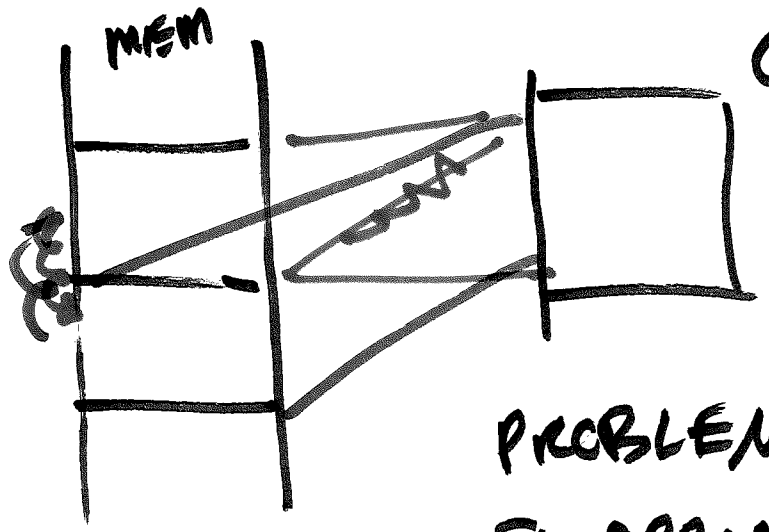
CS 451 / 551 / ECE 541

ADVANCED COMPUTER ARCHITECTURE

SESSION no. 12

DIRECT-MAPPED



PROBLEM: EXCESSIVE SWAPPING AT BLOCK BOUNDARIES.

