USC

Pipelined Data Path for an IEEE-754 64-Bit Floating-Point Jacobi Solver

Gerald R. Morris

University of Southern California

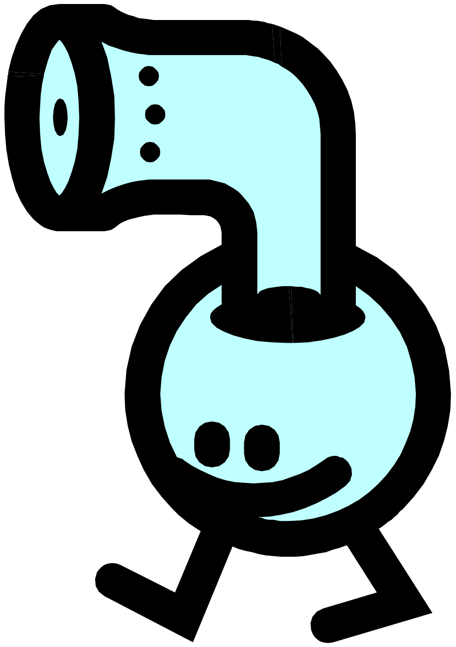
September 20, 2005

USC Viterbi

School of Engineering

Overview

- Iterative Methods
- Jacobi Data Path Design
- Experiments
- Related Work
- Acknowledgments
- Questions



Iterative Methods

- Approach
  - manipulate equation to look like $\mathbf{x} = f(\mathbf{x})$ [1]
  - iteratively compute $\delta+1^{th}$ value as $\mathbf{x}^{(\delta+1)} \Leftarrow f(\mathbf{x}^{(\delta)})$ [2]
- Example: scalar quadratic equation with known solution

$$0 = (x+2)(x-5)$$

$$0 = x^2 - 3x - 10$$

$$x^2 = 3x + 10$$

$$x = \sqrt{3x+10} \qquad ([1])$$

$$x^{(\delta+1)} \Leftarrow \sqrt{3x^{(\delta)}+10} \qquad ([2])$$

δ	$x^{(\delta)}$	$\sqrt{3x^{(\delta)}+10}$
0	1.000	3.606
1	3.606	4.563
2	4.563	4.867
3	4.867	4.960
4	4.960	4.988
5	4.988	4.996
6	4.996	4.999
7	4.999	5.000

Jacobi Iterative Method

- **Jacobi iterative method solves** $A\mathrm{x} = \mathrm{b}$
  - $n \times n$ **diagonally dominant real-valued matrix,** A
  - **the unknown real-valued** n**-vector,** x
  - **constant real-valued** n**-vector,** b
- **Derivation: plug** $A = L + U + D$ **into** $A\mathrm{x} = \mathrm{b} \rightarrow (L + U + D)\mathrm{x} = \mathrm{b}$
  - **lower triangular matrix,** $L = \{a_{ij} \mid i>j \text{ else } 0\}$
  - **upper triangular matrix,** $U = \{a_{ij} \mid i<j \text{ else } 0\}$
  - **diagonal matrix,** $D = \{a_{ij} \mid i=j \text{ else } 0\}$

vector form

$$\boxed{\mathbf{x}^{(\delta+1)} \Leftarrow D^{-1}\left[\mathbf{b} - (L+U)\mathbf{x}^{(\delta)}\right]}$$

point form

$$\boxed{x_i^{(\delta+1)} \Leftarrow \frac{1}{a_{ii}}\left[b_i - \sum_{j=1 \mid j\neq i}^{j=n} a_{ij}x_j^{(\delta)}\right]}$$

