The Processor That Don't Cost a Thing

Peter Hsu, Ph.D.
Peter Hsu Consulting, Inc.

http://cs.wisc.edu/~peterhsu
DRAM+Processor

• Commercial demand
  – Heat stifling industry's growth
    • Heat density limits small systems
    • Total power dissipation limits large systems
  – Acceptance of parallel programming

• Feasible solution
  – Enough memory per chip to make useful system
  – Powerful microarchitecture using DRAM process
System Architecture Evolution
Circa 1995

- Super
- Server
- Games
- PC
- Embedded

Itanium

X86
System Architecture Evolution
Circa 2000

- Super
- Server
- Games
- PC
- Embedded

- Super
- games
- PC
- Embedded

Discrete  Continuous
Two Kinds of Computations

- **Discrete**
  - Integers, pointers
  - Unpredictable branches
  - Irregular data patterns
  - Caching effective
  - Low power efficiency
  - Cluster of few processors
  - Winner: x86

- **Continuous**
  - Floating point numbers
  - Loops
  - Data streams
  - Caching ineffective
  - High power efficiency
  - Cluster of many
  - Winner?
System Architecture Evolution

Circa 2005

- X86
- Server
- PC
- Embedded
- Super
- Games
- CELL
- DSP

Discrete
Continuous

Itanium
A Question of Granularity

- Trend: integrate 100 FPU @ multi GHz
  - Main memory off chip
    - Pin bandwidth woefully inadequate, need huge cache(s)
    - Terrible heat density, IR drop, hotspots...
      - Giant heat sink, noisy fan
- Why concentrate computing?
  - If there's parallelism, no need to be close together
  - If inherently serial, cannot use parallel resources
System Architecture Evolution
Circa 2010?

Server
Super
Games
DRAM+CPU?

X86
PC
Embedded

Discrete
Continuous
Display Technology

• Today's HDTV
  – 1-2M pixels
  – 500MB main memory
  – GB/s bandwidth
  – GFLOPS to render
  – Home cinema
    • $10,000

• Wallpaper display
  – 1B+ pixels
  – 50GB main memory
  – TB/s bandwidth
  – TFLOPS to render
  – Faux picture window
    • $10,000
    • Power major concern
Today's High-End Embedded System

- "Power" applications
  - H.264 MPEG4
  - Video games
  - Autonomous robots
- Like small PC
  - 512 MB, 5 GFLOPS
  - 20W, $100
Tomorrow's Low-Cost Computer

- 10x more “efficient”
  - 2W, $10
- Enables 100x volume
  - Sub $100 devices
  - Cheap clusters
  - Aggregate 1000's to form large systems
Off-Chip Bandwidth is Expensive

- 512M DDR2, 89mm²
- 1.4GB/s, 8% I/O, 0.8W

- 256M GDDR2, 52mm²
- 6.4GB/s, 21% I/O, 1.5W
Most Efficient Granularity

- Crunch data within DRAM
  - Drastically reduce off-chip communication
- Must maintain memory cost-effectiveness
  - Similar area (yield)
  - Comparable power (package)
  - Don't change process (duh)

- How much computing is possible inside DRAM?
Why DRAM+CPU Now?

- Many previous disappointments
  - PIM, IRAM...
- Inherently a high performance solution
  - Power no issue for low bandwidth chip-to-chip
    - Separate DRAM & CPU chips not inefficient
  - Large applications need large memory (100's MB)

- Sufficient memory capacity
Outline

- Motivation
- About DRAM
- What can we realistically integrate?
- A viable microarchitecture
- Fun things to do
DRAM Structure

- Base array
  - 256 x 512 cells
  - Subword lines
  - Tungsten bit lines

- Mat
  - 16 x 16 subarrays
  - M1 main word lines
  - M2 global DQ lines
Area Utilization (65% Cells)

- **X direction**
  - 5% subword drivers
  - 10% X decoder
  - 10% off-chip I/O

- **Y direction**
  - 10% sense amps
  - 5% Y decoder
  - 10% DQ drivers
Process

• Transistors
  – Priority is low leakage
  – Relatively thick oxide, high threshold, long channel
  – Roughly 3 generations slower than logic process

• Interconnect
  – Small bit line pitch requires high resistance tungsten
  – Only need 2 aluminum or copper layers
  – Wide, low resistance M2 for long distances
Interconnect Details

M2  P  M0  M1

Top

...  M2  M1

DRAM  Logic  P
DRAM Timing

2.5ns

15mm | X-dec | wordline | bitline | DQ | 15mm

Taa, Tcy = 12.5ns

Subarray

BL equalize | WL up | BL sense | BL write | WL down
Integration Ramifications

• Pros
  – Close to main memory
    • Low latency, good BW
  – Dense complex cells
    • Flip-flops, latches
    • Full adders, multipliers
  – Clock gating effective
    • Easy power mgnt

