DRAM+CPU: Build It, They Will Come

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

http://cs.wisc.edu/~peterhsu

Computer Industry Seems Stuck

- Saturated market for high performance
  - Who needs 5 GHz or 16 core PC?
  - Why design servers? PC clusters *are* servers
  - Stay-at-home video game consoles going extinct
- System-on-a-chip was false hope
  - Specialization & high mask charge don't mix
  - Thin profit cannot support R&D

Toys to Teraflops

- Semi industry = memory
  - Generic DRAM, flash for everyone and everything
- Computer industry must evolve
  - Generic computer chip good enough for everything
  - Add value: software, system, interconnect
- Challenge: base unit extremely cheap
  - Like $1, to sell billions
  - Unencumbered by patents, IP rights

Sell a LOT of Computers

- Consumer product
  - Consumed: eaten, wears out, broken, thrown away
  - Re-purchase: craving, peer pressure, useful?
- Silicon way too durable
  - Embed in less stable material (fabric)
  - Micro mechanical parts (wears out)
  - Organic electronics (deteriorate in sunlight)
  - “System design” problem...

Smart Fabric

- Shadows cue 3-D
  - Painting basics
  - Change with movement
- Colors, shadows, transparency
  - Real time update
  - Who's looking at you?
  - Which direction?

Biological Complexity Model

- Basic unit (cell) cheap, generic
  - As powerful, efficient as possible, within constraint
  - Viable standalone (bacteria)
- Aggregation (multicellular organism)
  - Replicate and subset (stem cell, specialization)
  - Locally wasteful, globally optimal

Meta Architectures

Discrete computing

- Pointers
- Unpredictable control
- Caching
- Little parallelism
- Not real time