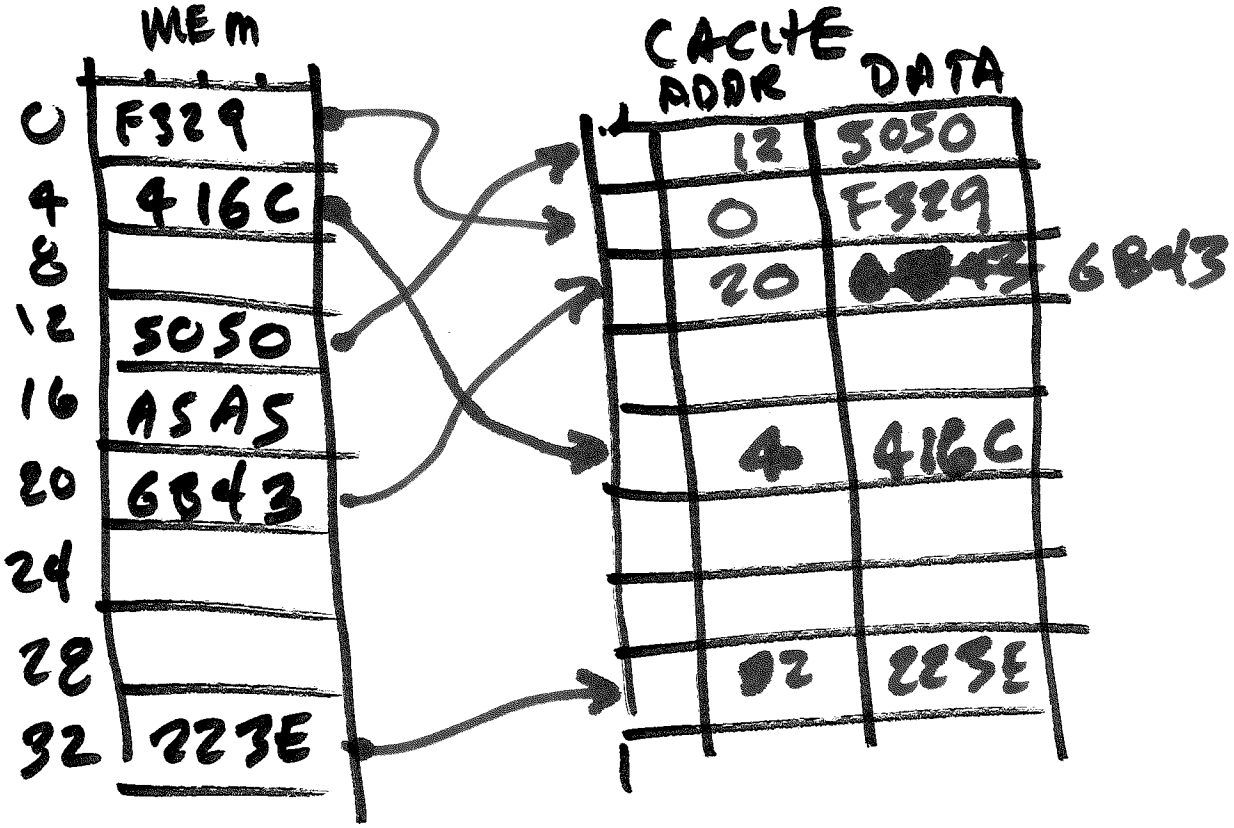
University of Idaho

DISADVANTAGES (CHALLENGES)

1. MAPPING ADDRESSES - COMPLICATED
   UNIQUE TAG FOR EACH MEMORY LOCATION
   SEARCH: ADDRESS = KEY
   PARALLEL
   e.g. 512 CACHE ENTRIES =>
   512 COMPARATORS + LOGIC
   CONTENT-ADDRESSABLE MEMORY (CAM)
2. LOSE ADVANTAGES OF BLOCK TRANSFER
   EACH CACHE TRANSACTION (MOVE) IS ONE WORD.

~~DIRECT - MAPPED~~

