CS 294-4 IRAM: Intelligent DRAM
Lecture 1: Introduction
1/17/95

Dave Patterson

Computer Science Division
University of California, Berkeley

http://www.cs.berkeley.edu/~patterson/

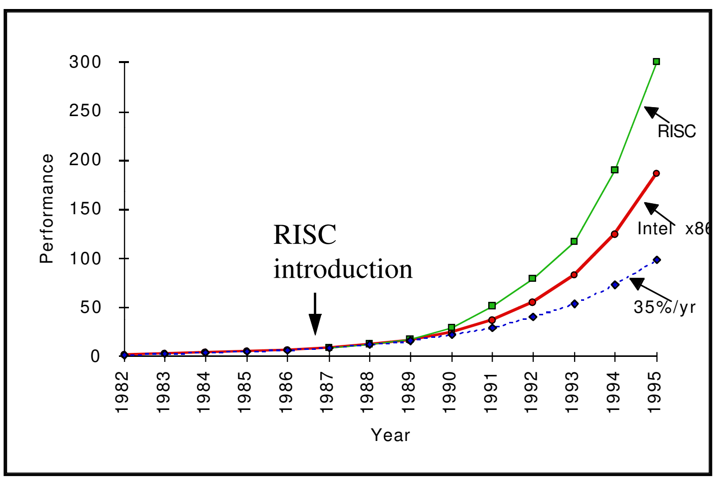
Outline

- Technological motivation for IRAM
- Why IRAM might succeed
- Sketch of IRAM Issues
- CS 294-4 Organization
- List of Topics and Guest Lectures
- Introduction of Classmates
- Handouts
- Initial Discussion

Technological Motivation for IRAMs

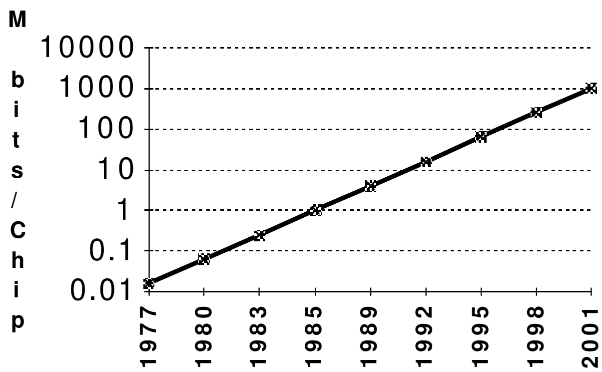
- **Microprocessor- DRAM performance gap**
- **Bridging gap makes HW & SW more complex**
- **Still doesn't work well for some applications: data bases, sparse matrix, ...**
- **Internally, DRAMs have huge bandwidth**
- **Can a simpler processor inside a DRAM overcome the performance gap?**

Processor Performance: 2X / 1.5 Yr.

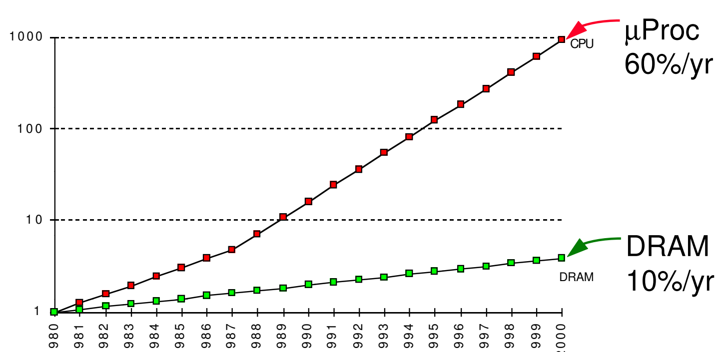


DRAM capacity growth

- **Trend: 4X every 3 years:**



Processor-DRAM Performance Gap



- **1980: no cache (HW controlled buffer) in µprocessor**
- **1995: 2 levels of cache on Alpha 21164 µprocessor**