Server, desktop, controller

Continuous computing

- Floating point
- Loops, arrays
- Streaming data
- Much parallelism
- Real time

Supercomputer, games, DSP

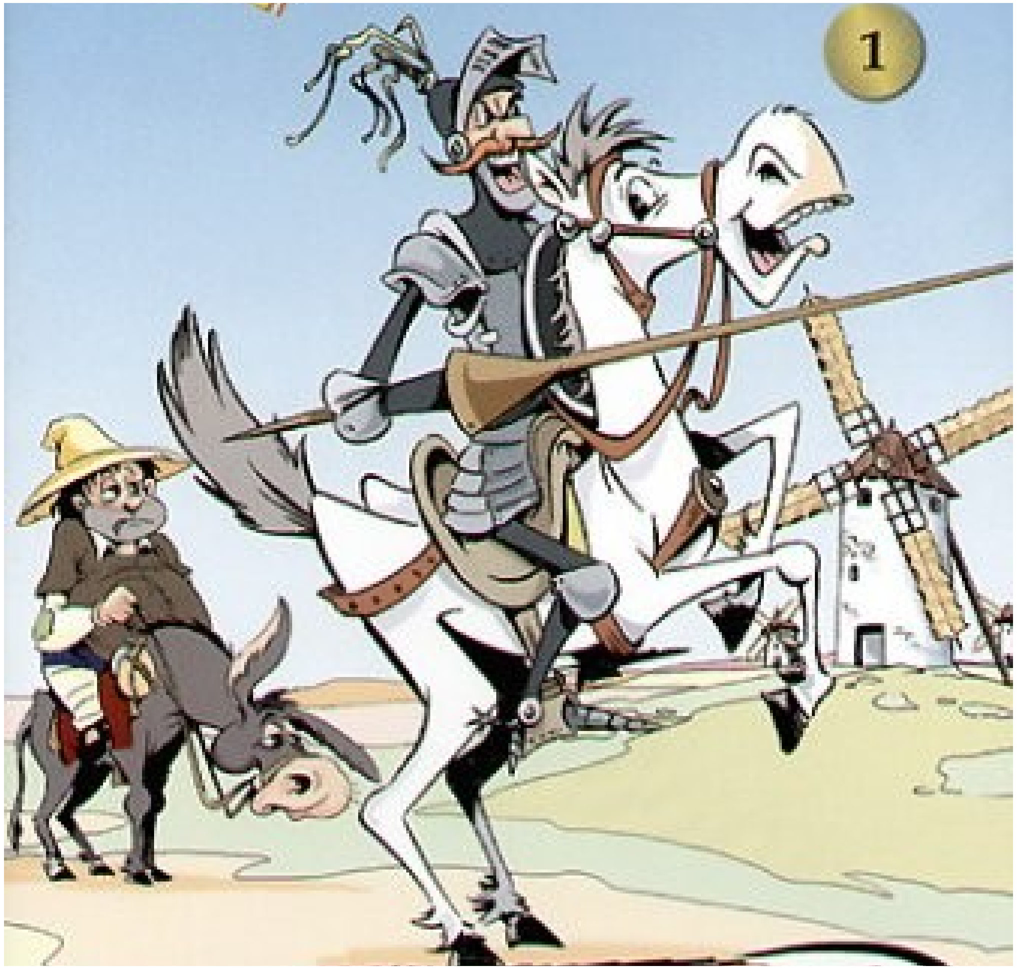
Focus on Continuous Computing

- “Serious”
  - Weather prediction, protean folding, nuclear reaction
- Games
  - Car crumpling, blood spurting, burning house
- Lifestyle
  - Smart fabric, prosthetics, pornography

Sell a LOT of computers

Cava Project

- “Free” computer
  - Open source ISA
  - Open source RTL
  - Open source Compiler
  - Open source OS
- *“One chip to run them all, and in the software bind them”*