FULLY - ASSOCIATIVE CACHE



16 bit mem

ADVANTAGE: AVOIDS BOUNDARY PROBLEM

3. IN REPLACING A CACHE ITEM — WHICH ITEM TO REPLACE?
"REPLACEMENT STRATEGY"

A. LRU - LEAST RECENTLY USED.
. EXPLOITS TEMPORAL LOCALITY
. LOSES SPATIAL LOCALITY

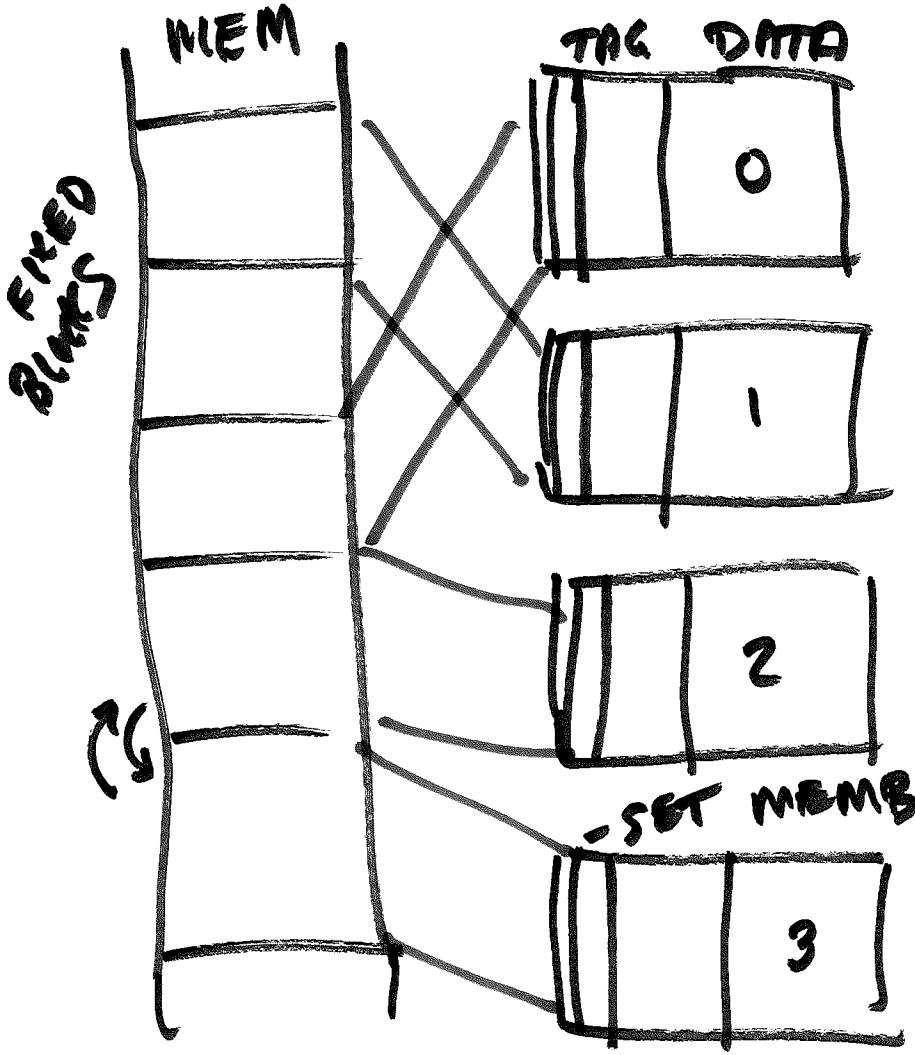
Q: HOW TO KEEP TRACK OF ORDER IN WHICH CACHE ENTRIES WERE USED?
PRIORITY SEARCH.
. HARDWARE COST ↗