• Cons
  – Few global wires
    • Extremely hierarchical microarchitecture
  – Low P&R density
    • Simple control scheme
Bottom Up Design Philosophy

- **Boundary conditions**
  - Logic area (yield, cost, sales volume)
  - Bandwidth (power, routing overhead)
  - Process limitations (logic construct density)

- **Microarchitecture**
  - How many FPUs (area & bandwidth, cache?)
  - Frequency (power, number of banks)
  - Appropriate control structure (area)
4 Gbit in 45nm (2009)

16.2 x 6.8 = 110mm^2
Area

- 45nm process like 130nm logic
  - 150K gates/mm^2 (33% utilization)
    - M1 5600 tracks/mm, M2 2800
- How many FPUs?
  - IEEE 32b ~0.2mm^2
  - 12 FPUs ~30% of 10mm^2 logic area
    - 30% PLL, charge pump, I/O, etc.
- Leaves 40% = 600K gates
Power

• Magic number: 2W
  – Convection cool, no heat sink, low junction temp.
  – Cheap package, minimum system cost

• Logic area 0.9W
  – 180nm FPU 1.8V 266MHz = 150mW [DIVA]
  – \(\frac{1}{4}\) area, \(\frac{1}{2}\)V, 1.5x frequency (400MHz) = 30mW
  – FPUs 40% area, same power density

• Chip 1.8W (0.8 mem + 0.9 logic + 0.1 I/O)
Bandwidth

- Memory power: bandwidth, distance, voltage
  - 130nm 1.5V 6.4GB/s = 0.8W internal + 0.7W I/O
  - DRAM chip size constant @ 80-100mm^2
  - 45nm voltage 0.9 (perhaps)

- How many ports?
  - \((\frac{1.5}{0.9V})^2 \times 6.4 = 18\text{GB/s} @ 0.8W\)
  - 18G / 4B / 12 words = 375MHz
  - One memory access / multiply-add
A Viable Microarchitecture

- Vectors
  - Wire efficient (no multiport RF), simple control
- Multicore
  - One memory port per core maximizes utilization
- Multistream
  - Tolerate memory latency in scalar code
- Cache-less
  - SRAM bad: leakage, power, yield, migration
4 Gbit in 45nm (2009)

16.2 x 6.8 = 110mm^2
DRAM Layout Drives Floorplan

1.4mm 3900 M2

0.8mm
4500 M1

460 M2 0.15mm

180um pitch

0.2mm

processor

bank

x decoder

y decoder & lane

dq drivers
Microarchitecture Summary

- **Memory**
  - 64b, 8 lanes, 8 banks each, 4x4 crossbar, pair arbitrate
  - 5 cycles latency
- **Frequency**
  - 400 MHz
- **Processor (6)**
  - 64b vector paired-single FPU
  - 4 scalar streams share mem port, hides latency
Physical Design of Processor

<table>
<thead>
<tr>
<th>Adder</th>
<th>Vector Registers</th>
<th>Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>1500 M2</td>
<td>840 M2</td>
<td>30 M1 / bit x</td>
</tr>
<tr>
<td>40K random control logic</td>
<td>I Buffers</td>
<td>64 = 0.35mm</td>
</tr>
<tr>
<td>700 M2</td>
<td></td>
<td>1250 M2</td>
</tr>
<tr>
<td>IU</td>
<td>IU</td>
<td>1700 M1</td>
</tr>
<tr>
<td>IU</td>
<td>IU</td>
<td></td>
</tr>
<tr>
<td>IU</td>
<td>IU</td>
<td></td>
</tr>
<tr>
<td>IU</td>
<td>IU</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I Buffers: 1250 M2
1700 M1

IU: 40K random control logic
460 M2

Adder: 1500 M2
840 M2

Vector Registers: 840 M2

Multiplier: 30 M1 / bit x
64 = 0.35mm

IU: 40K random control logic
460 M2
Package

- Fewest possible I/O
  - Diff. pairs in & out
  - Up to 10Gb/s each
- Chip scale BGA
  - 24 balls (14 pwr/gnd)
  - 1 cm^2, 3mm high
Cool Stuff

• Vector registers implemented as shift registers
  – Interleave bits of VR to bitslice crossbar
• Poor-man's simultaneous multistreaming
  – Concurrent scalar units, time-shared memory port
  – Low latency, simple control, extremely hierarchical
• Ultimate scatter/gather flexibility
  – Vector load/store by scalar units (4)
  – Fire-and-forget vector ops
System Architecture Evolution
Circa 2010?

Server

X86

PC

Discrete

Continuous

Super

Games

Embedded

DRAM+CPU?
Home Entertainment Possibilities

- 64 DRAM, 8 FPGA
  - 0.6 TFLOPS SP
  - 32 GBytes
- 8 FPGA routers
  - 32 fast signals
  - USB, video, etc.
- Sub $1000 cost
  - 200 Watts