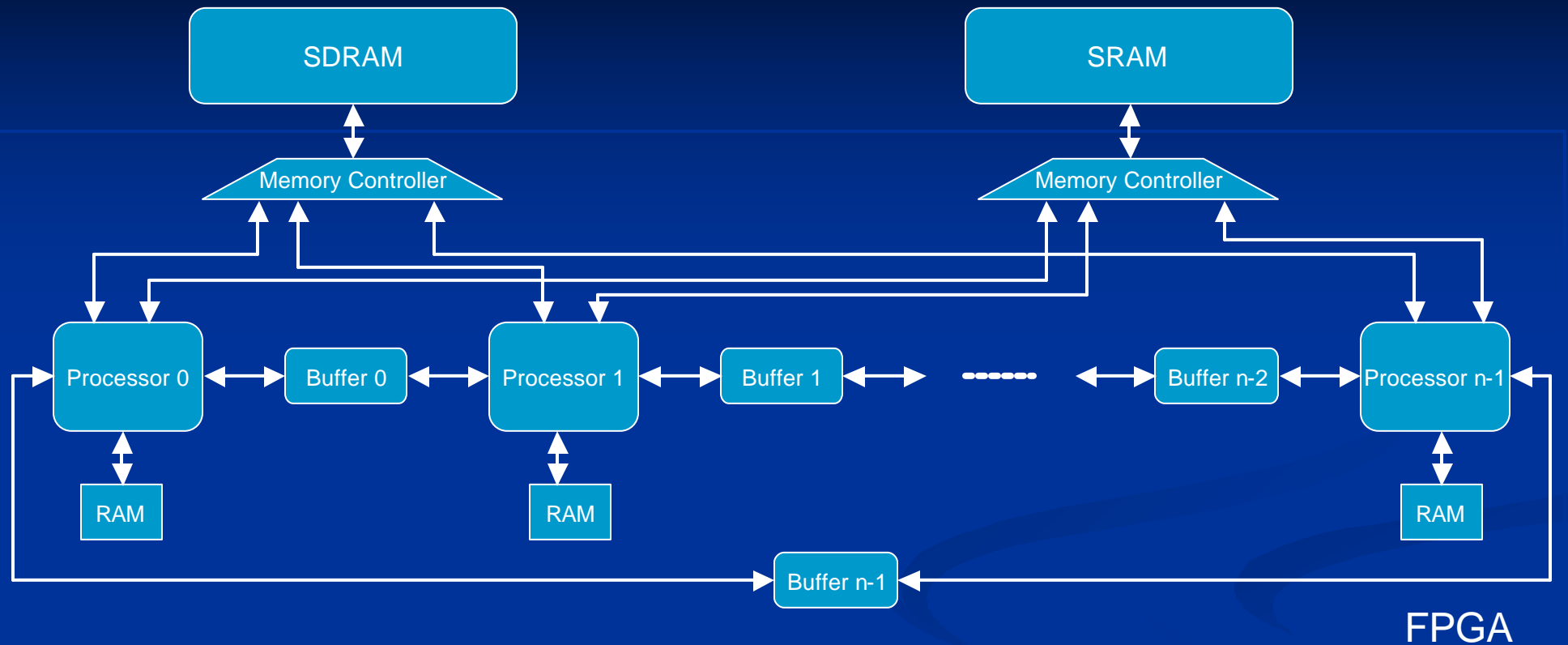


# Minimum-Degree Ordering Algorithm (MMD)

- ✍ Reordering columns to reduce fill-ins while performing LU factorization
- ✍ Reduce floating point operations and storage
- ✍ Compute column permutation pattern once
- ✍ Apply throughout power-flow analysis for that set of input bus data

	Without MMD			With MMD		
	Div	Mult	Sub	Div	Mult	Sub
<b>IEEE 30-bus</b>	887	25806	25806	270	6535	6535
<b>IEEE 118-bus</b>	4509	340095	340095	719	55280	55280
<b>IEEE 300-bus</b>	31596	5429438	5429438	3058	640415	640415

