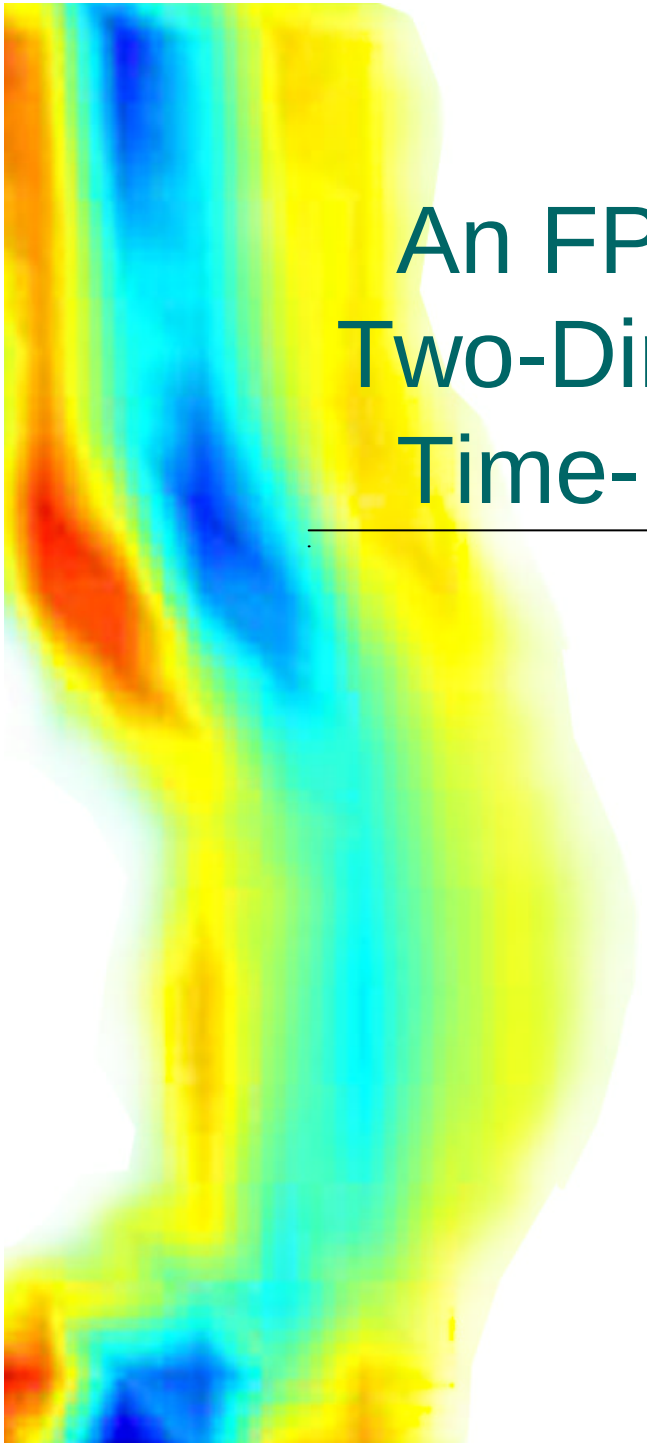


An FPGA Implementation of the Two-Dimensional Finite-Difference Time-Domain (FDTD) Algorithm

Wang Chen
Panos Kosmas
Miriam Leeser
Carey Rappaport

Northeastern University
Boston, MA

FDTD Algorithm and Implementation

- Finite Difference Time-Domain
  - Method for solving Maxwell's equations
  - Used for buried object detection
- Hardware Implementation
  - 3D to 2D model simplification
  - Data dependency analysis
  - Fixed-point quantization

Finite-Difference Time-Domain Method

- A direct time-domain solution of Maxwell's equations
- Accurate and flexible for solving electromagnetic problems
- Discretize time and electromagnetic space

Maxwell's Equations

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} - \sigma_m \vec{H} - \vec{M}$$

$$\nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \sigma_e \vec{E} + \vec{J}$$

$$\nabla \cdot \vec{D} = \rho_e$$

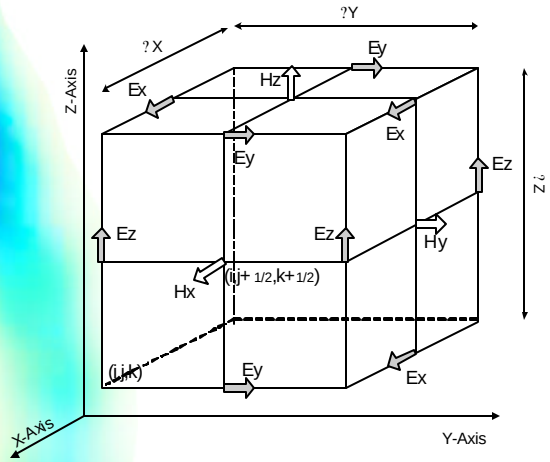
$$\nabla \cdot \vec{B} = \rho_m$$

$$\vec{B} = \mu \vec{H}$$

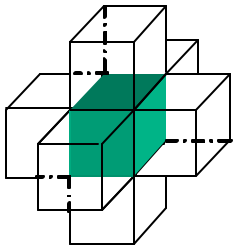
$$\vec{D} = \epsilon \vec{E}$$

FDTD Method (cont'd)

Yee Cell



Adjacent Cells



Taylor Series Expansion

$$\frac{\partial f(x_o)}{\partial x} = \frac{f(x_o+\Delta x) - f(x_o-\Delta x)}{2\Delta x} + O[(\Delta x)^2]$$

One FDTD Equation

$$\frac{\mu(j,j+1/2,k+1/2)}{\Delta t}[H_x^{n+1/2}(i,j+1/2,k+1/2)$$

$$-H_x^{n-1/2}(i,j+1/2,k+1/2)] = \frac{1}{\Delta z}[E_y^n(i,j+1/2,k+1)$$

$$-E_y^n(i,j+1/2,k)] - \frac{1}{\Delta y}[E_z^n(i,j+1,k+1/2)$$

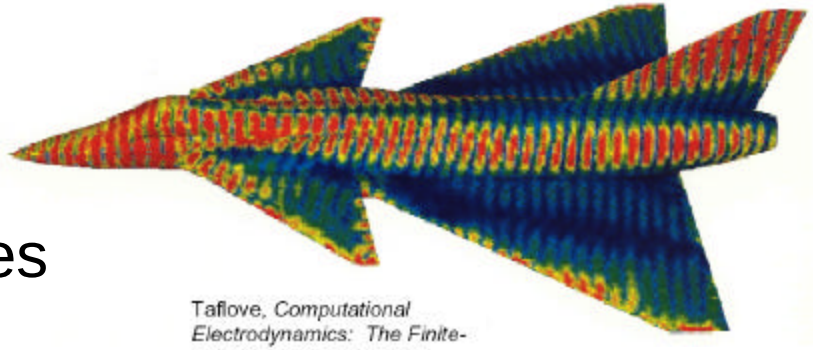
$$-E_z^n(i,j,k+1/2)] - \frac{\sigma_m(i,j+1/2,k+1/2)}{2}$$

$$\times[H_x^{n+1/2}(i,j+1/2,k+1/2)+H_x^{n-1/2}(i,j+1/2,k+1/2)]$$

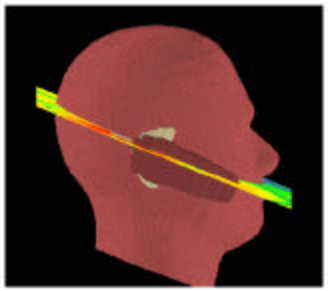
$$-M_x^n(i,j+1/2,k+1/2)$$

FDTD Applications

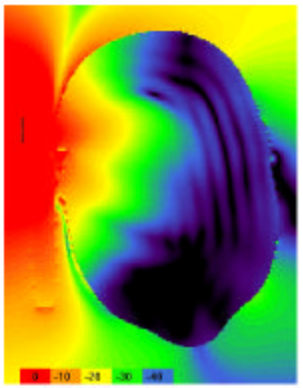
- Antenna Design
- Discrete Scattering Studies
- Medical Studies
  - The study of the cell phone electromagnetic waves' effect on human brain
  - The study of breast cancer detection using electromagnetic antenna



Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 1995.