Gap Causes HW & SW Complexity

- **Processor Complexity**
  - 1980s processors ignore caches, stall on cache miss
  - 1995 processors continue execution during cache miss
  - Instructions execute out of order with scoreboard to keep track of when dependent instructions can proceed
  - Instructions to manipulate cache for correctness/speed
- **Memory System Complexity**
  - Most transistors on microprocessor for caches
  - Expensive external caches (16 MB for some systems)
  - Translation buffer (TLB) misses to access largest cache!
- **Compiler Complexity**
  - Compiler optimization targets memory hierarchy
  - Compiler must reverse engineer cache size, block size so as to produce fast programs

Works poorly for some applications

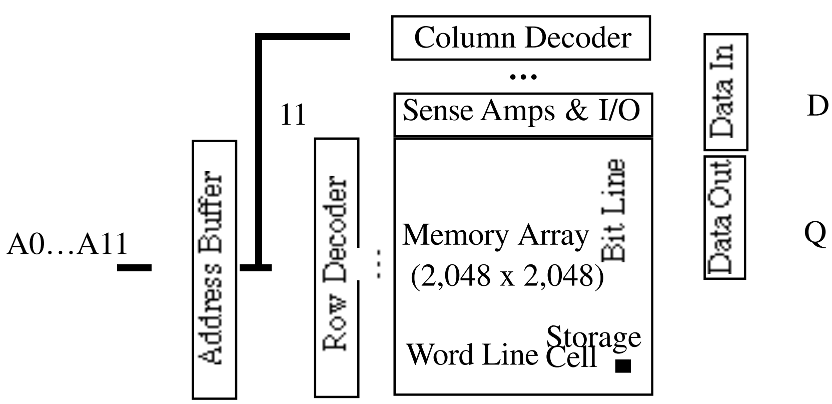
- **Sites and Perl [1996]**
  - Alpha 21164 has 2 levels of cache, total of 112 Kbytes
  - Clock rate of 300 MHz (or 3.3 ns per clock cycle)
  - Can launch 4 instructions every clock cycle ("4-way superscalar")
  - "Many computer customers notice that performance on their programs has a poor correlation with published SPEC performance numbers. Customers and manufacuturers also notice that thier fast microprocessors often spend 75% of all CPU cycles waiting for memory accesses... we believe that [SPEC92] are not very realistic, and that the SPEC95 suite will not be much of an improvement."
  - Running Microsoft SQLserver database on Windows NT operating system, it operates at 12% of peak bandwidth (Clock cycles per instruction or CPI = 2.0)
  - "There is very little correlation between addresses. The implication of this is profound -- caches don't work."

Vector processing better for some apps

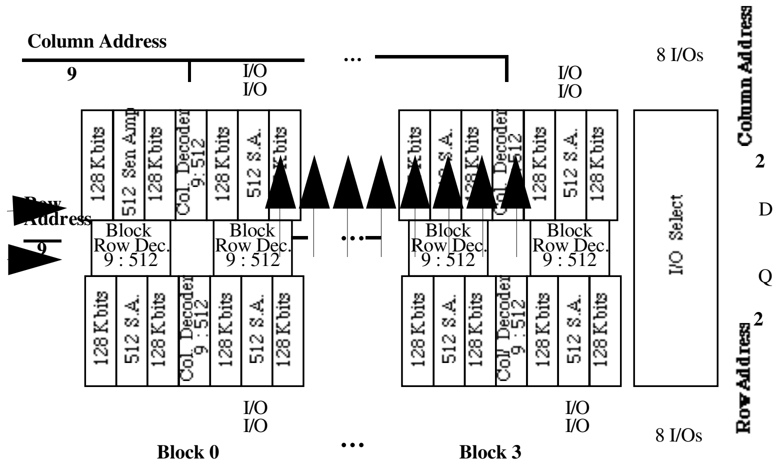
- Cache (1993) vs. Vector (1988)