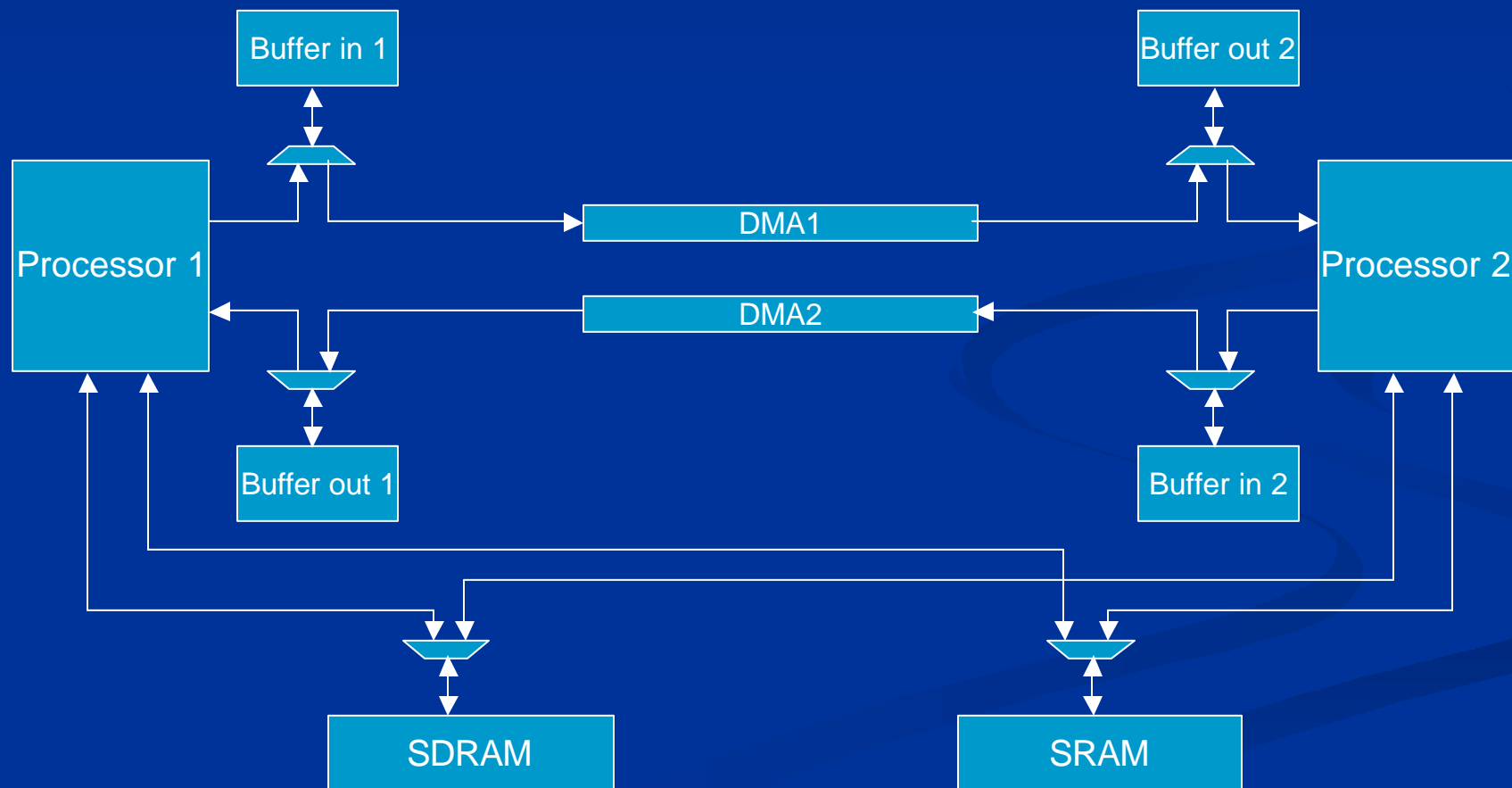
# RING TOPOLOGY

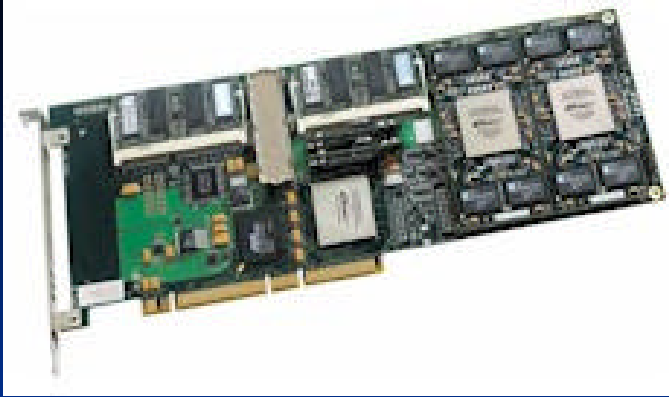


- Nios Embedded Processors
- Buffers are on-chip memories; interprocessor communication

# Hardware Model using Nios Processor

- ✍ Communication: used DMA for passing messages
  - ✍ Buffers are on-chip memories
  - ✍ Trapezoids are arbitrators