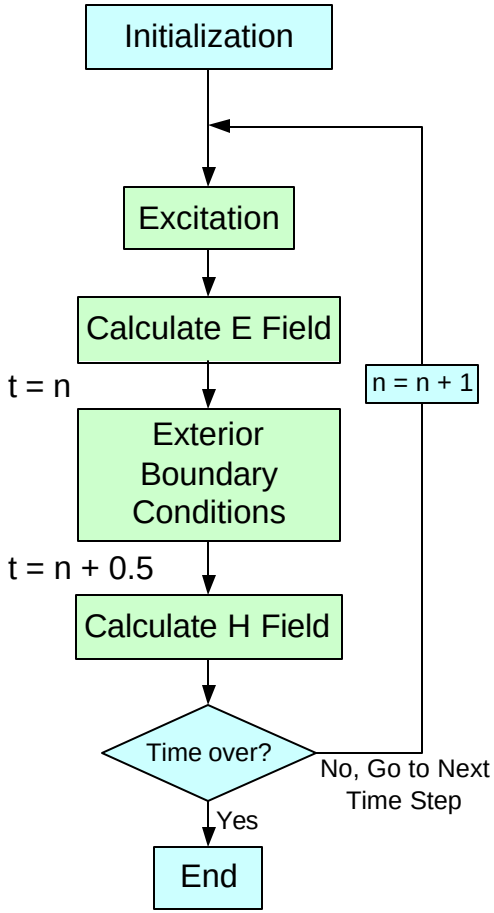


Cut plane through the cellphone

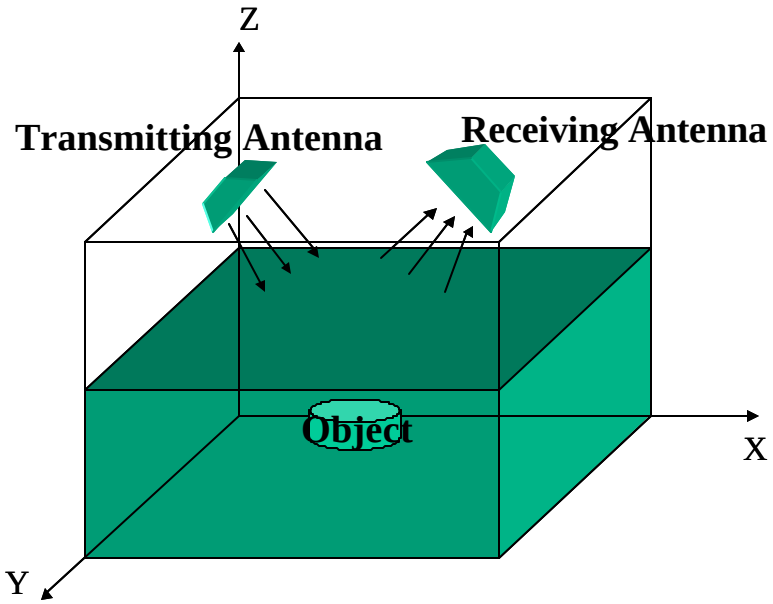


Remcom Inc. website: http://www.remcominc. com/html/index.html

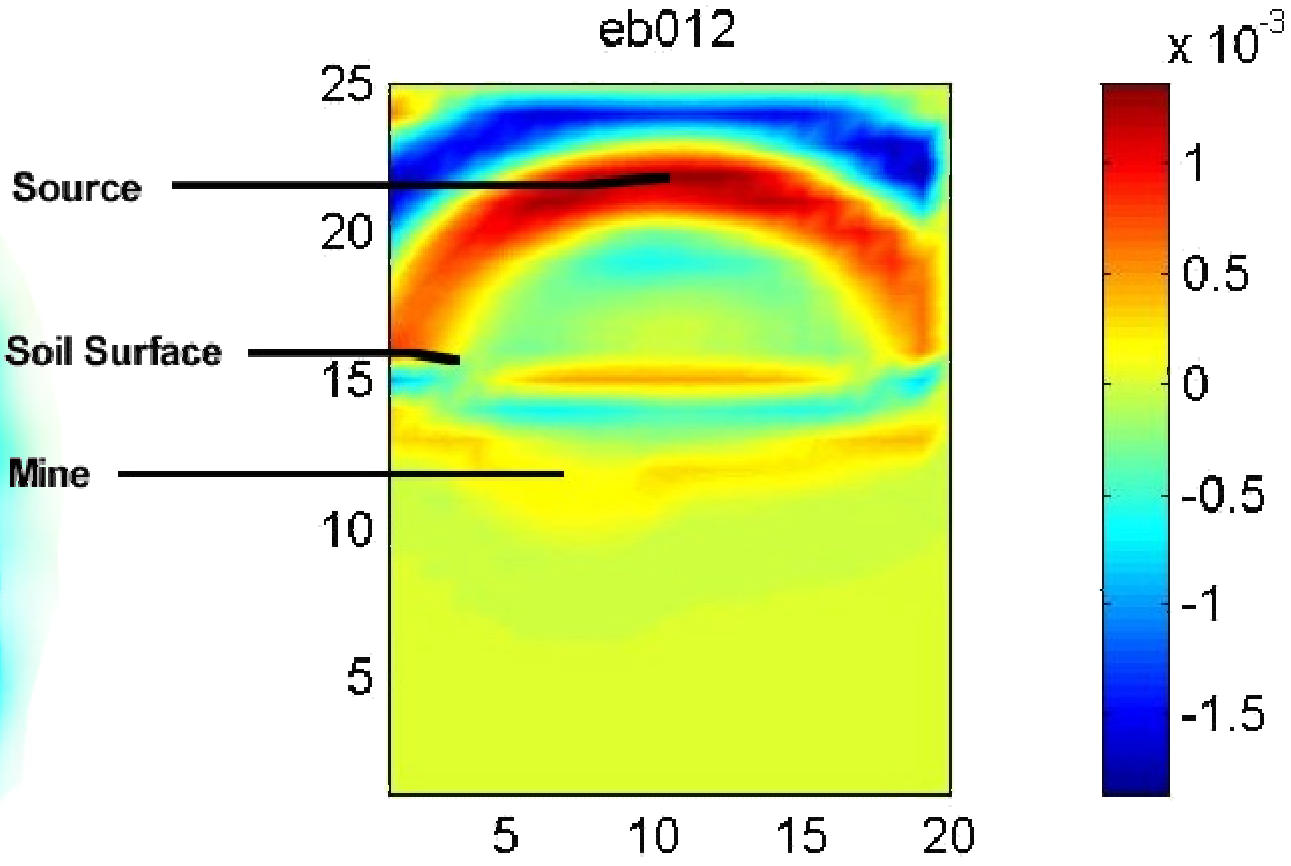