Jacobi Iterative Method

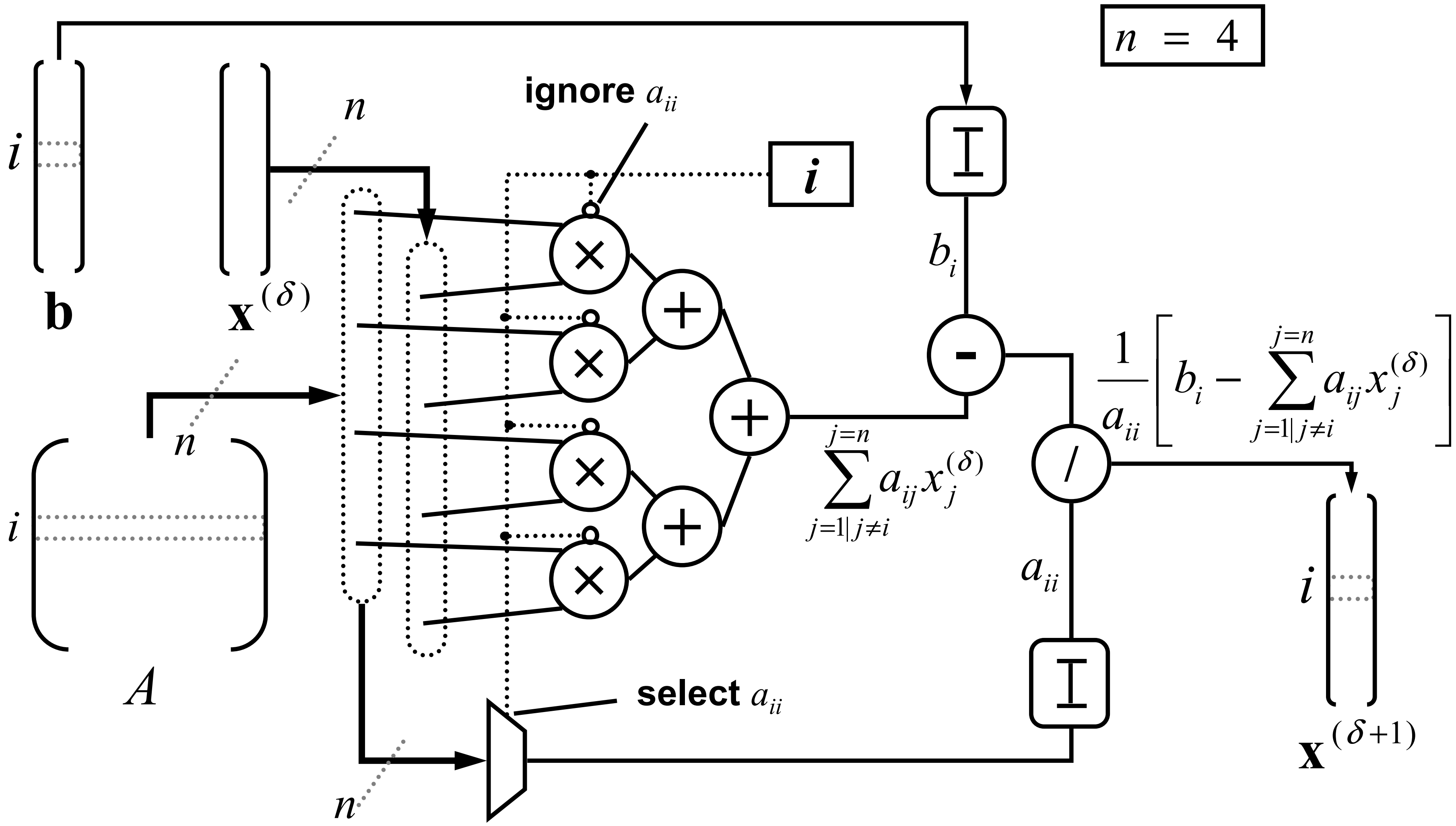
- **Example: 2×2 linear equation with known solution**

$$A\mathbf{x} = \mathbf{b}$$

$$\begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 12 \\ 14 \end{bmatrix}$$