LINEAR FEEDBACK SHIFT REGISTER (LFSR)

B. RANDOM - LOSE TEMPORAL LOCALITY
BUT ~~SIMPLE~~ SIMPLE

University of Idaho

SET - ASSOCIATIVE ~~CACHE~~ CACHE



4-WAY SET ASSOCIATIVE

RESOLVE BOUNDARY PROBLEM

HARDWARE MUCH SIMPLER

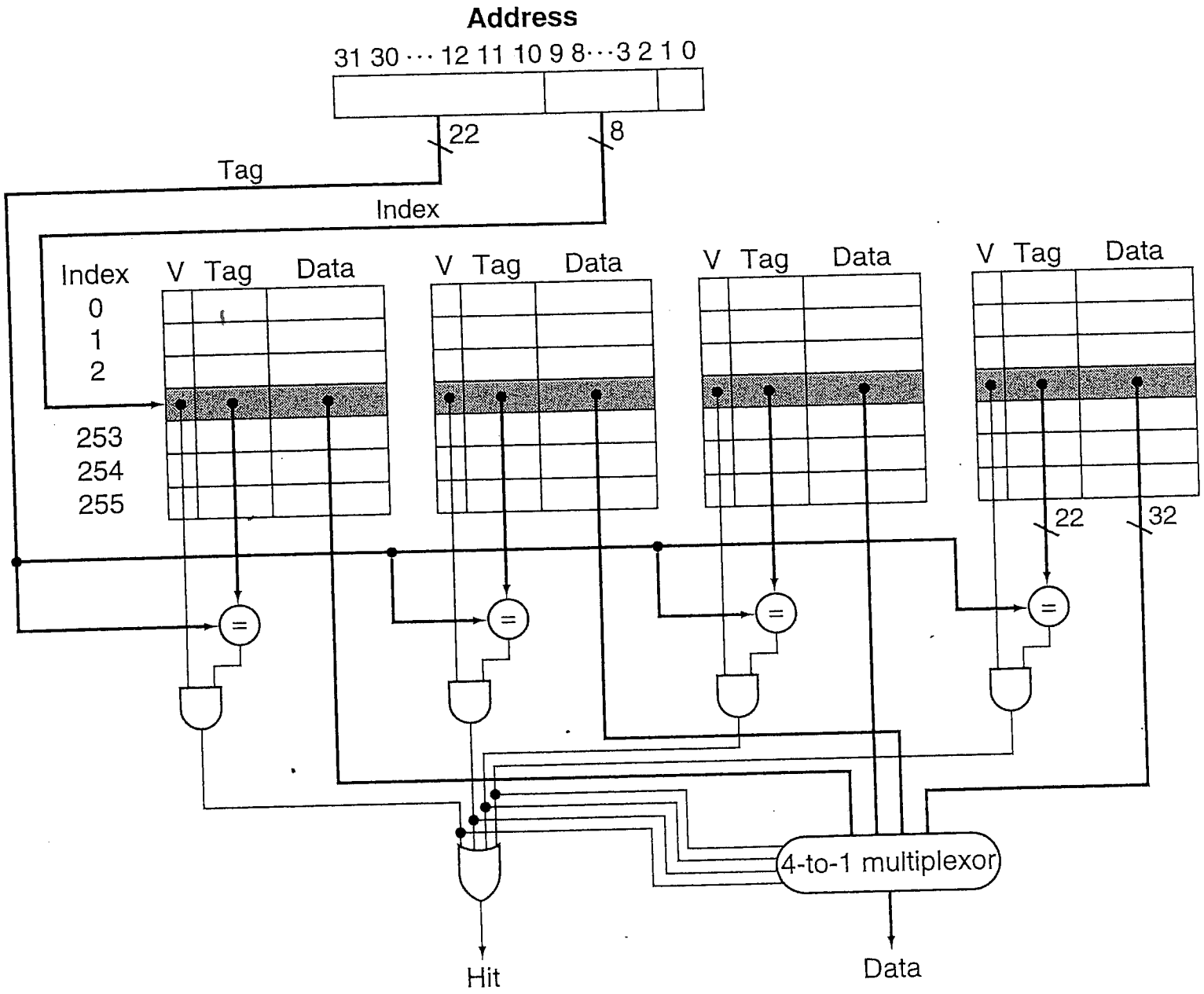
. 4 COMPARATORS

. REPLACEMENT STRATEGY

. LRU

· BLOCK TRANSFER

· SPATIAL LOCALITY


Address
31 30 ··· 12 11 10 9 8···3 2 1 0
22
8
Tag
Index
Index
V Tag Data
V Tag Data
V Tag Data
V Tag Data
0
1
2
253
254
255
22
32
=
=
=
=
4-to-1 multiplexor
Hit
Data