Buried Object Detection Forward Model



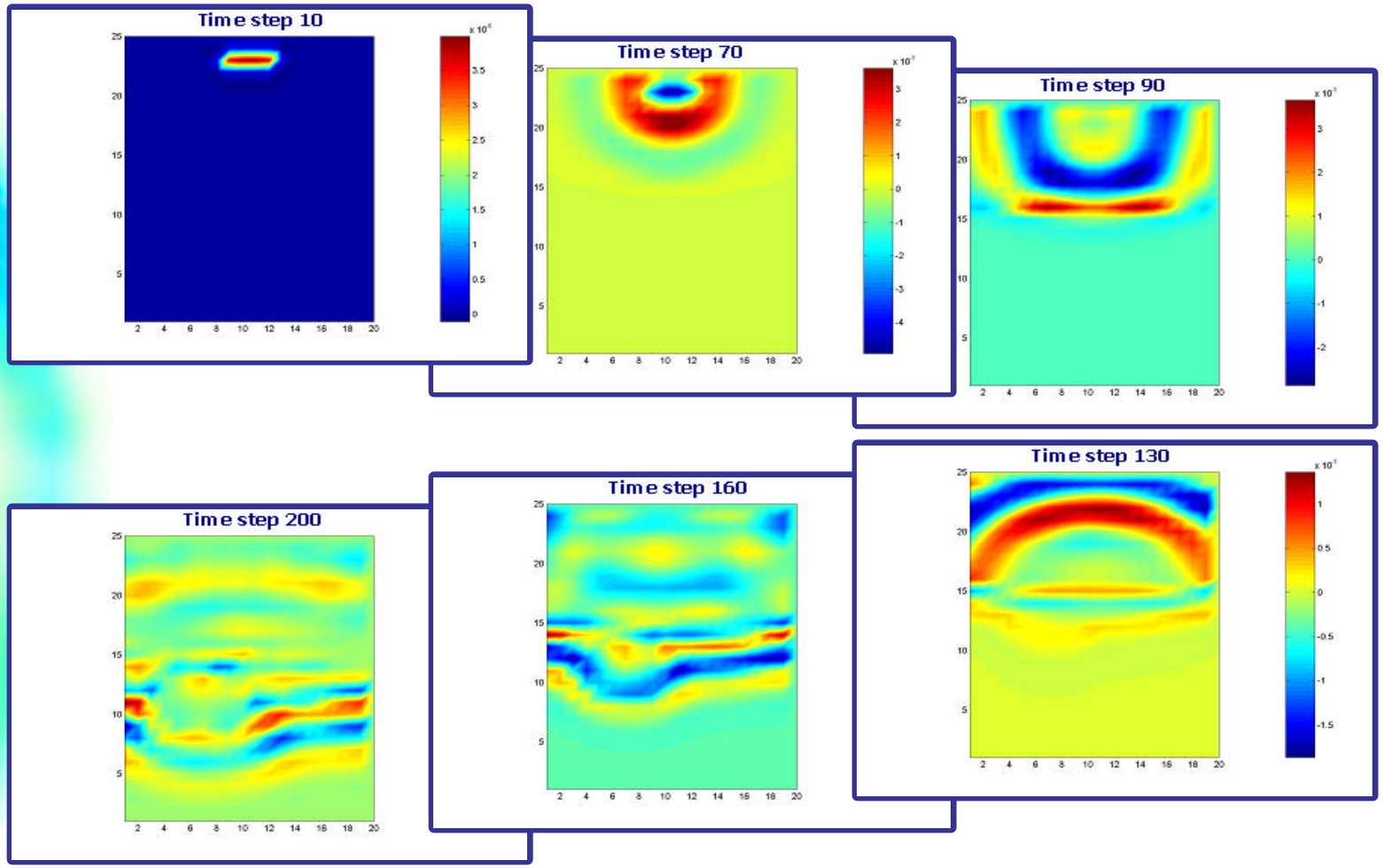
Buried Object Detection Model Space

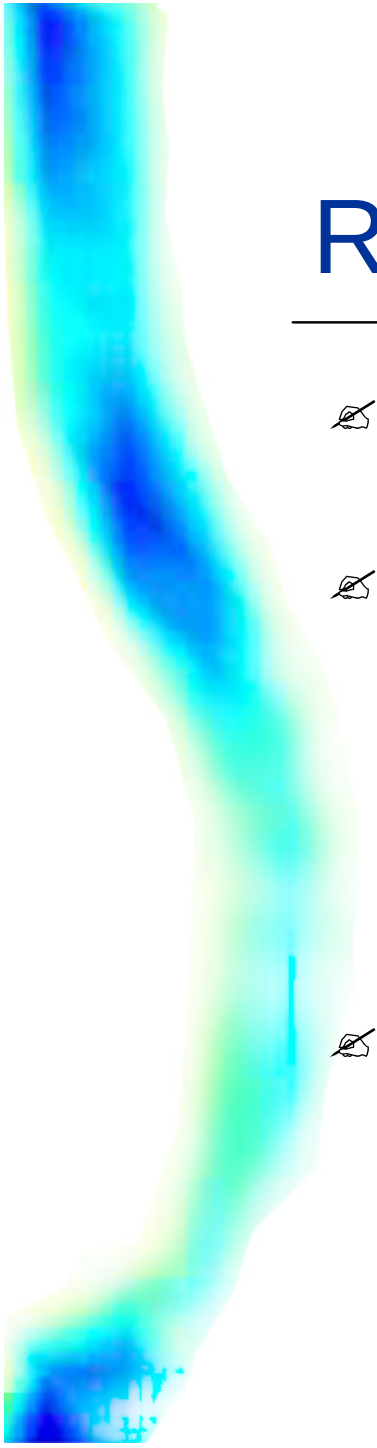


FDTD Simulated Model Space



FDTD Simulated Model Space (cont'd)

