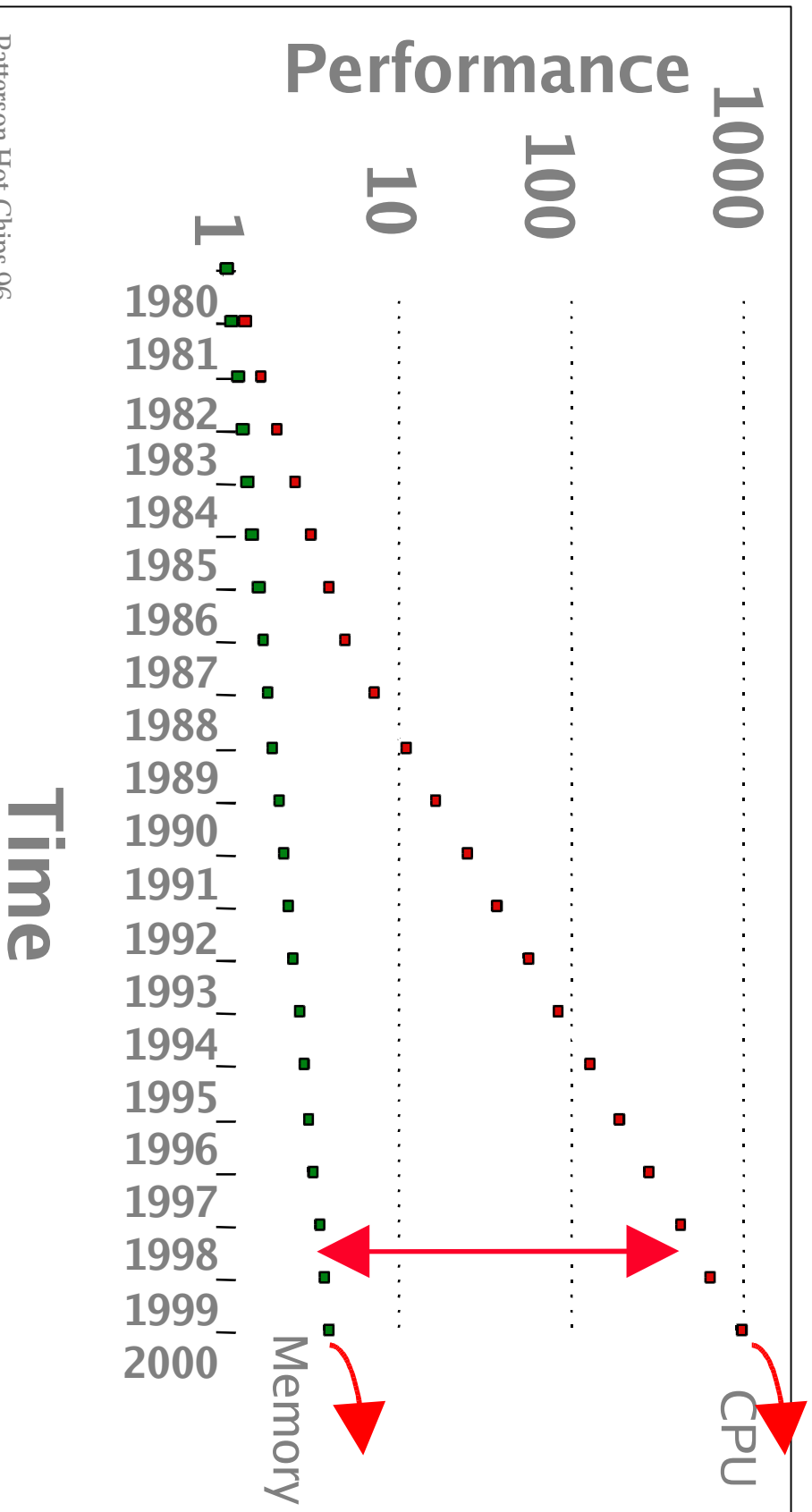# What / Why is memory?

- What
  - Preservation of state
  - Place(n) to place(v) your data
  - Communication mechanism
  - Separates computers from PDA
- Why
  - Time-space trade off

What is the difference between a register and memory?

- Memory gives us indirection

Processor-Memory Gap

Performance

1000

100

10

1

1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000

Time

Memory

CPU

What is the memory hierarchy and why?

- What
  - From disk to registers
    - Registers
    - Level 1 cache
    - Level 2 cache
    - Level 3 in some systems
    - DRAM
    - Disk
    - Diskarray / removable device / Web storage
- Why
  - Time econmic tradeoff

A Cache is a bet..