δ	$\left(\mathbf{x}^{(\delta)}\right)^{\mathrm{T}}$		$\left(D^{-1}\left[\mathbf{b}-(L+U)\mathbf{x}^{(\delta)}\right]\right)^{\mathrm{T}}$	
0	1.000	1.000	3.333	3.250
1	3.333	3.250	1.833	2.667
2	1.833	2.667	2.222	3.042
3	2.222	3.042	1.972	2.944
4	1.972	2.944	2.037	3.007
5	2.037	3.007	1.995	2.991
6	1.995	2.991	2.006	3.001
7	2.006	3.001	1.999	2.998
8	1.999	2.998	2.001	3.000
9	2.001	3.000	2.000	3.000

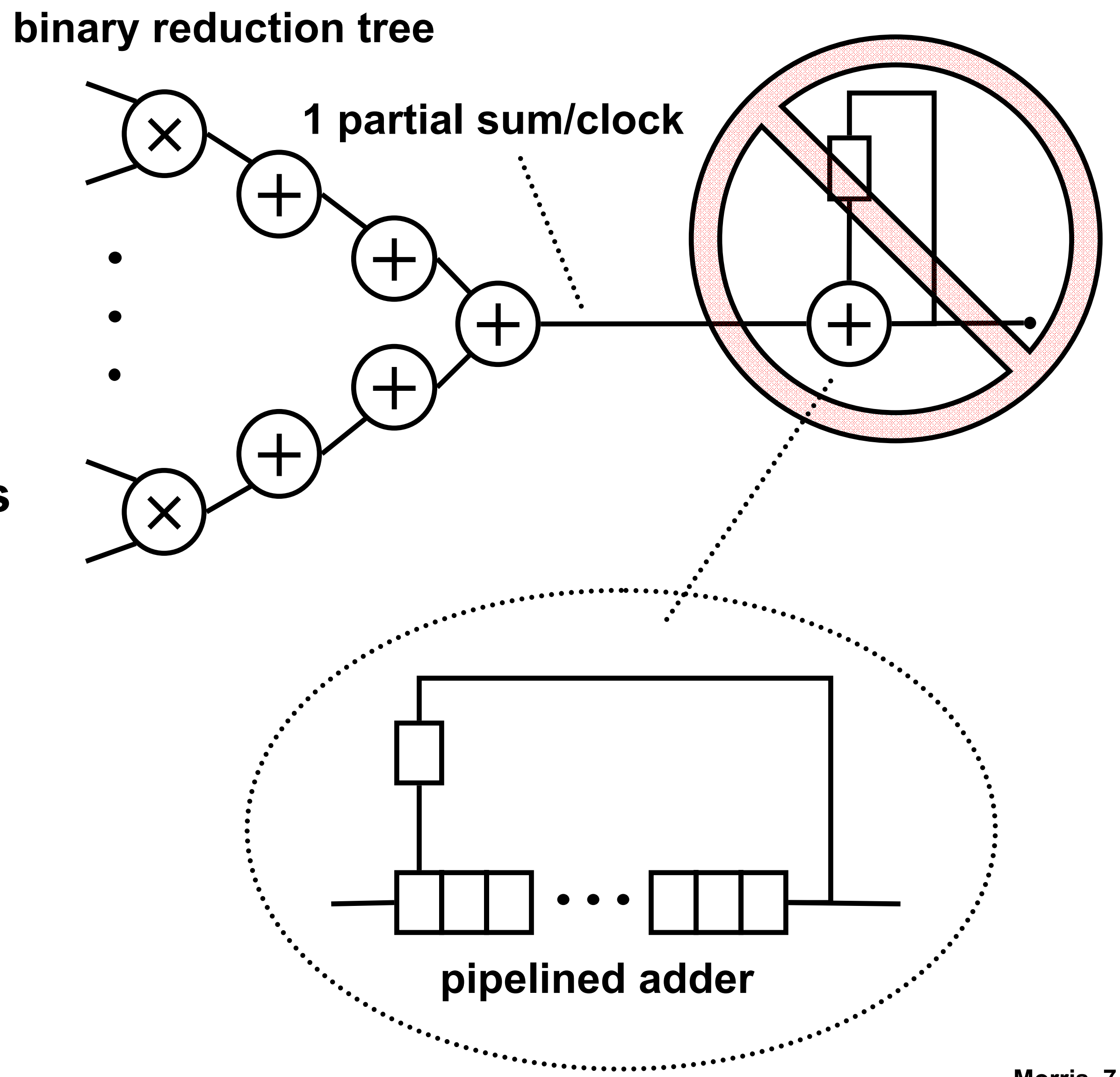
Idealized Jacobi Circuit



$n = 4$
i