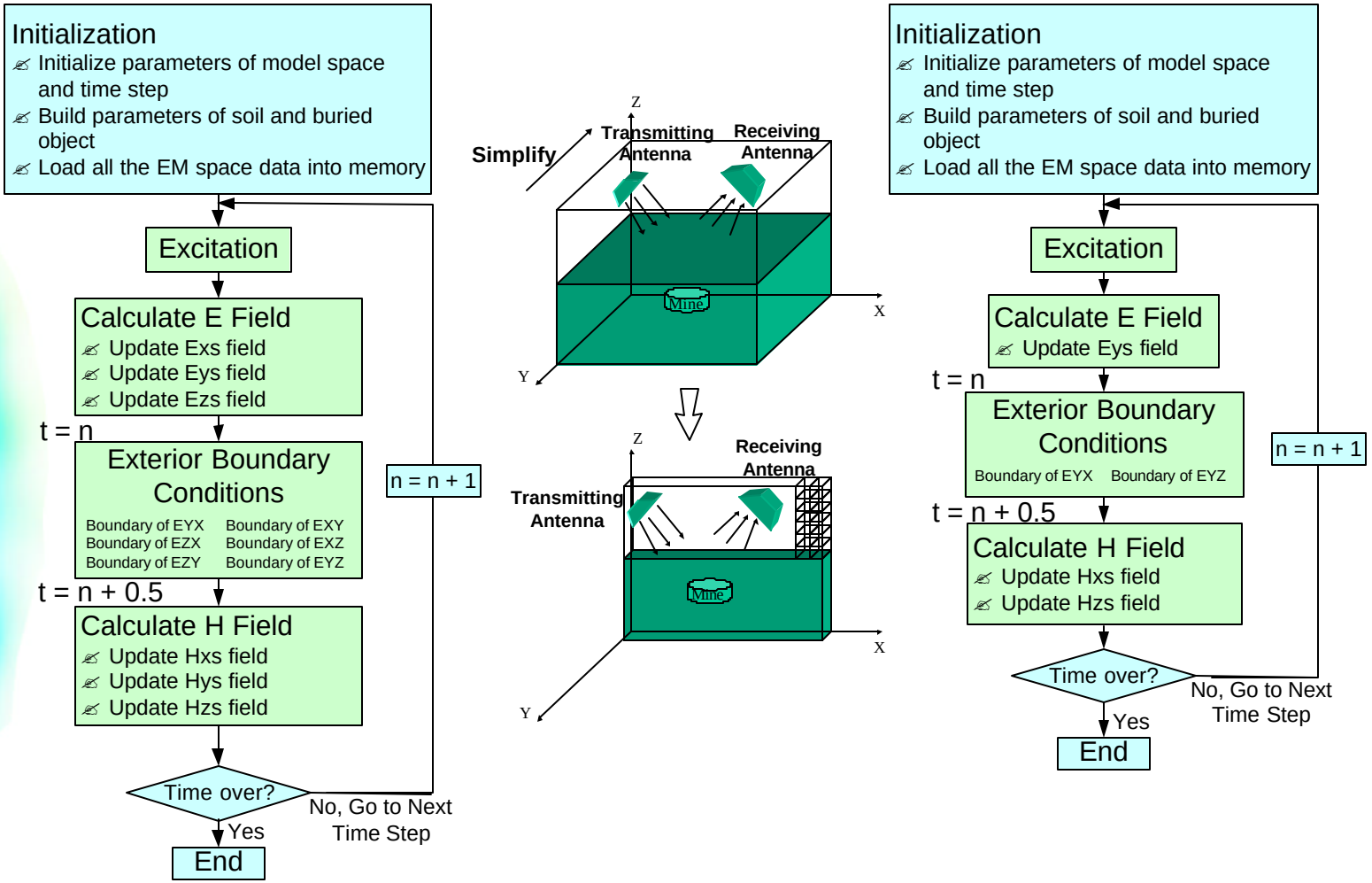




Related Work

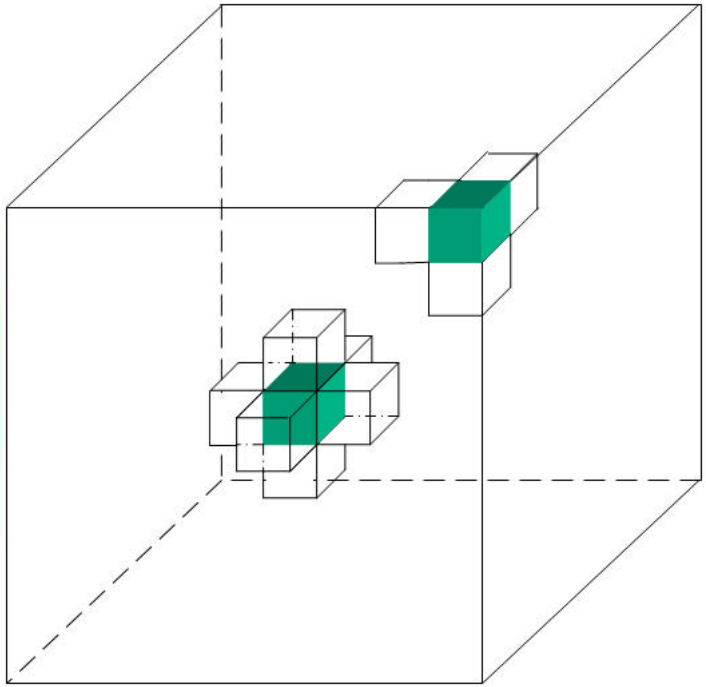
- Software acceleration of FDTD
  - Parallel computers do not provide significant speedup
- FPGA implementations of FDTD
  - 1D FDTD on hardware: architecture is too simple
  - Full 3D FDTD on hardware developed at UDel
    - Design is slower than software:
      - uses complex floating-point representation
      - no parallelism or pipelining
- Our 2D FDTD hardware implementation
  - 24 times speedup compare to 3.0G PC:
    - fixed-point representation
    - expandable structure