	IBM RS6000/530	Cray YMP
Clock	71.5 MHz	167 MHz
Cache/Mem	256 KB/512 MB	0.25 KB/256 MB
Linpack	140 MFLOPS	160 MFLOPS
Sparse Matrix	19	163
(Cholesky	16	89
Blocked,	20	168
each line	17	124
a different	20	178
matrix	19	125
structure)	14	57
	17	100
	18	109

DRAM logical organization



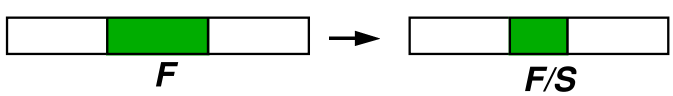
DRAM physical organization



Simpler processor might win? Amdahl's Law

- **Suppose enhancement E speeds up fraction *F* of program by factor *S*:**

$$Speedup_{overall} = \frac{1}{(1 - F) + F / S}$$



or the Law of Diminishing Returns

For example,
if performance is limited by memory speed,
we get little benefit from a faster processor

Simpler processor might win?

- Historical precedent: IBM 360
  - 360/91 with register renaming, out-of-order execution (forerunner of IBM PowerPC 620, MIPS R10000)
  - 360/85 with cache (a better memory system)
  - Clock rate of 91 vs. 85: 60 ns vs. 80 ns
  - Memory speed: 750 ns vs. 1040 ns
  - Memory interleaving: 8-way vs. 4-way
  - 360/85 faster on 8 of 11 programs
- Historical precedent: RISC-I vs. VAX-11/780
  - leave out complex, rarely used instructions
  - making the commonly used instructions run very fast, even if execute more
  - RISC-I built in less expensive technology

Is IRAM a New Idea?

- Graphics accelerators: DRAM + small logic
  - 3D DRAM from Sun/Mitsubishi (1995)
- Processor with SRAM
  - Caches on microprocessors (1980s) (SRAM 4X to 6X more area per bit than DRAM)
- Logic in Memory
  - Associative processing: all elements search for match (e.g. tag of TLB) in 1950s
  - Array processing/Single Instruction Multiple Data (SIMD): mix ALUs with memory elements in 1960s
- Parallel processing building block: processor + SRAM + network interface
  - Inmos Transputer (1978)
  - Bill Dally of MIT J-machine (1988)

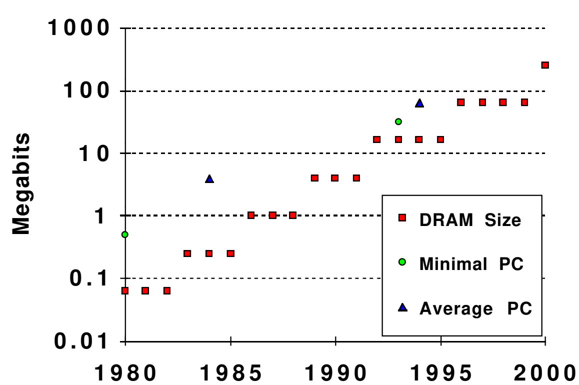
Why might IRAM succeed this time?

- Expensive Contortions of Current Systems
  - Simpler processor with huge bandwidth better performance and much better cost-performance?
- Past efforts memory limited => multiple chips per system => solve parallel processing 1st
  - Gigabit DRAM => 128 MB; sufficient for many programs?
- DRAM manufacturers need a new standard
  - Past efforts built own memory or used separate DRAM; now can get attention of DRAM manufacturers?
- Embedded applications offer large second target to conventional computing