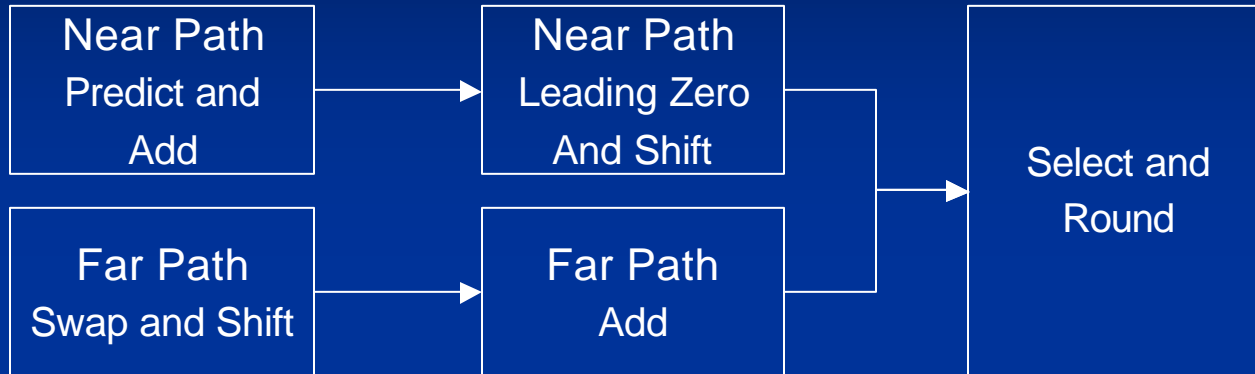


# Stratix FPGA

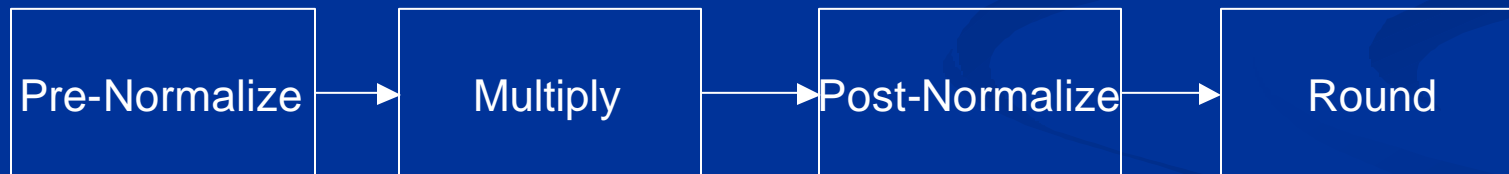
<b>Stratix 1S25F780C6 Resources</b>	
LEs	25,660
M512 RAM blocks (32 x 18 bits)	224
M4K RAM blocks (128 x 36 bits)	138
M-RAM blocks (4K x 144 bits)	2
Total RAM bits	1,944,576
DSP Blocks	10
Embedded multipliers	80
PLLs	6