3D to 2D Model Simplification



Exterior Boundary Conditions

Mur-type Absorbing Boundary Condition



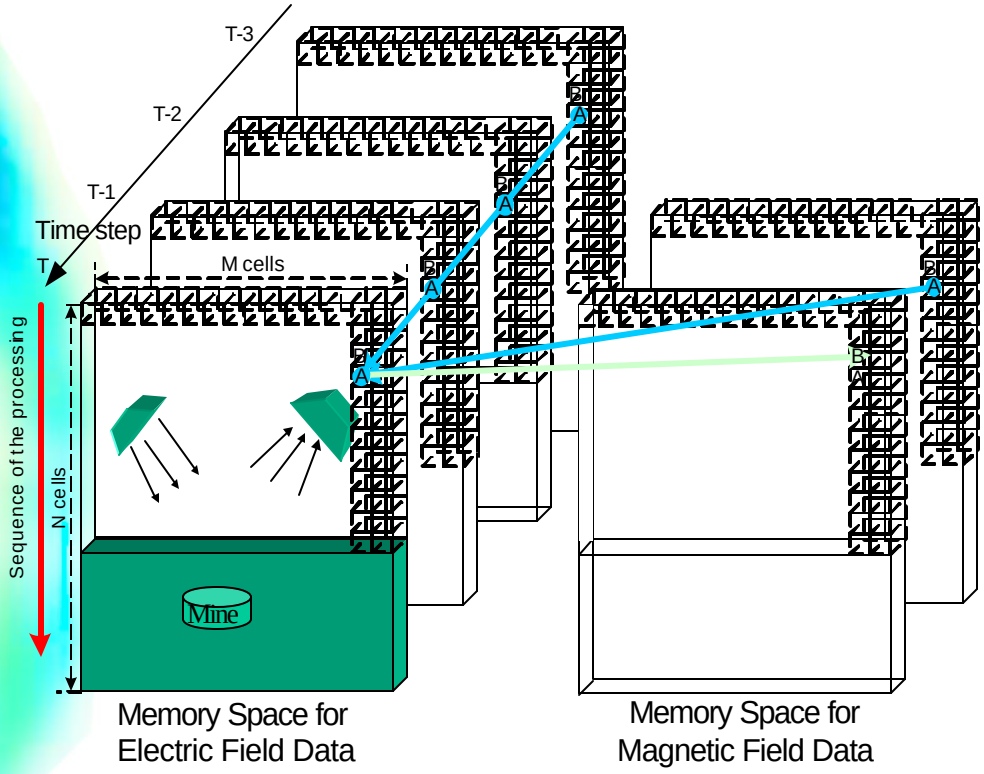
3D Model Space

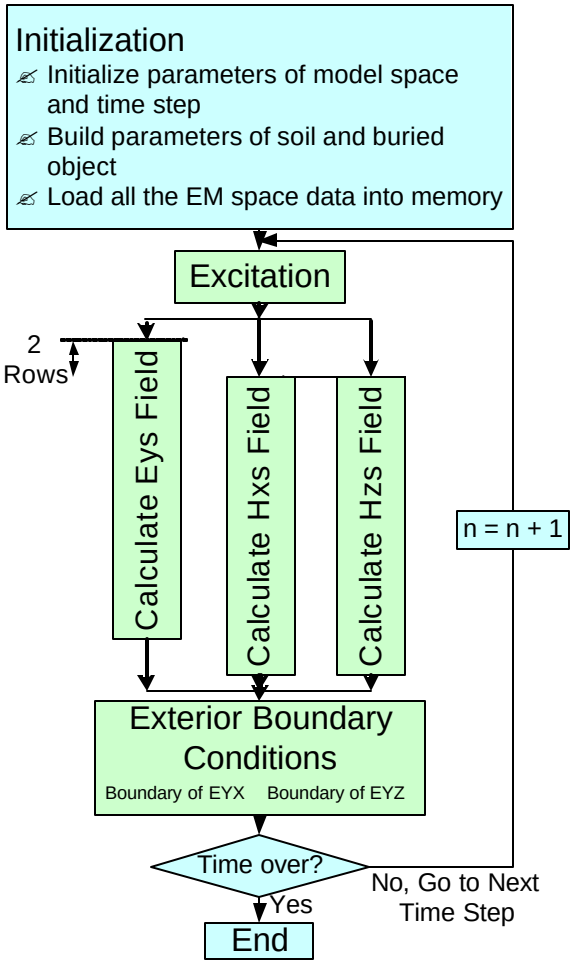
6 Faces and 12 Edges

2D Model Space

4 Edges

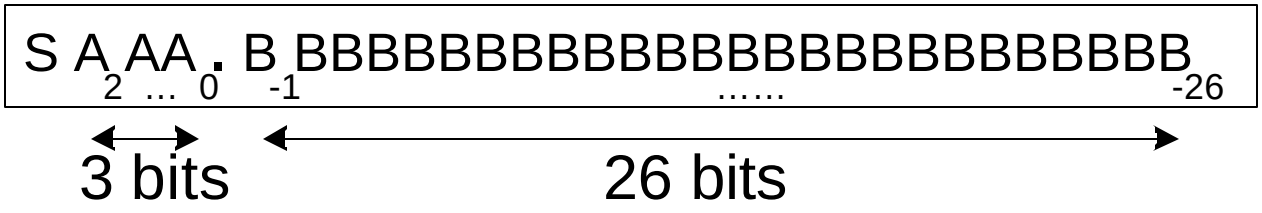
Data Dependency Analysis





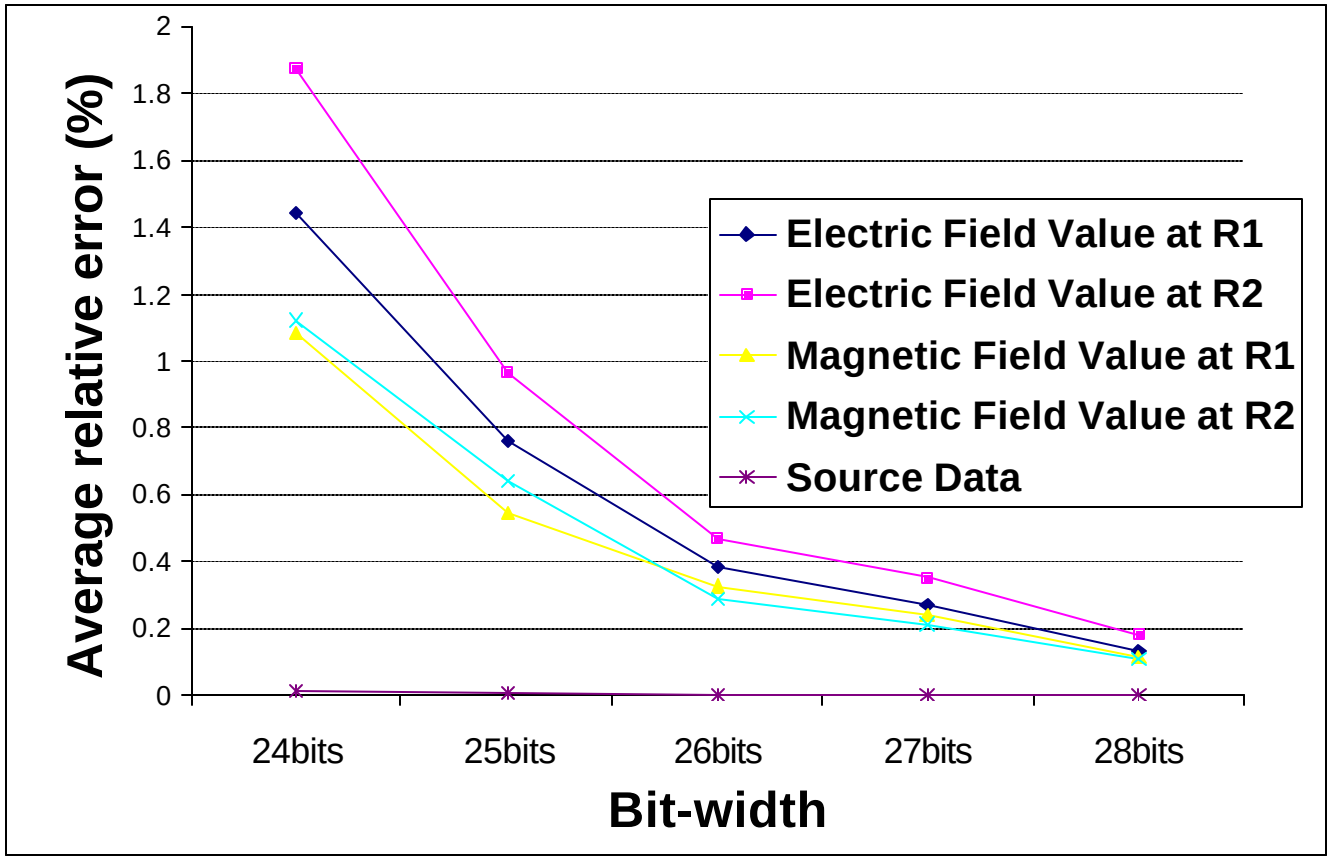
Hardware Acceleration

- Smart memory interface
- Parallelism
- Pipelining
- Quantized fixed point representation
  - Less area in datapath -- more parallelism
  - Careful error analysis to ensure accurate results