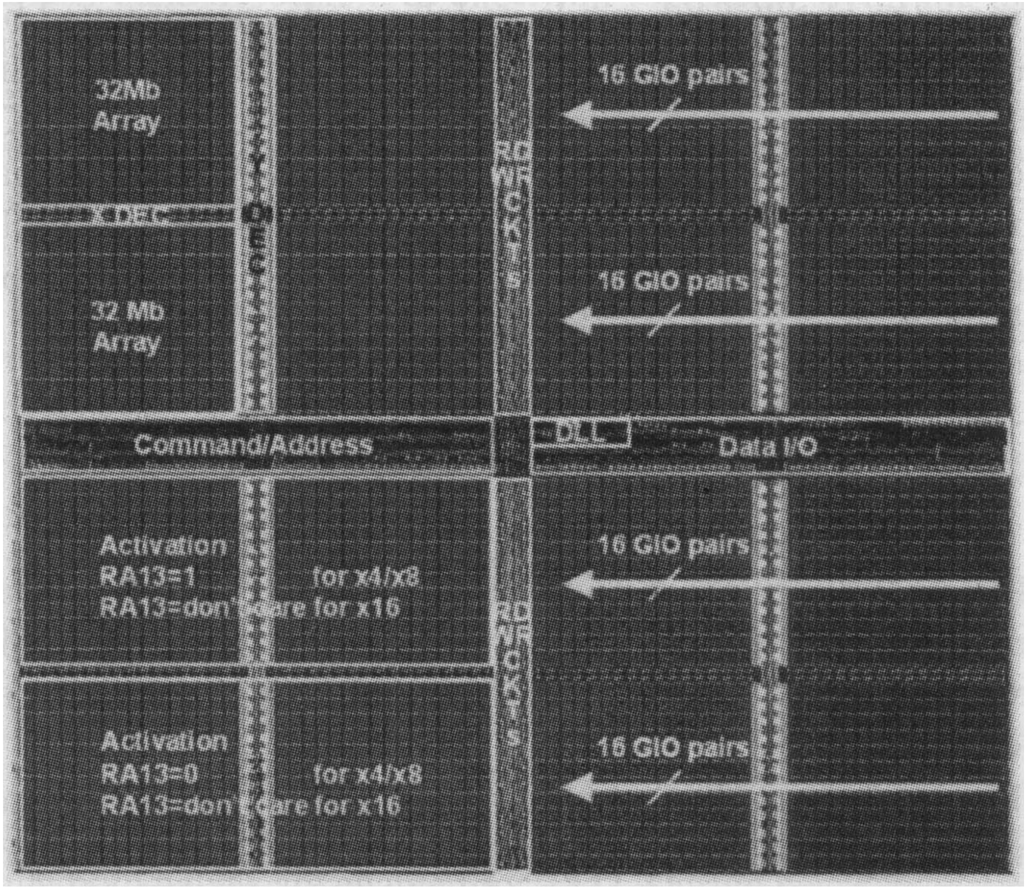
Yes, I'm a hopeless romantic

Base Unit

- Start with DRAM
  - Majority silicon real estate in computer is memory
  - Majority of fab capacity worldwide
  - Incredibly optimized cost-performance
- Don't screw it up
  - Integrate computer without changing process
  - Maintain yield: logic replace I/O area
  - Stay within power envelope

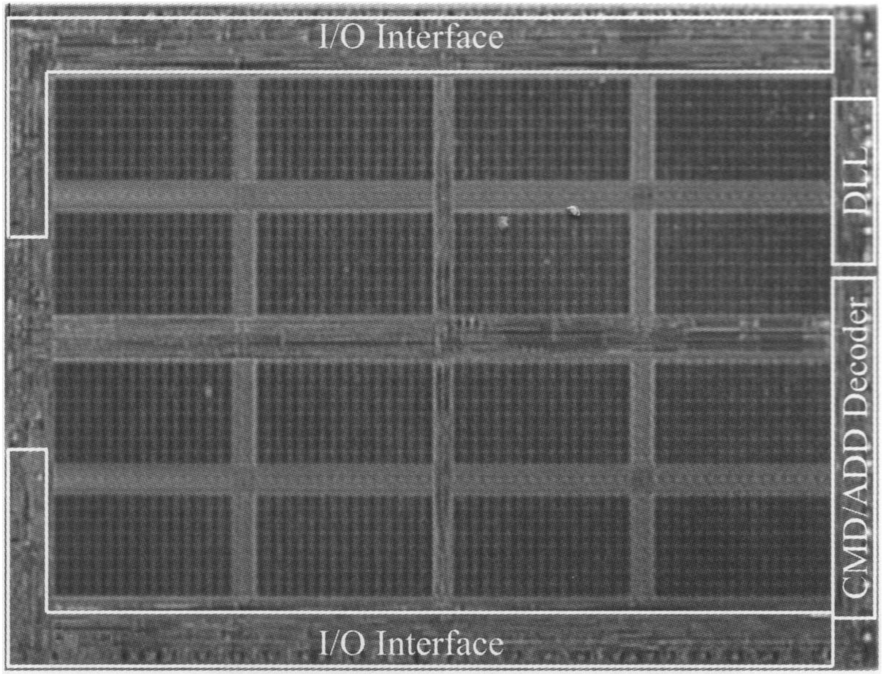
Typical DRAM: Samsung 512 Mb

- 157mm^2@120nm [JSSC]
  - 89mm^2@90nm
- 700 Mb/s ×16
  - I/O 8% (12mm^2)
- Transistors N:P
  - 330:150 µA/µm
  - 640:280 [TSMC 90nm]
  - 535:236 [Charter 0.13]
- 1 tungsten + 2 aluminum



Increasing I/O Bandwidth

- $78mm^2$@110nm [JSSC]
  - 256 Mb “GDDR”
- 1600 Mb/s ×32
  - I/O 21% ($16.5mm^2$)
- Expensive
  - < 50% cell area
  - 84 BGA
  - 1.5 W (I/O 33%)
  - Difficult shrink