University of Idaho

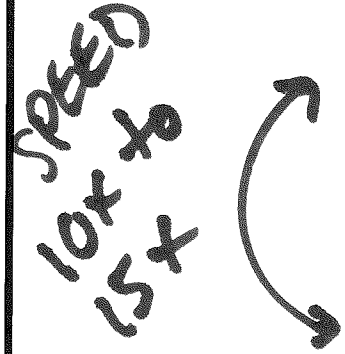
CACHE PERFORMANCE

PURPOSE OF CACHE

GOAL: SPEED AVG MEMORY ACCESS

CONSTRAINT: AFFORDABLE COST

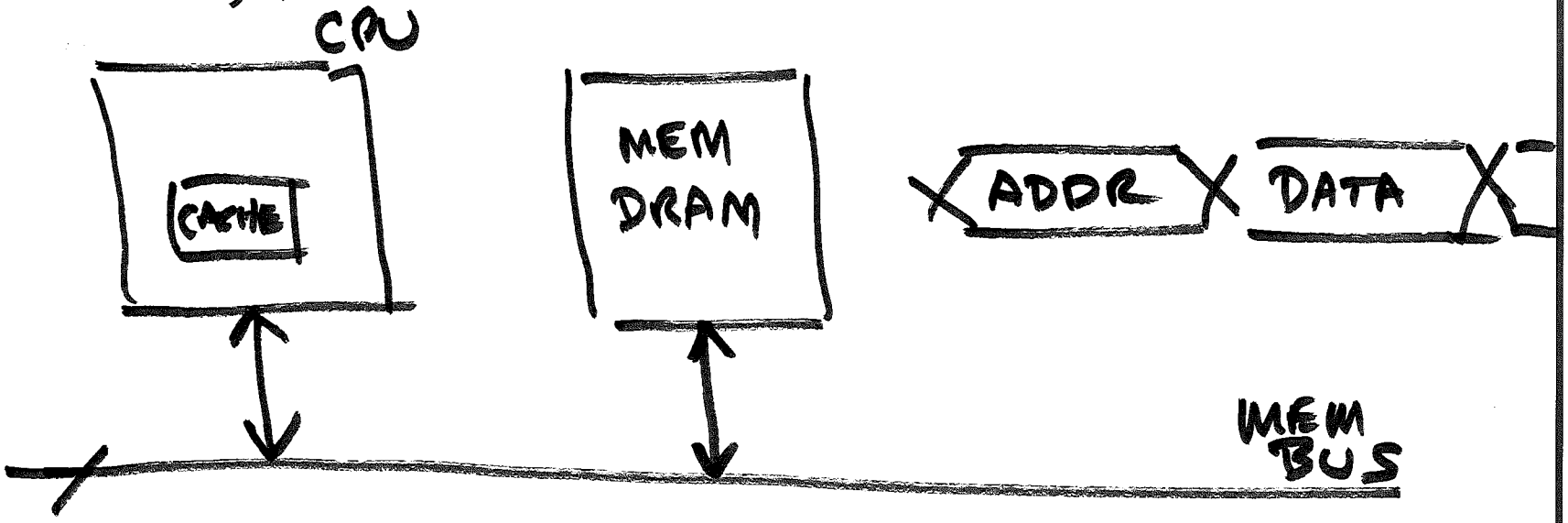
LOCALITY - SPATIAL & TEMPORAL



MAIN MEM: DRAM
Cheap, large capacity, slower

CACHE: SRAM
Expensive, small cap, faster

University of Idaho



MISS RATE = 1 - HIT RATE

- FRACTION OF TIMES DATA NOT FOUND IN CACHE

MISS PENALTY = PERFORMANCE COST IN CACHE MISS

UNITS: MEMORY (SYSTEM) BUS CYCLES
SLOWER THAN CPU CYCLES

University of Idaho

ASSUME: 4 WORD CACHE BLOCK (LINE)
WORD ACCESS, 4 BYTES / WORD

MISS PENALTY

MP = 1 + 4 x 15 + 4 x 1 = 65 BUS CYCLES

- 1: ADDR TRANSFER
- 15: 15 CLKS
- 4 x 15: LATENCY / WORD
- 4 x 1: 4 WORDS
- 4 x 1: DATA TRANSFER

BYTES TRANSFERRED

BANDWIDTH MISS: $\frac{4 \times 4}{65} = .25$ BYTES / CYCLE

HOW TO IMPROVE THIS?