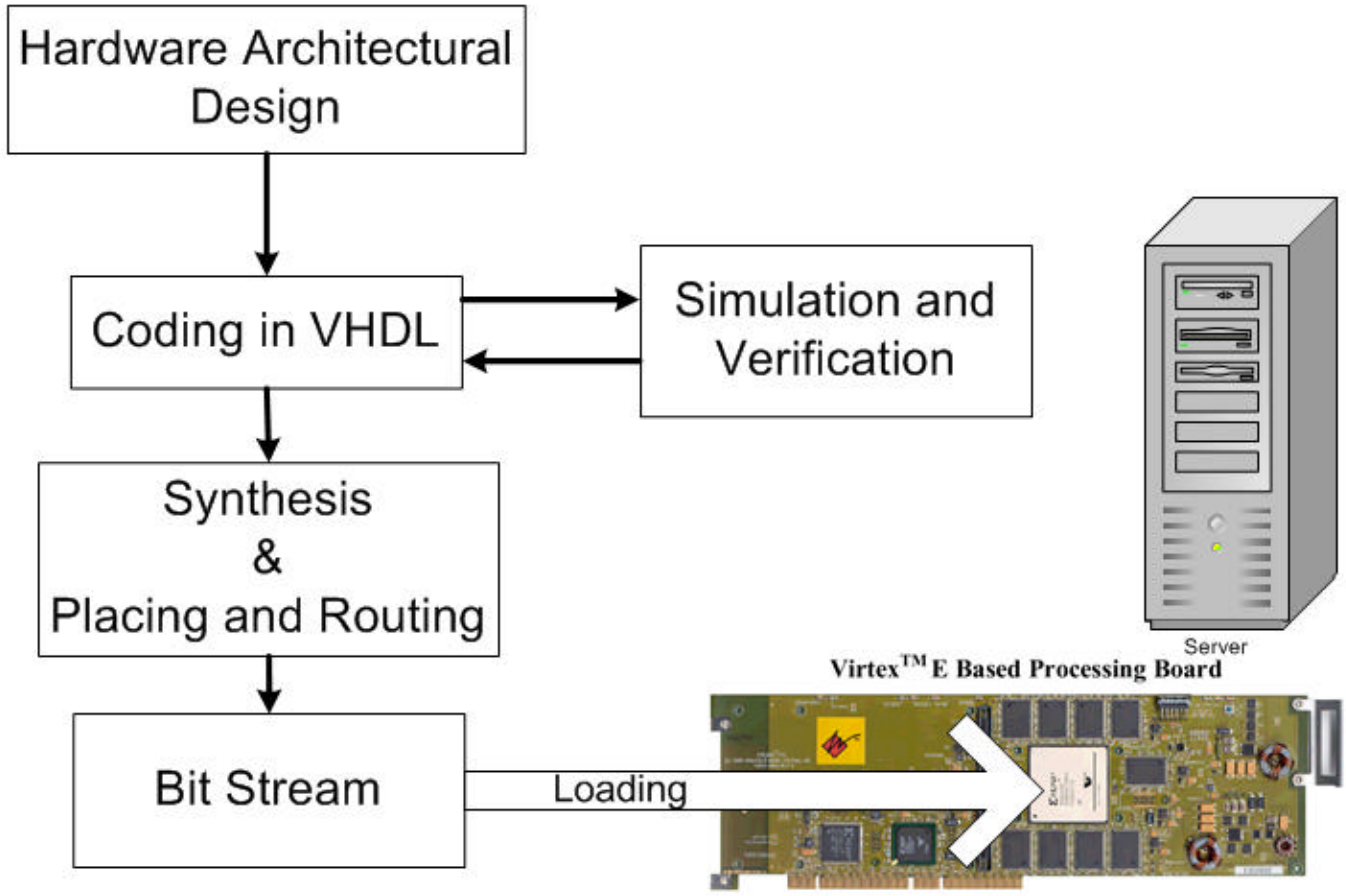


Fixed-point quantization

$$Relative\ error = \frac{|floating\ point\ data - fixed\ point\ data|}{|floating\ point\ data|}$$

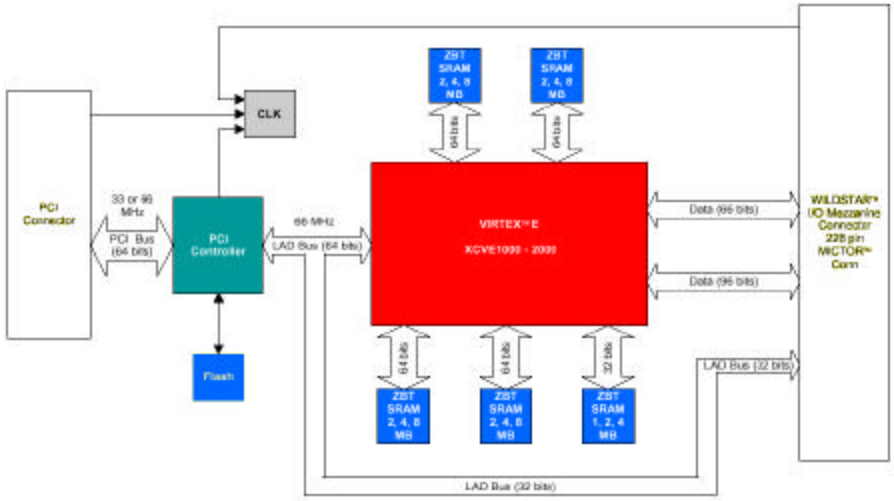


Design Flow


Hardware Architectural Design
Coding in VHDL
Simulation and Verification
Synthesis & Placing and Routing
Bit Stream
Loading
Server
Virtex™ E Based Processing Board

Firebird FPGA Board from Annapolis

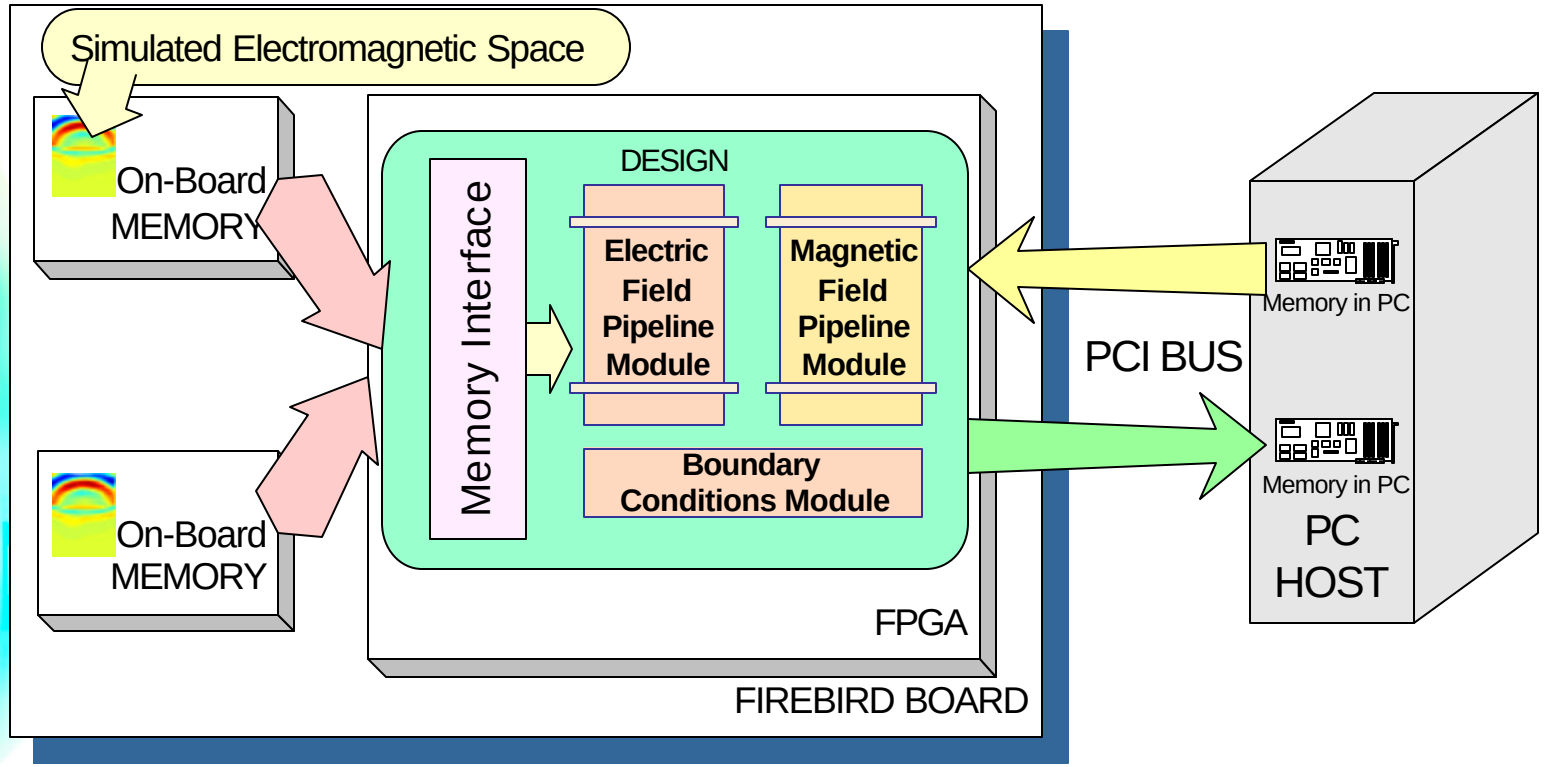
- A Xilinx VIRTEX-E XCV2000E with 2.5 million system gates
- Processing clock up to 150MHz FDTD runs at 70 MHz
- Five independent memory banks (4 x 64-bit, 1 x 32-bit) 288Mbytes in total
- 6.6Gbytes/sec of memory bandwidth
- 3Gbytes/sec of I/O bandwidth