Powerful, Efficient as Possible

- Logic 15% die area
  - Including (many fewer) I/O
- Performance -3 generations
  - Transistors 30% speed v. logic process
  - Gate density only 10%
- 65nm DRAM ~ 180nm logic
  - 1 Gbit memory, 1 M gates, 400 MHz, dozen I/O

Cava Architecture

- Vectors
  - “All but 10 parallel apps are vectorizable”
- Symmetric multiprocessor
  - Processor per data lane to memory
- Simultaneous multi-thread
  - No cache, non-speculative, precise exception
- Compact ISA
  - Memory is precious

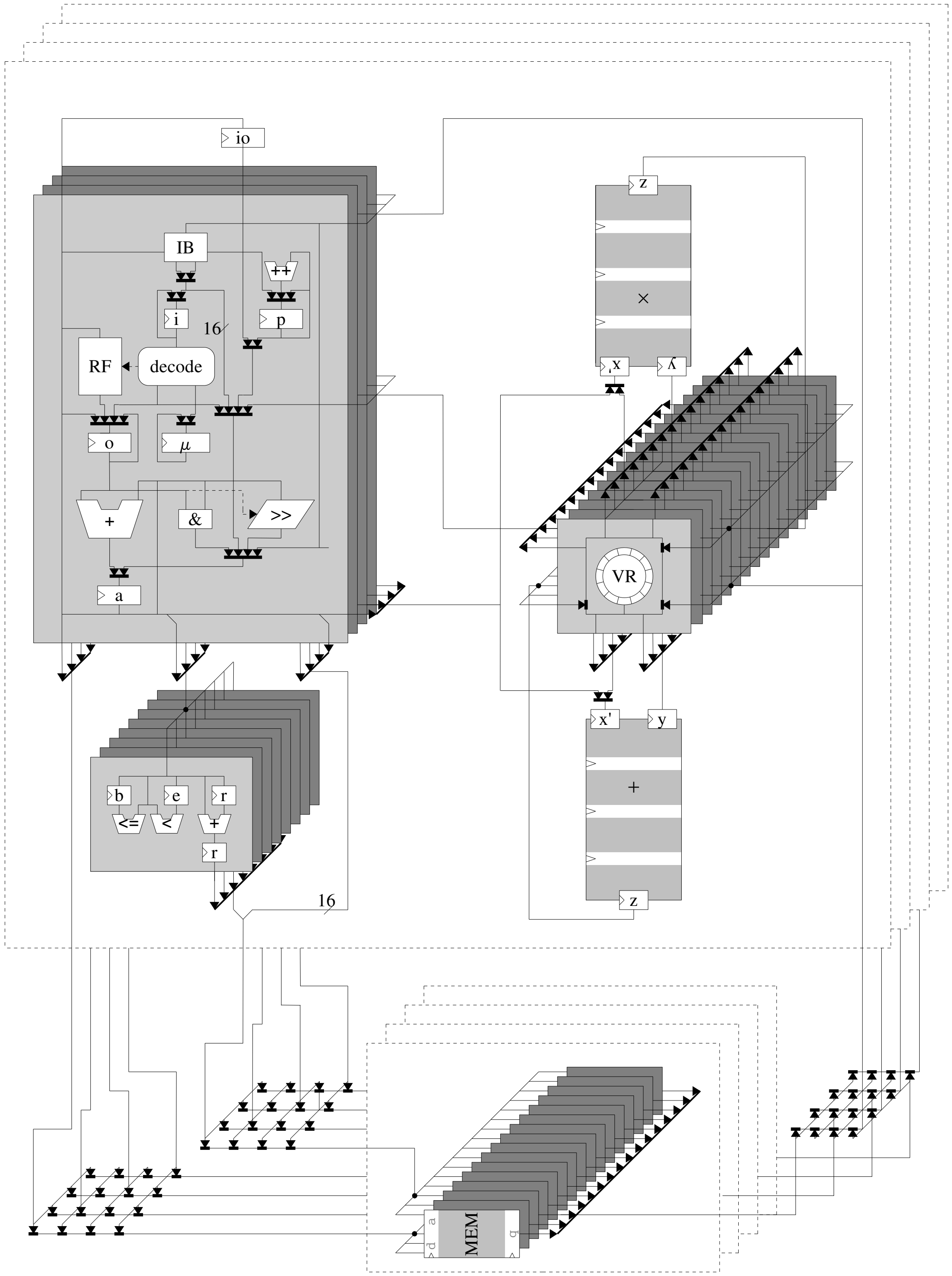
Cava v. CELL-like Stuff

- Base unit granularity
  - Cava: one chip, 3 GFLOPS, 1 W
  - CELL: 1 logic + N memory, 100? GFLOPS, 20? W
- Programming model
  - Cava: single address space, uniform latency
  - CELL: multiple address spaces, unequal latencies
- Business model
  - Cava: successful 3rd world fabs good for humanity

Competitive Landscape

- Power limits DRAM bandwidth
  - Approximately 10 GB/s per 1 W chip
  - Streaming: 5 GFLOPS per DRAM (mul-add/word)
- Maybe doesn't matter how organize
  - Many DRAM/one cpu, several cpu/one DRAM
  - VLIW, superscalar, SIMD, vectors
  - Directory shared memory, message passing, COMA

Production cost, profit margin, critical mass


io
IB
++
i
p
16
RF
decode
o
μ
+
&
>>
a
b
e
r
<=
<
+
r
16
z
×
x
y
VR
x'
y
+
z
MEM
a
d
q

Low Tech Implementation

- Extreme hierarchy
  - Deja vu: MSI chips, small PCB's, wire-or backplanes
- Planar layout
  - Mead & Conway VLSI, linear array multipliers
- Basic circuits
  - Avoid dynamic, low swing, ratioed, SRAM

Engineering skills available worldwide