$\mathbf{b}$
n
$\mathbf{x}^{(\delta)}$
ignore a_{ii}
i
b_i
$\frac{1}{a_{ii}}\left[b_i - \sum_{j=1 \mid j\neq i}^{j=n} a_{ij}x_j^{(\delta)}\right]$
$\sum_{j=1 \mid j\neq i}^{j=n} a_{ij}x_j^{(\delta)}$
n
i
A
a_{ii}
i
$\mathbf{x}^{(\delta+1)}$
select a_{ii}
n

Design Issues

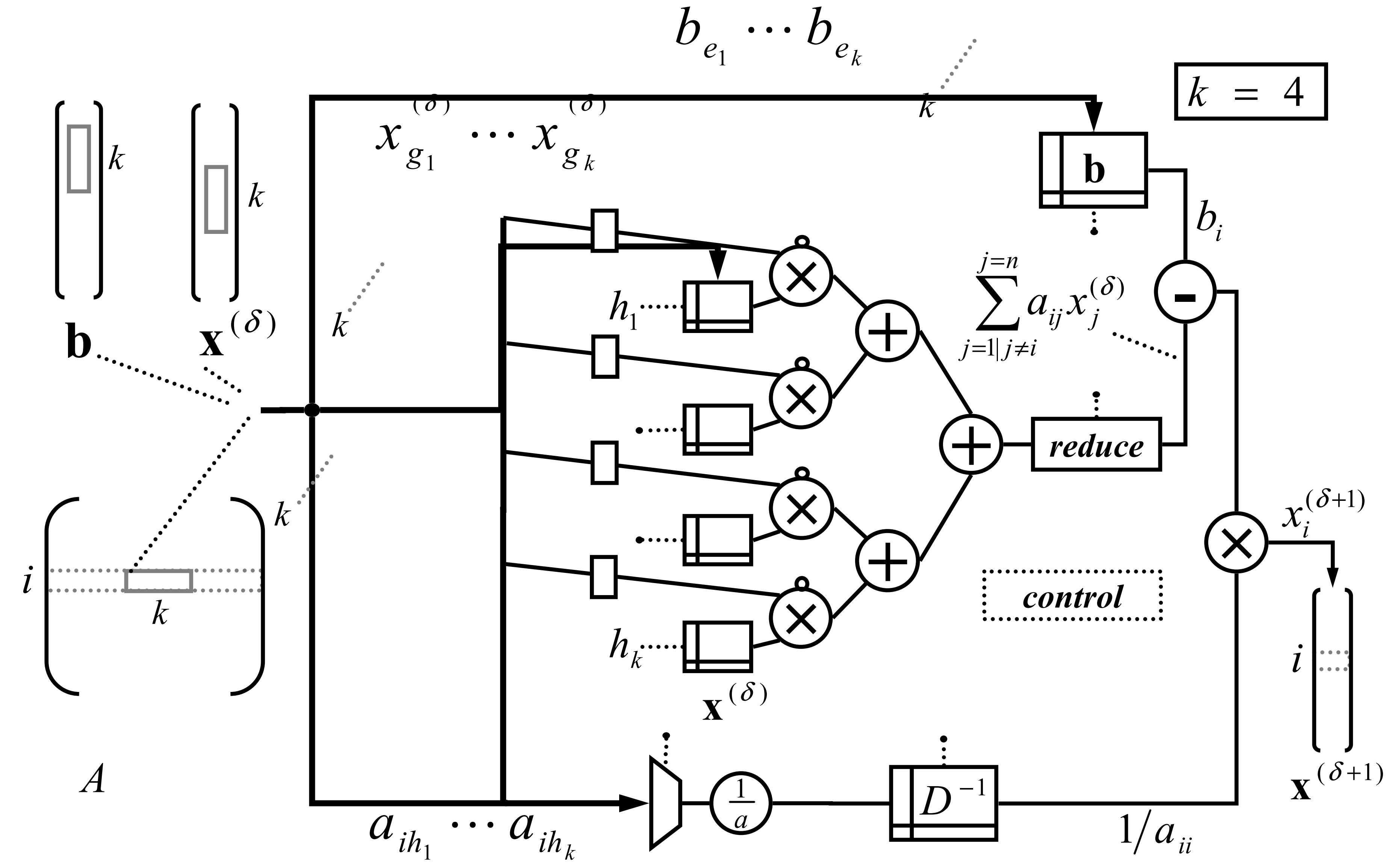
- Pipelined FPU's
  - many stages
  - higher frequency
- Scalability
- Reduction
  - adder "loop" fails
  - pipeline latency
- Serialized
  - division last
  - very costly