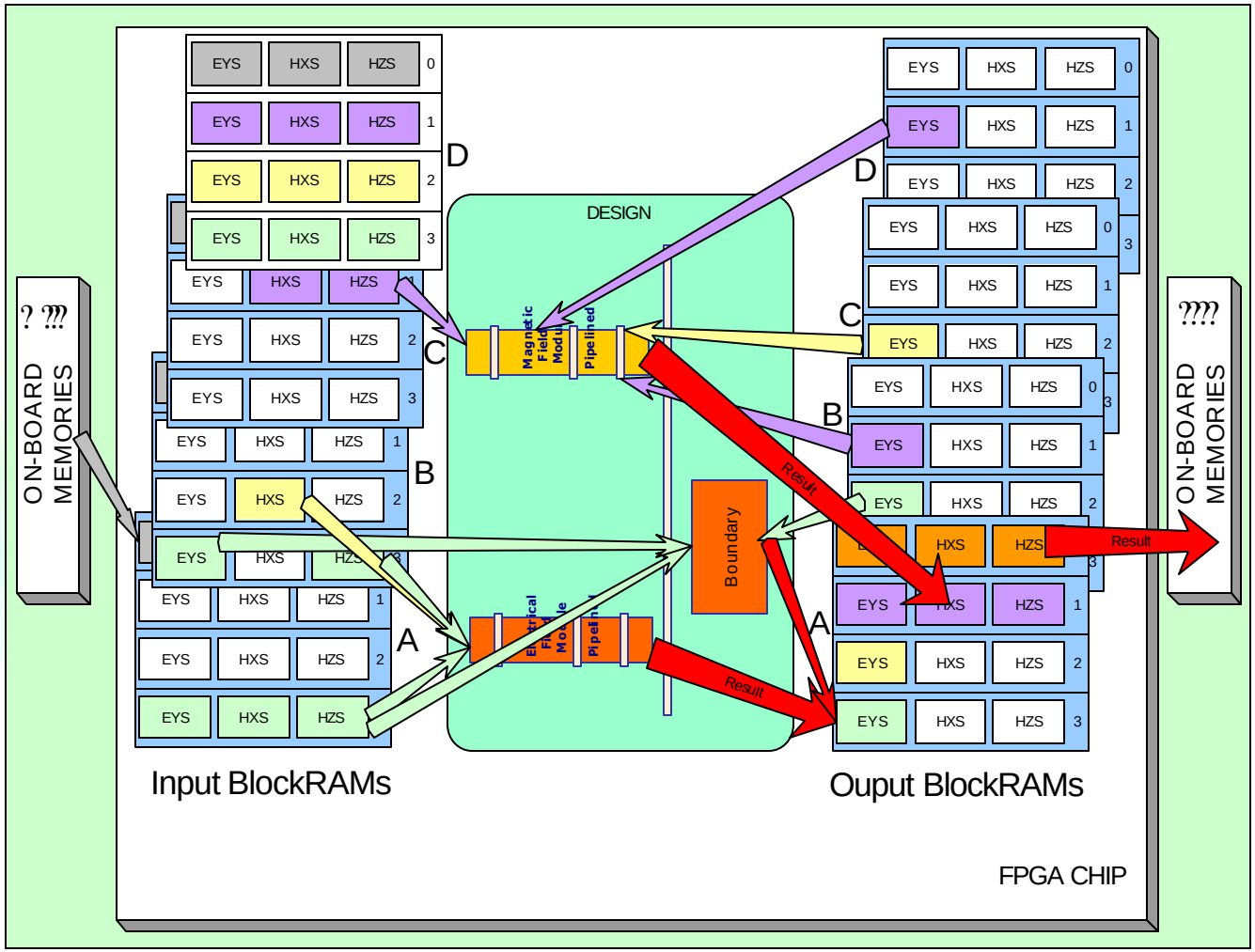
Utilization of Xilinx XCV2000E FPGA Chip

	Slices	BlockRAM
Number Available	19200	160
Number Used	8837	86
Percentage Used	46%	54%