Megabits per DRAMs and PC

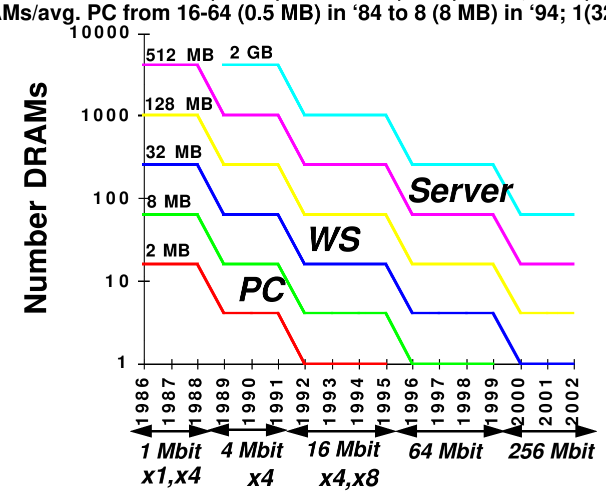
Minimal PC from 64 KB in '80 to 4 MB in '93; 16 MB in '00?

Average PC from 0.5 MB in '84 to 8 MB in '94; 32 MB in '00?



DRAMs per System (Based on Figure 2-7 and 2-8, Przybylski 1994)

DRAMs/min. PC from 32 (64 KB) in '80 to 2 (4 MB) in '93; 0.25 (8 MB) in '00?
DRAMs/avg. PC from 16-64 (0.5 MB) in '84 to 8 (8 MB) in '94; 1(32 MB) in '00?



IRAM Research Agenda

Chip design
- Speed, area, power of processor in DRAM process?

Instruction Set Architecture for IRAM
- Changing instruction set and multiple IRAM programs

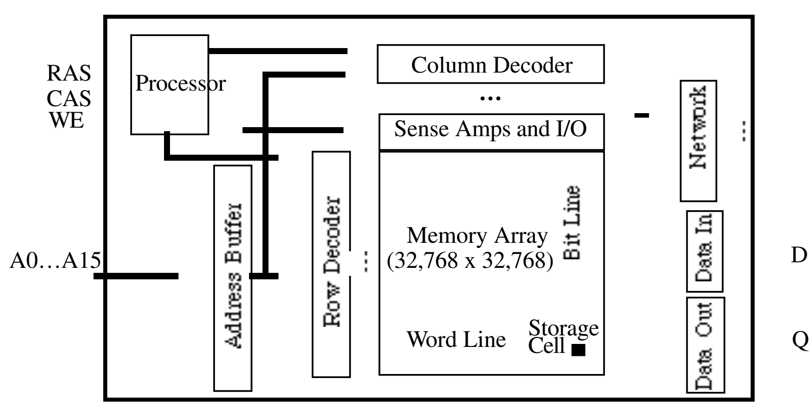
Compiler/Runtime control
- Explicit memory management: Protection, Paging, "Cache", Vector, Interrupts?

Applications
- Number crunching: Sparse Matrix (gather/scatter)
- Real-time/ Embedded: predictable performance, optional general purpose features

IRAM Chip Design Research Agenda

- How much larger and slower is a processor designed in a straight DRAM process vs. a standard logic process
- Speed of page buffer vs. registers, cache
- New DRAM interface based on fast serial links (2.5 Gbit/sec or 300 MB/sec per pin)
- Quantify Bandwidth vs. Area/Power tradeoff
- Area overhead for IRAM vs. a DRAM
- Extra power dissipation for IRAM vs. a DRAM
- Performance of IRAM with same area and power as DRAM ("processor for free")

Potential IRAM logical organization



IRAM Architecture Research Agenda

- How much slower can a processor with a high bandwidth memory be and yet be as fast as a conventional computer?
- Compare memory management schemes (e.g., vector regs, scratch pads, wide TLB/cache)
- Compare schemes for running large programs, i.e., span multiple IRAMs
- Quantify value of compact programs and data (e.g., compact code, on-the-fly compression)
- Quantify pros & cons of standard instruction set vs. custom IRAM instruction set

IRAM Compiler Research Agenda

- Explicit SW control of memory management vs. conventional implicit HW designs
  - Protection (software fault isolation)
  - Paging (dynamic relocation, overlap I/O accesses)
  - "Cache" control (vector register, scratch pad)
  - I/O interrupts/polling
- Evaluate benchmark performance in conjunction with architecture research
  - Number crunching (Vector vs. superscalar)
  - Memory intensive (database, operating system)
  - Realtime benchmarks (stability and performance)
  - Pointer intensive (GCC compiler)
- Impact of Language on IRAM (Fortran77 vs. HPF, C/C++ vs. Java)