1. MAKE MEMORY SYSTEM WIDER

   Eg. BUS IS 2 WORDS WIDE

MISS PENALTY

~~MP = 1 + 4X~~

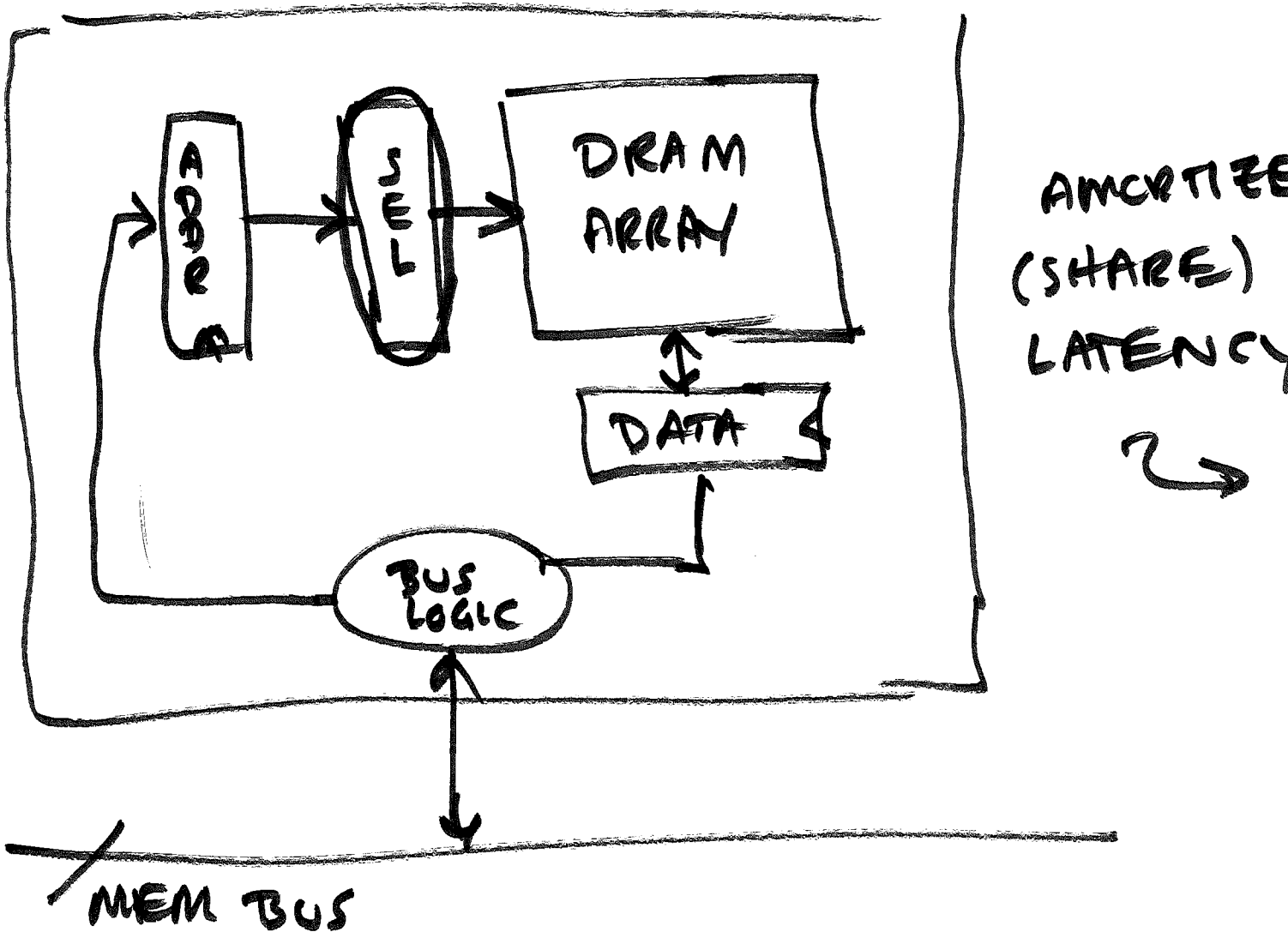
MP = 1 + 2×15 + 2×1 = 33 BUS CYCLES

ADDR, DATA, LATENCY, DATA

BANDWIDTH: $\frac{4 \times 4 \text{ BYTES}}{33 \text{ CYCLES}} = .48 \frac{\text{BYTES}}{\text{CYCLE}}$

COST: AREA (INTERCONNECT)