Solution

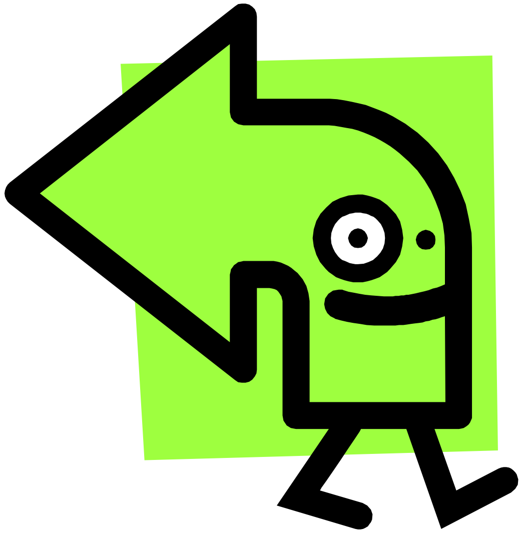
- Pipelined k-vector data path
  - deeply pipelined FPU's
  - input one k-vector per clock
- Multiplexed inputs
  - store x, b in on-chip BRAM
- Reduction circuit
  - accumulates n/k partial sums
  - no stalls, no buffer overflow
- Hidden divider latency
  - store D^{-1} in on-chip BRAM
- Flexible (dense or sparse)

Dense Matrix



Sparse Matrix?

- Same pipelined data path
- Compressed sparse row (CSR) format
- Minor modifications
  - input column (col)
  - input row pointer (ptr)
  - controller mods