Interleaved Memory, Naturally

- Large DRAM inherently hierarchical
  - Long wires, many loads: independent segments
  - Timing/noise uncertainties: replicate control
- Power limits concurrency
  - VDD drop, supply noise limit number of active banks
  - Bandwidth/cost already highly optimized

- Classic theory: # banks = latency**2

Performance

- Scalar code
  - Thread IPC ~ 0.25
  - 85% mem utilization
    - 2/3 instructions
    - 1/3 data
- Vector code
  - Mem, add, mul / cycle
  - Cray-1 like

- 30% memory [2.0 CPI]
  - 20% load (7+1.5 cycles)
  - 10% store (3)
- 20% branches [1.1]
  - 7% fall through (2)
  - 7% taken, hit (3)
  - 7% taken, miss (9+1.5)
- 50% ALU [0.6]
  - 40% short (1)
  - 10% long (2)

Gate Count

- Vector unit (1×)
  - FP multiplier 30K
  - FP adder 30K
  - VR 30K (16·32·32·2g/b)
- Thread unit (4×)
  - RF+IB 1.5K (28·32·2g/b)
  - ALU, control 5K (200f/f)
- Address Map (8×)
  - Base, limit, reloc 1.3K (52f/f)
- Memory Port (1×)
  - Drivers, etc. 20K
- Total
  - Processor 150K

Random Thoughts

- Smallest generic base unit?
  - Must run normal OS (Linux, 64+MB, VM)
- Parallel programming language?
  - Never. Naturally parallel apps already happy
- x86?
  - Mandatory L1 cache proxy for larger register file
- Endian?
  - Little. Please...

Conclusion

- Single chip approaching critical mass:
  - Memory capacity (OS, apps, I/O)
  - Performance (clock speed, parallelism)
- Discontinuity in cost/effectiveness
  - Dark horses, white knights, giants tripping...
  - Time to make (or lose) a lot of money
- Will Cava happen?
  - Probably not, but something like it will