2. MEMORY BANKS - INTERLEAVING



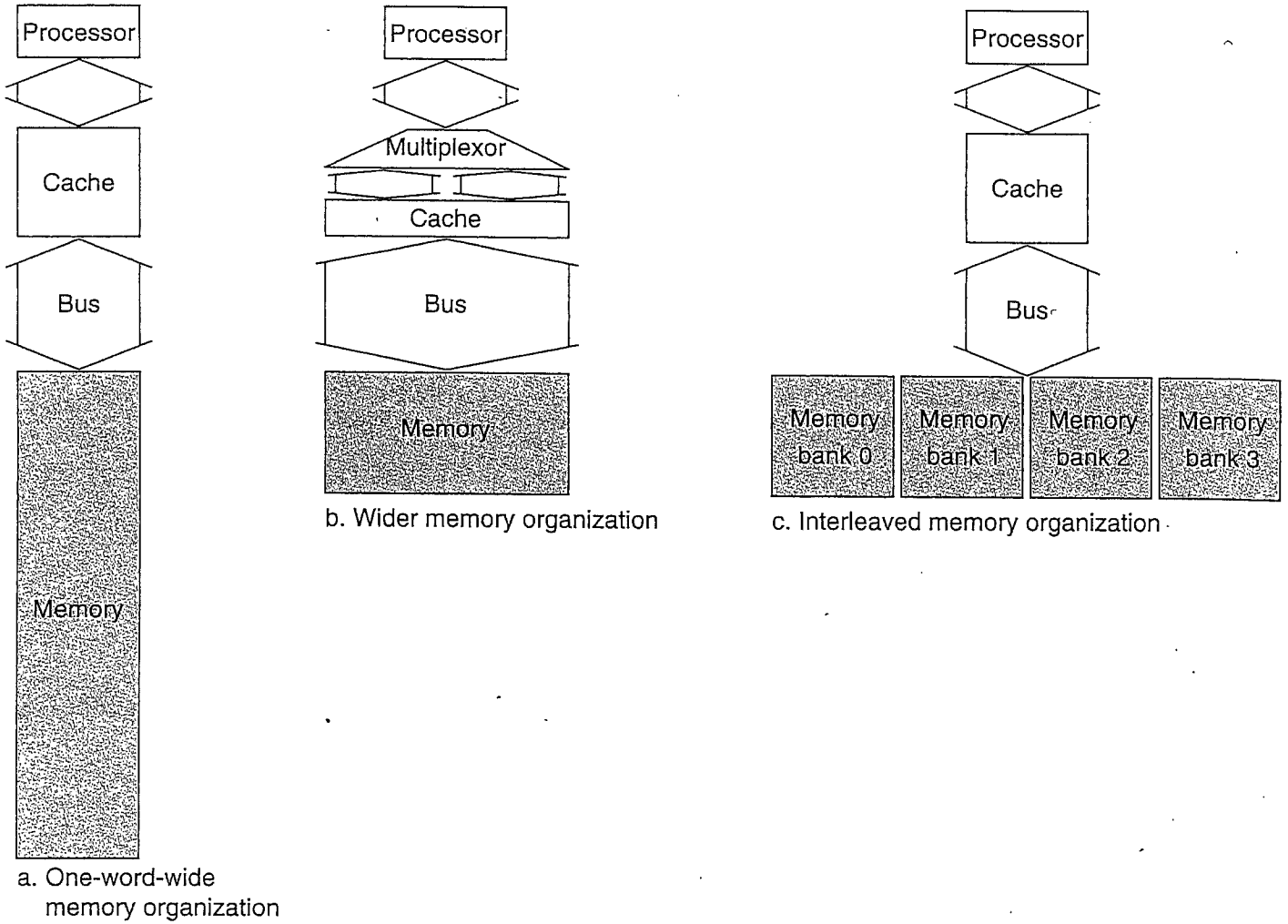


FIGURE 5.11 The primary method of achieving higher memory bandwidth is to increase the physical or logical width of the memory system. In this figure, memory bandwidth is improved two ways. The simplest design, (a), uses a memory where all components are one word wide; (b) shows a wider memory, bus, and cache; while (c) shows a narrow bus and cache with an interleaved memory. In (b), the logic between the cache and processor consists of a multiplexor used on reads and control logic to update the appropriate words of the cache on writes.

MEMORY BANKS

- EACH MEM MODULE HOLDS A DIFFERENT WORD IN CACHE LINE
- SAME ADDRESS (+ OFFSET)
- ALL MODULES ACCESSING INTERNALLY SIMULTANEOUSLY

MP = 1 + 1 × 15 + 4 × 1 = 20 BUS CYCLES

University of Idaho

MEASURING CACHE PERFORMANCE

CPU TIME:

(CPU CYCLES + MEM STALL CYCLES) X

CLOCK CYCLE TIME

- NO CACHE MISS => NO STALL CYCLES

IF → DEC → EX/ADDR → MEM → WB

ASSUME READ & WRITE HAVE SAME COST

MEMORY STALL CYCLES =

$$\frac{\text{MEMORY ACCESSES}}{\text{PROGRAM}} \times \text{MISS RATE} \times \text{MISS PENALTY}$$

MEM BUS CYCLE = n × CPU CYCLES

EX ASSUME INSTRUCTION CACHE & DATA CACHE

INSTR CACHE MISS RATE: 2%
DATA CACHE " " : 4%

CPI = 2 CLOCKS / INSTR.

MISS PENALTY: 100 CPU CYCLES

$\vec{I}$ = INSTR. COUNT FOR PROGRAM

INSTR MISS CYCLES: $I \times 2\% \times 100 = 2 \times I$

DATA MISS CYCLES: $I \times 36\% \times 4\% \times 100$
$= 3.44\, I$

LOADS & STORES ARE 36% INSTRUCTIONS

> 3 CYCLES OF STALL / INSTR.

ACTUAL CPI = 2 + 3.44 = 5.44

2 — IDEAL CPI

3.44 — PENALTY

PERFECT CACHE? NO MISSES

$\frac{5.44}{2} = 2.72$ × BETTER PERFORMANCE THAN REAL CACH W/ STALLS.