Compressed Sparse Row Format

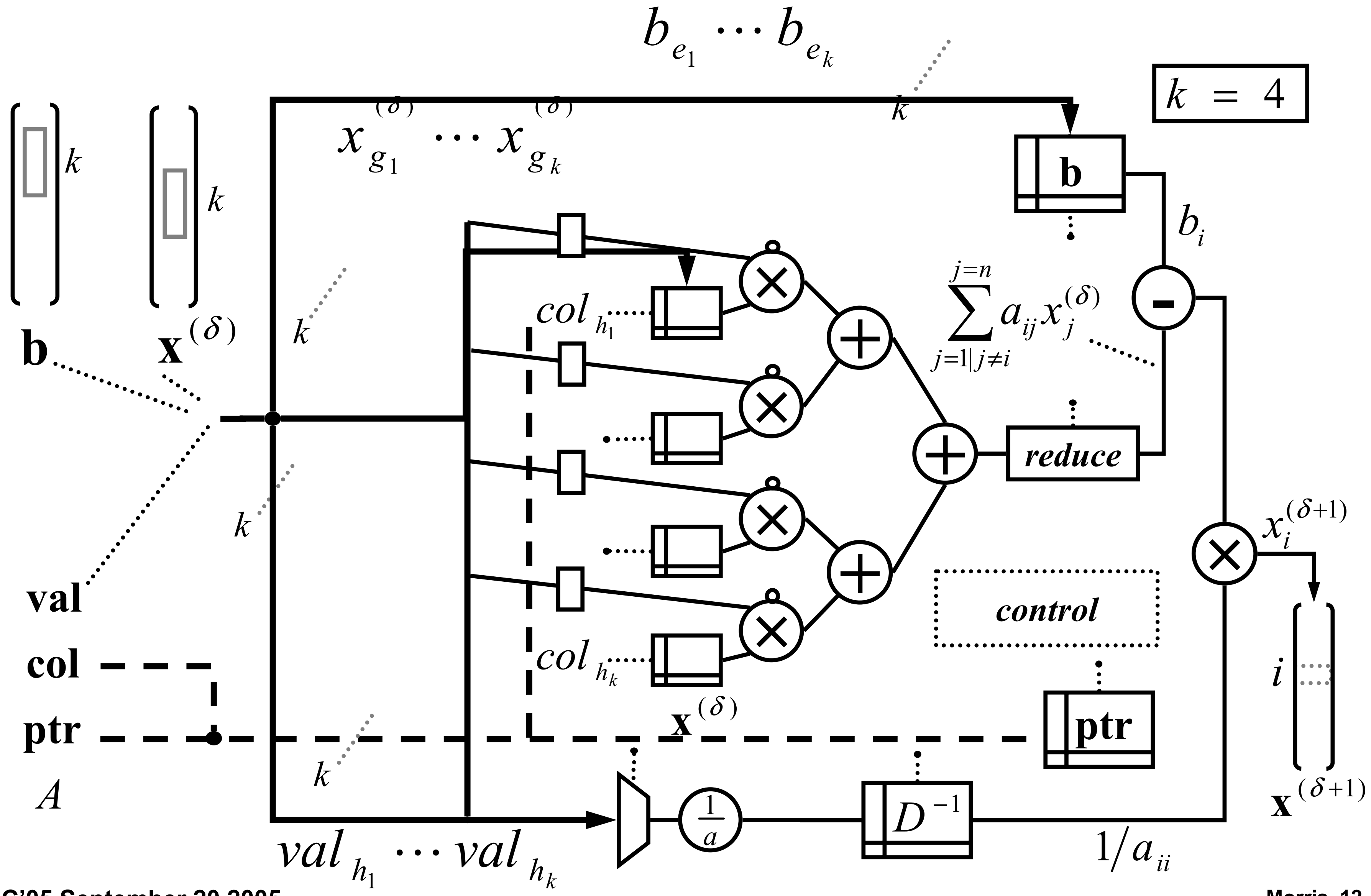
- Mostly 0's (*nnz* non-zeros)
- Use three vectors
  - val: non-zero values (row wise)
    - | val | = *nnz*
  - col: column index of each non-zero
    - | col | = *nnz*
  - ptr: index in val where row *i* starts
    - | ptr | = *n* + *1*
    - by convention, $ptr_{n+1} = nnz+1$
- Number of non-zeros/row
  - calculated as $len_i = ptr_{i+1} - ptr_i$

- Example for $n = 4$

$$A_{n\times n} = \begin{bmatrix} 5 & 0 & 0 & 2 \\ 0 & 8 & 0 & 0 \\ 1 & 0 & 6 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