# Floating Point Unit

## FADD



## FMUL



## FDIV



# Floating Point Unit

## IEEE-754 Support

- Single Precision Format, Round to Nearest
- Round Nearest Rounding, Denormalized Numbers, Special Numbers

	LEs	MULs	Fmax (MHz)	Latency (clk cycles)	Pipeline Rate (clk cycles)
FADD	1088	0	91	3	1
FMUL	926	8	107	4	1
FDIV	1166	8	65	9	N/A
NIOS + FPU	6134		63.88		

# PERFORMANCE ANALYSIS

## ✍ Why?

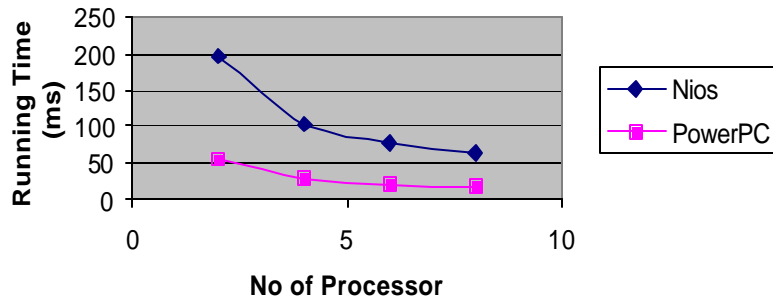
- ✍ Prototype is not intended for industrial performance
- ✍ Show potential performance as a function of available H/W resources

## ✍ Analysis performed for high-performance multi-processor system

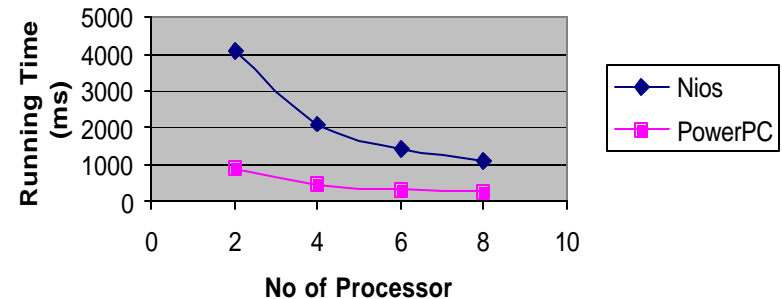
- ✍ Estimate of number of clock cycles and timing
  - ✍ Communication latency
  - ✍ Arithmetic latency
- ✍ Model in MATLAB
- ✍ 80 MHz pipelined Floating-point Unit
- ✍ Variables
  - ✍ Number of Processors (2, 4, 6, 8)
  - ✍ Size of input data (power flow IEEE1648-bus, IEEE7917-bus)
- ✍ System constraints: memory access, size of FPGA chip, system speed

# TIMING OF PERFORMANCE MODEL

Running Time vs No of Processor for matrix size 2982x2982 from IEEE1648-bus



Running Time vs No of Processor for matrix size 14508x14508 from IEEE7917-bus



- Nios embedded processors on Altera Stratix FPGA, running at 80 MHz

✎ 8 processors:

- 1648-bus: 64.179 ms
- 7917-bus: 1106.676 ms

- 400 MHz PowerPC on Xilinx Virtex 2 FPGA with 80 MHz FPU

✎ 8 processors:

- 1648-bus: 18.992 ms
- 7917-bus: 256.382 ms

- WSMP – 1.4 GHz, 256 KB Cache, 256 MB SDRAM, Linux OS

- 1648-bus: 146.435 ms
- 7917-bus: 666.487 ms