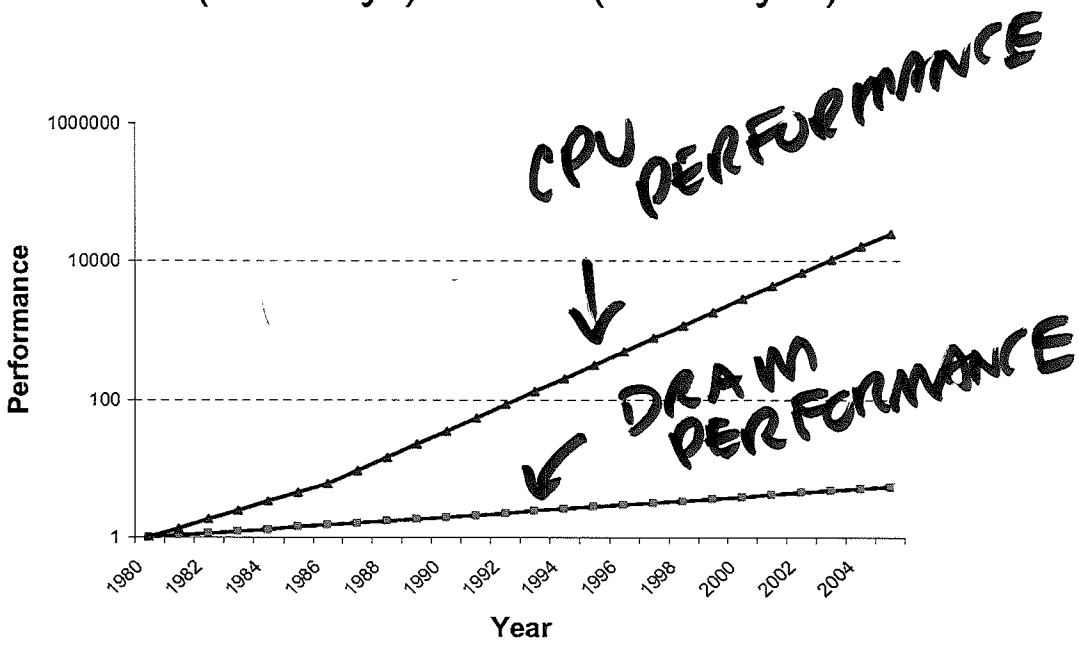
DRAM Access Latency

- Access times are a speed of light issue
- Memory technology is also changing
  - SRAM are getting harder to scale
  - DRAM is no longer cheapest cost/bit
- Power efficiency is an issue here as well

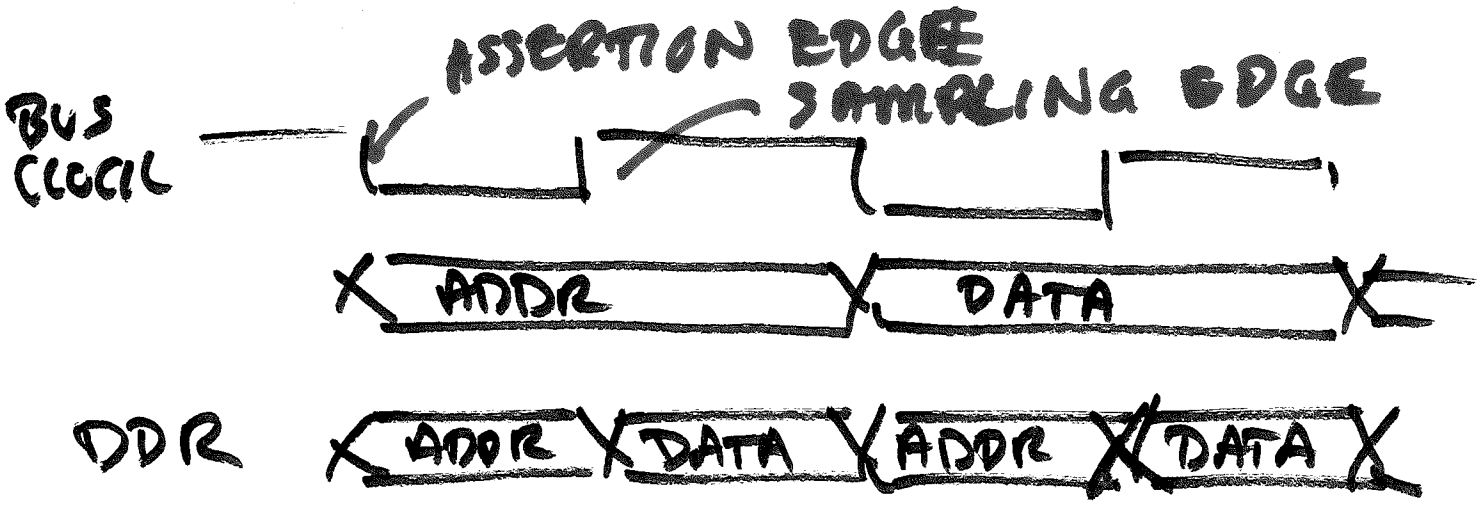
Images removed due to copyright restrictions.

µProc
60%/yr.
(2X/1.5yr)

DRAM
9%/yr.
(2X/10 yrs)



DDR: DOUBLE RATE DRAM



DDR2, DDR3 - JEDEC STANDARDS

SDRAM -

- DRAM WITH A WRAPPER TO MAKE IT BEHAVE EXTERNALLY LIKE SRAM
- DRAM DENSITY, SRAM SIMPLICITY