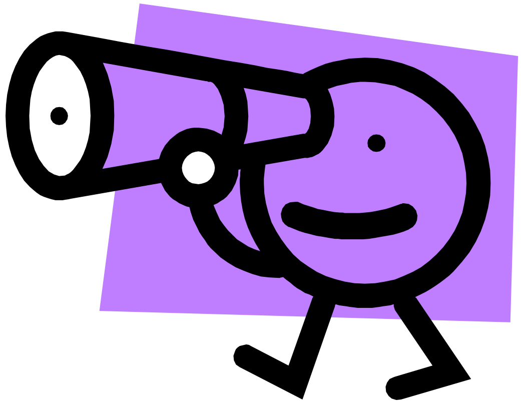
i	1	2	3	4	5	6
val	5	2	8	1	6	3
col	1	4	2	1	3	4
ptr	1	3	4	6	7	
len	2	1	2	1		

Sparse Matrix!

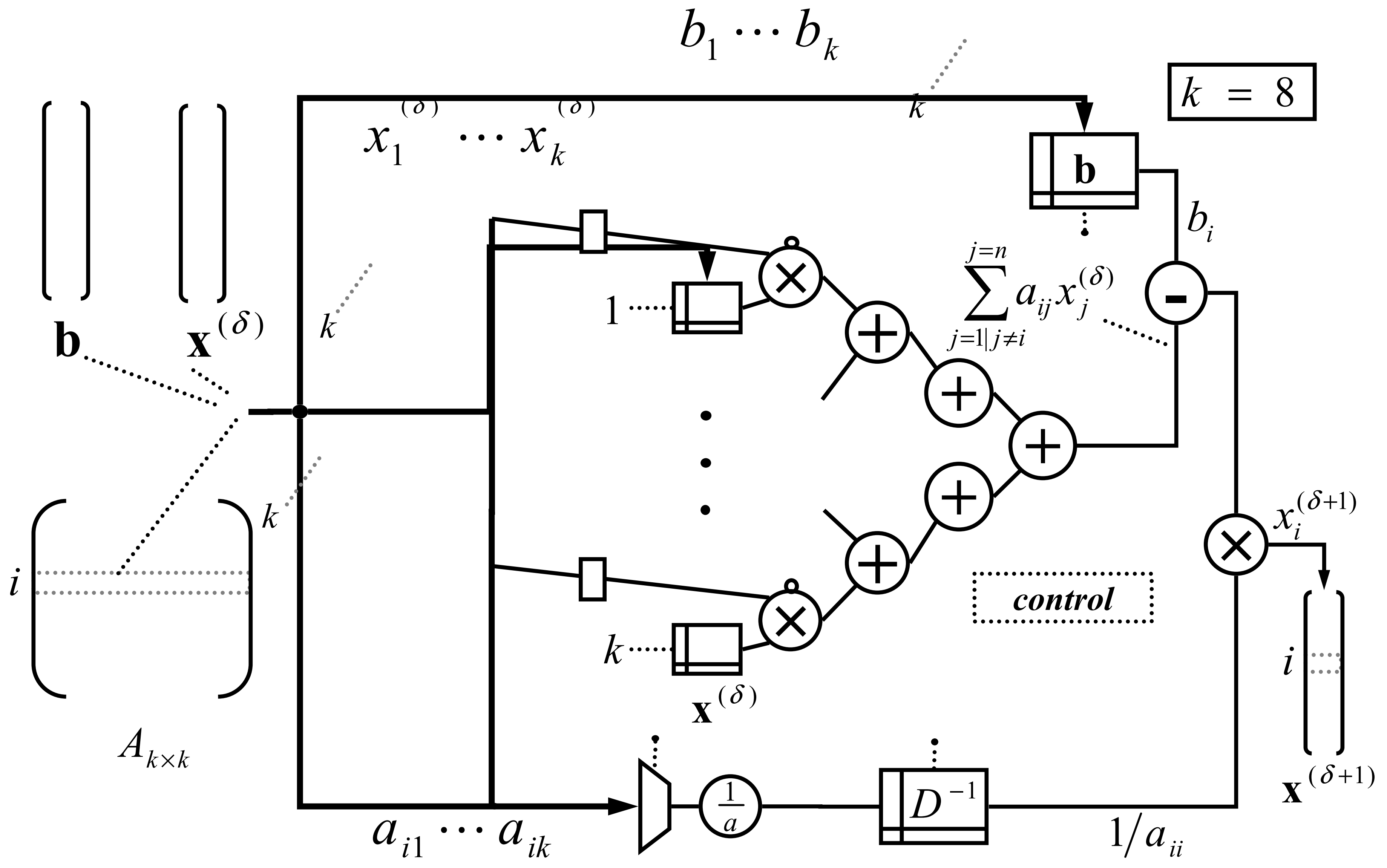


Experimental Data Path Implementation

- Tools
  - VHDL
  - Xilinx ISE 6.3.03i
  - ModelSim XE II 5.8c
  - Synplify Pro 8.1
- 64-Bit IEEE-754 floating-point IP cores
  - adder
  - multiplier
  - divider
- Targets
  - Virtex-II Pro XC2VP70-6
  - Virtex-II Pro XC2VP100-6



Data Path



$b_1 \cdots b_k$
$k = 8$
$x_1^{(\delta)} \cdots x_k^{(\delta)}$
k
b
b_i
$\sum_{j=1 \mid j \neq i}^{j=n} a_{ij} x_j^{(\delta)}$
-
$\mathbf{b}$
$\mathbf{x}^{(\delta)}$
k
1
k
i
$A_{k \times k}$
$x_i^{(\delta+1)}$
control
k
$\mathbf{x}^{(\delta)}$
i
$\mathbf{x}^{(\delta+1)}$
$\frac{1}{a}$
D^{-1}
$a_{i1} \cdots a_{ik}$
$1/a_{ii}$

Experimental Results

IP Core Post-PAR Statistics

IP core	latency [cycles]	f_{max} [MHz]	area [slices]
adder	14	170	1078
multiplier	10	170	935
divider	58	140	3015

Jacobi Data Path Post-PAR Statistics

FPGA device	k	I/O pins	latency [cycles]	f_{max} [MHz]	area [slices]
XC2VP70	8	581	52	103	20188
XC2VP100	8	581	52	107	20527

Related Work