- What do you lose?
  - $
  - Look ahead grabbing more data than needed
  - Increased worst-case
- What can you win?
  - Speed in the common case

Stream buffers – implicit prefetching

- Acquire next n lines speculatively
- Prefefch around last cache line
  - Next n cache lines
  - Stride prefetching
  - Markov model prefetching
- Multiple stream buffers

Cache size/speed tradeoff
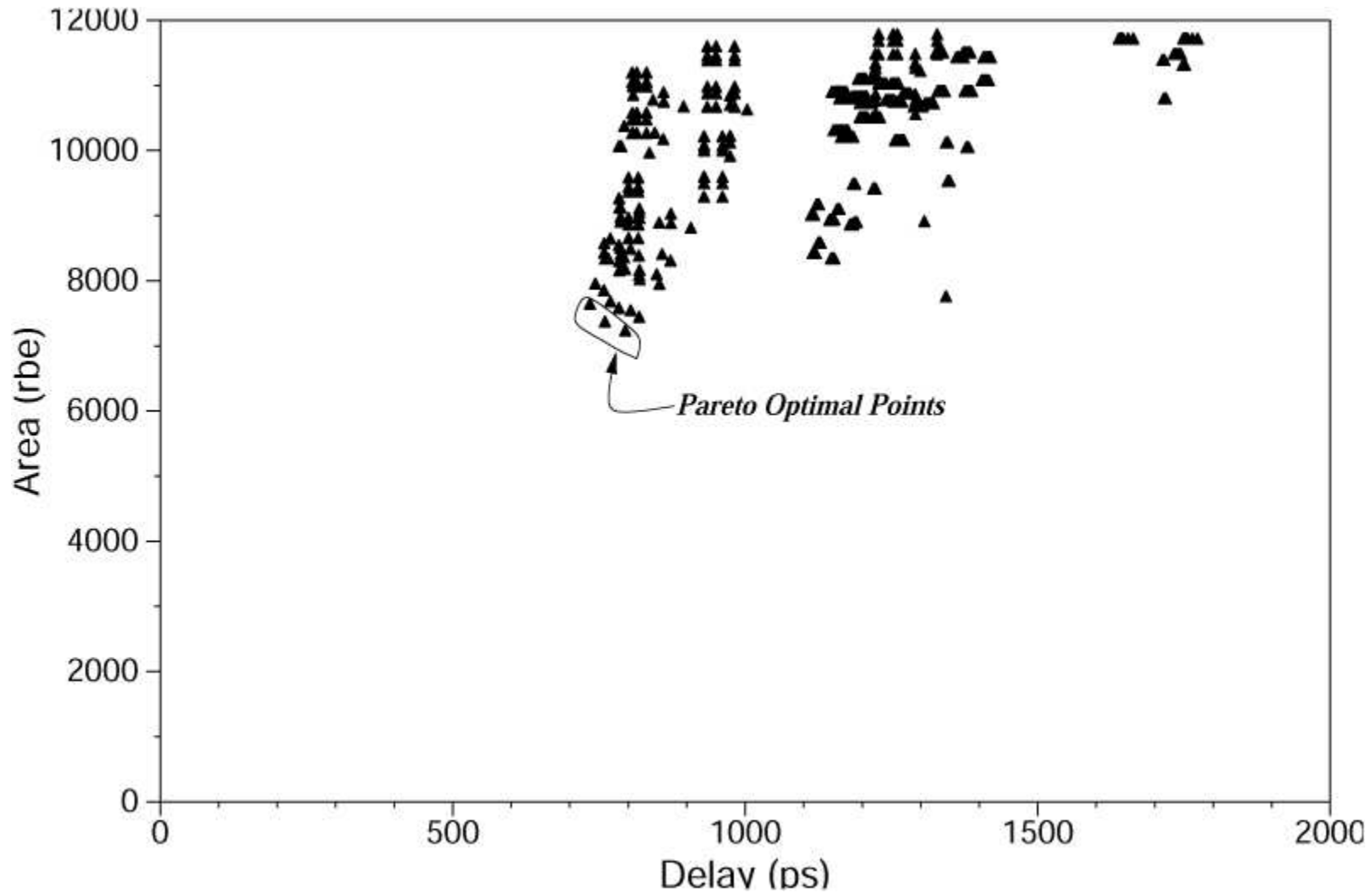
Where are we going from here?

- Problems
  - Smoother degradation L1/l2/l3/l?
  -

- Solutions
  - Faster memory technology
  - Compiler/programming prefetching
  - Cache conscious algorithms
  - Better prefetching algorithms
    - Fuzzy logic / adaptive

- for(i=0; i<= assoc; i++)
  - a[i*huge#]++

(swiped from Luna)

Don't Need to Pay Much for Your Miss (II)
-- Lockup-free Instruction Fetch/Prefetch

if (catch-hit)
    get-from-cache
else if (catch-miss){
    Judge the miss states from MSHR (in-input-stack indicator, partial write codes, valid indicator);
    if (totally written)
        read from cache;
    else if (in-input-stack)
        read from input stack;
    else if (partially written || already-asked-for)
        by-pass;
    else{
        initiate MSHR;
        when data available do 1, 2, 3
parallely; }

1. if (send-to-CPU)
        send to CPU;
3. if (!totally written || !MSHR obsolete){
        if (input-stack full)
            FIFO remove one;
        write to input-stack;
        set MSHR.in-input-stack;
}
9. write to catch and set MSHR.partial-write-code
if (written || obsolete MSHR)
        MSHR.num-of-words-processed++;
if (MSHR.num-of-words processed overflow)
        clear MSHR.valid-indicator;