IRAM Research Opportunity: Quantity Cost/Performance Spectrum

Evolutionary

x86 + cache to DRAM via 1000s wires (flip chip)

x86 + cache in DRAM process

x86 + novel memory architecture in DRAM process

x86 + vector in DRAM process

New ISA + cache in DRAM process

New ISA , replace protection with compiler check

New ISA , replace paging with compiler control

New ISA , replace cache with compiler control

New ISA , replace interrupts with compiler control

New ISA + FPGA to tailor IRAM logic

Revolutionary

IRAM Conclusion

- **When change HW/SW interface?**
  - address limit (e.g., 16 to 32)
  - new technology (e.g., microprocessors)
  - Ignoring IRAM, future is x86 compatible, complex processors with 100Ms transistors for cache
  - IRAM best chance in next 10 years to redefine HW/SW interface?
- **IRAM rewards creativity as well as manufacturing, opportunity to shift balance of power in commodity DRAM industry**
- **If successful, potential impact in 6 to 8 years as large as RISC or RAID?**

CS 294-4 Course Organization

- **Modeled like RISC courses**
  - Real mix of EE and CS students
  - Lots of meetings early to gain background in less elapsed time (see if can find another hour/week)
  - Project meetings and results in the middle to exchange ideas, track progress
  - Final presentations to class+ final report on WWW so we all know state of world
- **Periodic discussions to gauge ideas**
- **Initial small assignments as test cases (2-3 short assignments, results on WWW)**
- **Guest experts early so as to learn the real stuff (lecture notes on WWW by students)**
- **Design review by DRAM/logic experts**
- **Last lecture: where to go from here**

CS 294-4 Part of Course Sequence

- **CS 294-4 (Patterson) Spring 1996 architecture/compiler/chip investigations**
- ***CS 250 (Sachs) Spring 1996 Learn VLSI, CAD tools***
- **CS 254 (Wawrzynek) Fall 1996 initial chip designs**

Lecture Topics

- DRAM Introduction/Review
- Memory Hierarchy Review
- Vector Introduction/Review
- Memory Hierarchy and Compilers
- Software Fault Isolation
- Issues in Real-Time Computing
- Review of Embedded Computers
- IRAM Interface: Serial Lines, Network
- HW/SW for programs that span multiple IRAMs
- Read papers, hear guest lectures, discuss topics

Example Guest Lectures

- DRAM fundamentals
  - Fri Jan 19: Steve Przybylski (original MIPS student, DRAM consultant)
- Example of DRAM and Logic
  - Wed Jan 24: Mike Deering of Sun Microsystems (UCB alumnus, architect of custom DRAM for graphics)
- DRAM interfaces/Fast Serial Lines
  - Fri Feb 2: Mark Horowitz of Stanford University (A founder of RAMBUS, custom DRAM with fast bus)
- Vector Computing
  - ?? of Fujitsu America (architect of VP300)
- Possibilities
  - Andreas Nowatzyk of Sun on IRAM multiprocessor
  - Peter Kogge of Notre Dame on IRAM multiprocessor
  - Steve Lucco of Colusa Systems on SW fault isolation

Introduce Classmates

- Name, year in grad school, area of reserach interest (if known), advisor (if known)
- Background that may be relevant (course, research and work experience)
- IRAM areas that are not a strength
- Planning to take CS294-4 for credit, planning to audit, still deciding?
- What you would like to see in this course, suggestions?

Handouts and Suggestions

- Initial Reader
  - Scientific American as overview for people of all backgrounds
  - Steve P.'s Introduction to DRAM basics (for Friday)
  - Mike Deering's paper on Frame Buffer DRAM (for Wed)
  - Mark Horowitz's paper on fast serial lines
- Recommendations for class?