Floating-Point IP Cores

G. Govindu, R. Scrofano, and V.K. Prasanna, "A Library of Parameterizable Floating-Point Cores for FPGAs and their Application to Scientific Computing," *International Conference on Engineering Reconfigurable Systems and Algorithms (ERSA'05)*

Reduction Circuits

G.R. Morris, L. Zhuo, and V.K. Prasanna, "High-Performance FPGA-Based General Reduction Methods," *Symposium on Field-Programmable Custom Computing Machines (FCCM'05)*

Complete Jacobi Design (to appear)

G.R. Morris and V.K. Prasanna, "An FPGA-Based Floating-Point Jacobi Iterative Solver," *International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'05)*

Conjugate Gradient (manuscript in preparation)

G.R. Morris and V.K. Prasanna, "Accelerating Conjugate Gradient on an FPGA-Augmented Reconfigurable Computer"

Acknowledgments

- My Lord, Jesus Christ
- My Wife, Lisa
- My Professor, Dr. Viktor Prasanna
- My Employer, ERDC MSRC (HPCMP)
- My Colleagues
  - Dr. Ward
  - Dr. Oppe
  - Dr. Fernandez
  - Ron and Ling

Questions

preguntas

¿

domande

問題

вопросы

?

fragen

Jerry Morris

grm@usc.edu
http://indus.usc.edu/grm