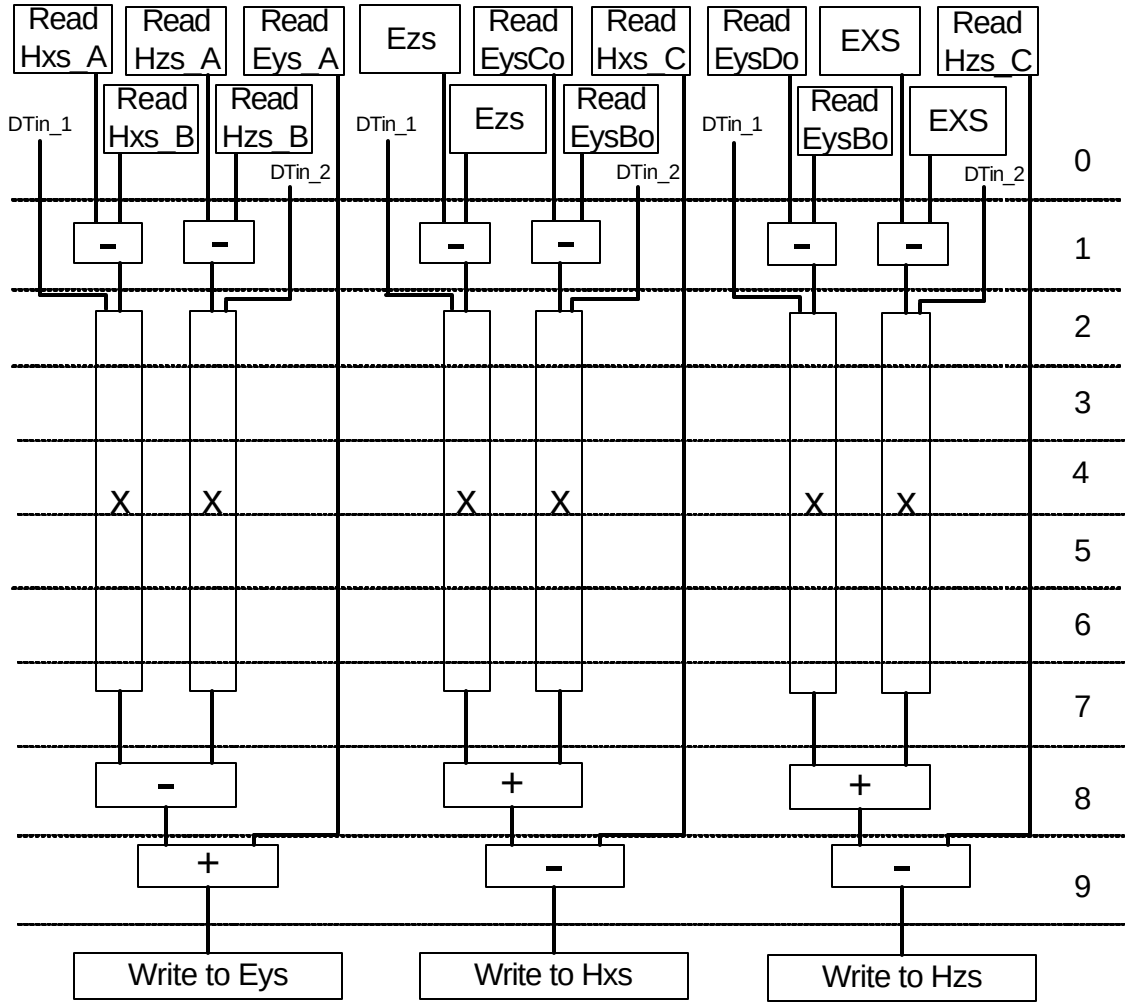
FDTD on Firebird Board



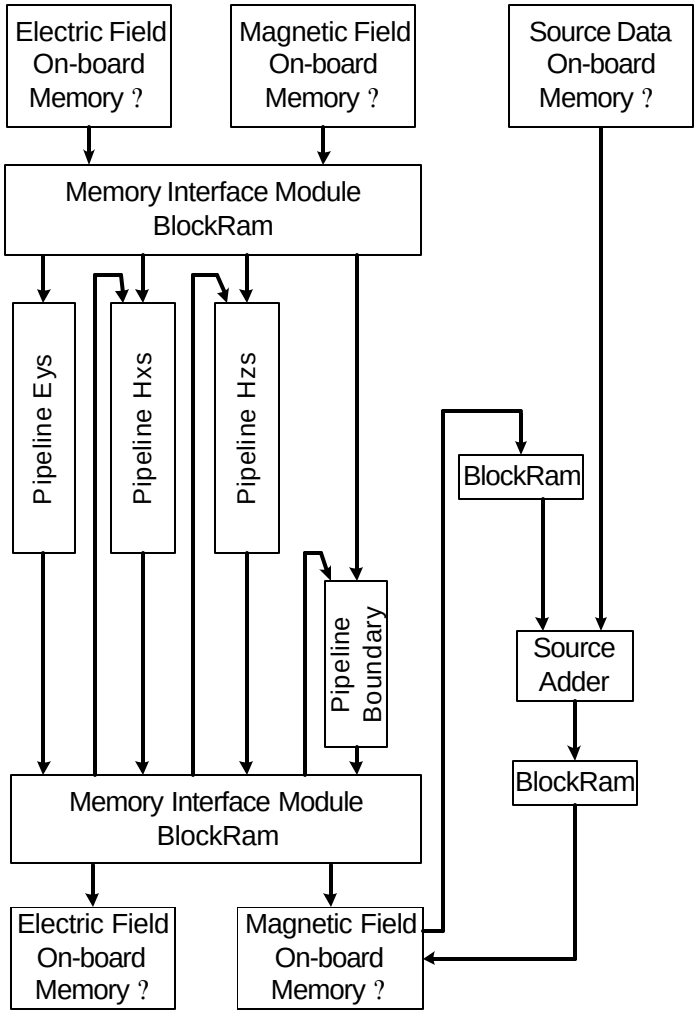
Memory Interface



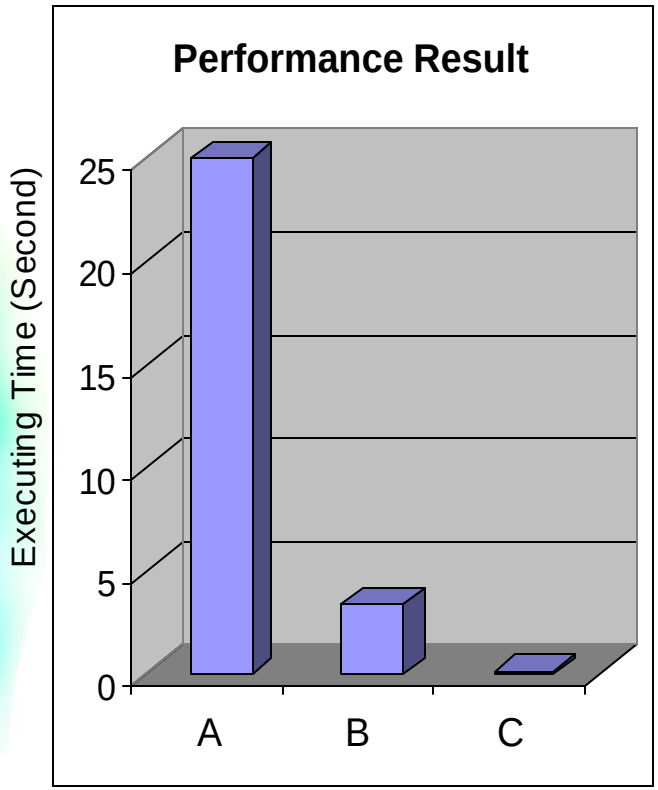
Pipelining and Parallelism



Data Flow


Electric Field On-board Memory ?
Magnetic Field On-board Memory ?
Source Data On-board Memory ?
Memory Interface Module BlockRam
Pipeline Eys
Pipeline Hxs
Pipeline Hzs
Pipeline Boundary
BlockRam
Source Adder
BlockRam
Memory Interface Module BlockRam
Electric Field On-board Memory ?
Magnetic Field On-board Memory ?

Results and Performance

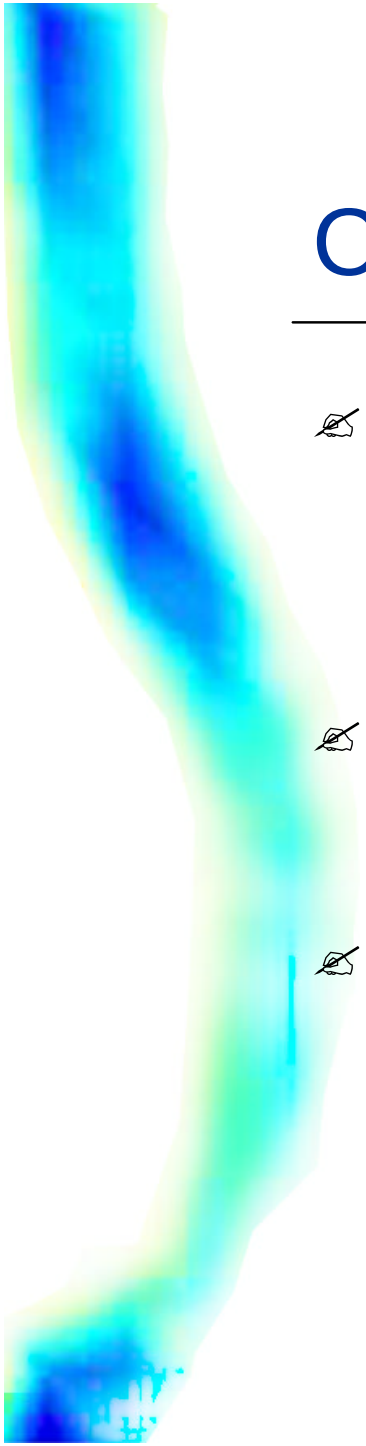


A Software Floating-point ~~ 25s
Fortran code at 440 MHz Sun Workstation

B Software Fixed-point ~~ 3.375s
C code at 3.0 GHz PC

C Hardware ~~ 0.145s
Design working at 70MHz

Model space 100*100 cells
Iterate 200 time steps



Conclusions

- FPGA Implementation of FDTD exhibits significant speedup compared to software: 24 times faster than 3GHz PC
- With larger FPGA, more parallelism will be available, hence more speedup
- Current design easily extendible to handle multiple types of materials, 3D space

Future Work

- Upgrade curent design to handle multiple types of materials
- Upgrade to 3D model space
  - Add three more field updating algorithms: same structure as the original three algorithms
  - Upgrade boundary condition updating algorithm
  - Redesign memory interface
- Apply FDTD Hardware to other applications