CS 451 / 551 / ECE 541

ADVANCED COMPUTER ARCHITECTURE

SESSION no. 17

University of Idaho

B.S. PROTECTION

PROCESS

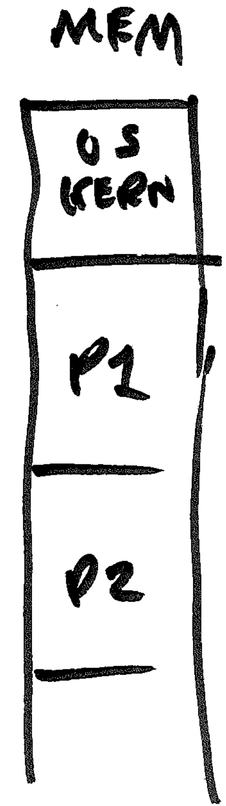
OLD DAYS: 1 PROCESS // JOB
OPERATOR STARTS & STOPS

THEN: TIME SHARING
- MULTIPLE USERS, "TIME SLICE"
- APPEARS SIMULTANEOUS, BUT
  - MULTITASKING
  - CONCURRENT
- EACH USER = A PROCESS
  - CAN SPAWN CHILD PROCESSES
- O/S ALSO PROCESSES

PROCESS ABSTRACTION: SUPER USEFUL

EACH PROCESS HAS <u>CONTEXT</u>:

- PC
- STACK
- STATUS (N, Z, V, C)



MULTIPLE PROCESSES IN MEMORY AT ONCE

DIFFERENT FROM <u>THREAD</u>:

- SHARED MEMORY (ADDRESS SPACE, CONTEXT)

EACH PROCESS & THREAD MUST RUN CORRECTLY IF INTERRUPTED.

University of Idaho

MEMORY PROTECTION

1) KEEP PROCESSES FROM INTERFERING -
   - OVERWRITING EACH OTHER'S MEMORY

ARBITRATE RESOURCES

2) SECURITY - PREVENT EXTERNAL AGENTS FROM MODIFYING RUNNING SYSTEM

University of Idaho

EACH PROCESS "OWNS" SOME PHYSICAL MEMORY

SIMPLE: USER SPACE - USER ACCESS PRIVILEDGES

SYSTEM (SUPERVISOR, KERNEL) SPACE

SYSTEM ACCESS PRIVELEGDES

COMPLEX: MULTILEVEL PROTECTION

INTEL: "RINGS" IA-32

LEVEL 0: SYSTEM, MOST RESTRICTED

LEVEL 1:

LEVEL 2:

LEVEL 3: USER, USER SPACE

BASICS OF MEMORY PROTECTION

- TEXT i86
- CONCEPTUALLY: LIKE FILE ACCESS IN UNIX

chmod:	user	group	world
	wrx	wrx	wrx ?
	root		

- GLOBAL ADDRESS SPACE -

  SHARED - ACROSS PROCESSES

- LOCAL ADDRESS SPACE

  ACCESSABLE BY "OWNER",

  SELECT OTHERS WITH HIGHER

  PRIVELEDGE

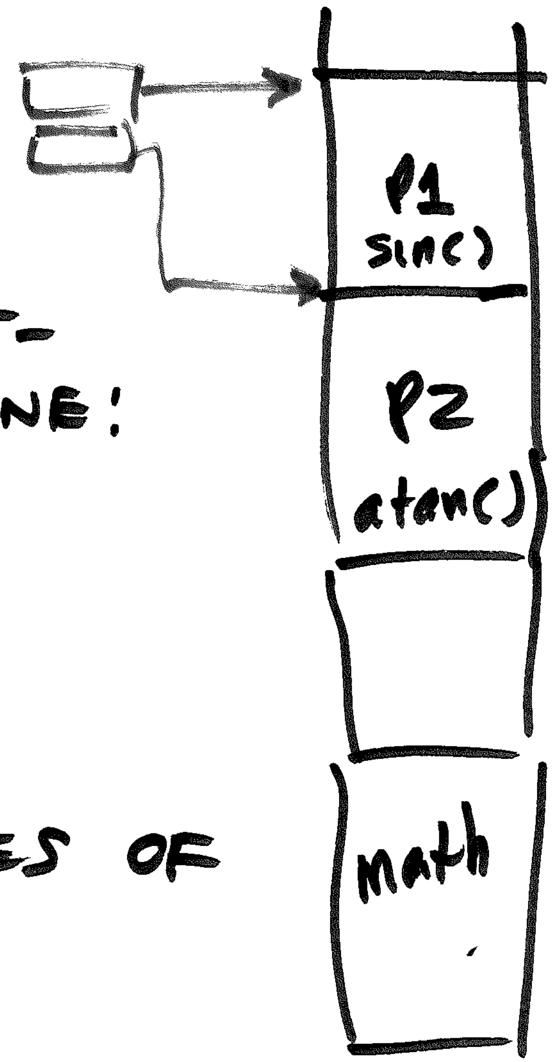
University of Idaho

ENFORCE IN HARDWARE

- SETS OF REGISTERS
- "WATCHDOG" CIRCUITS

ATTEMPTED ACCESS OUT-SIDE PRIVILEDGE ZONE!

"EXCEPTION"



SOME COMPLICATIONS

- SHARED LIBRARIES
  - INHERIT PRIVILEDGES OF CALLING ROUTINE

PARALLELISM

Implicit - Transparent to user

INSTRUCTION LEVEL ||ism (ILP)

- PIPELINING
- SUPERSCALAR - MULTIPLE EXECUTION UNITS
- SPEEDUP? 10X

EXPLICIT PARALLELISM

1. DATA LEVEL - CH. 4
   - Many repetitive operations on streams of data
   - Typically in a loop

University of Idaho

SIMD: Data (path) pipelines

Multimedia

2. THREAD LEVEL (CH. 5)

MIMD – Different instructions, Different data

Examples

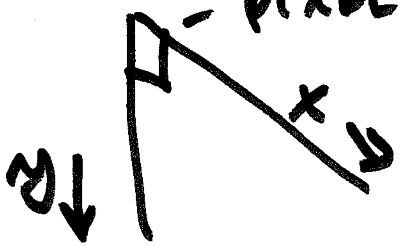
SERVER – eg. DATABASE SERVER

- MANY CLIENTS
- REQUESTS ASYNCHRONOUS
- EACH REQUEST SPAWNS A THREAD
- RUNS IN OWN TIME
- SPEED UP: DISPATCH THREADS TO MULTIPLE PROCESSORS

University of Idaho

DATA PARALLELISM: MEDIA EXAMPLE

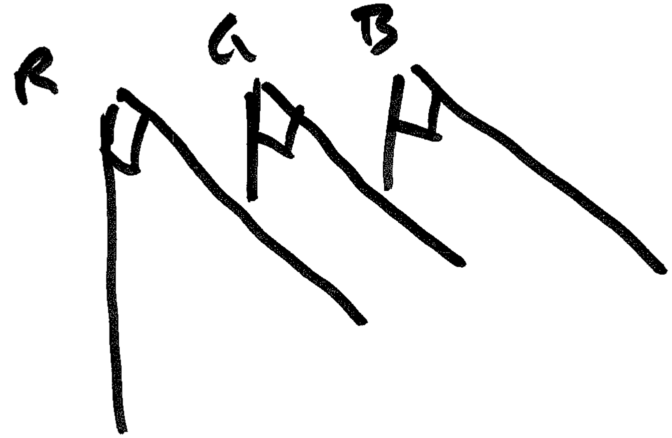
COLOR SPACE CONVERSION



PIXEL ∈ [0, 255]
- 0 — BLACK (NO LIGHT)
- 255 — MAX WHITE

IMAGE

COLOR: 3 IMAGE "PLANES"



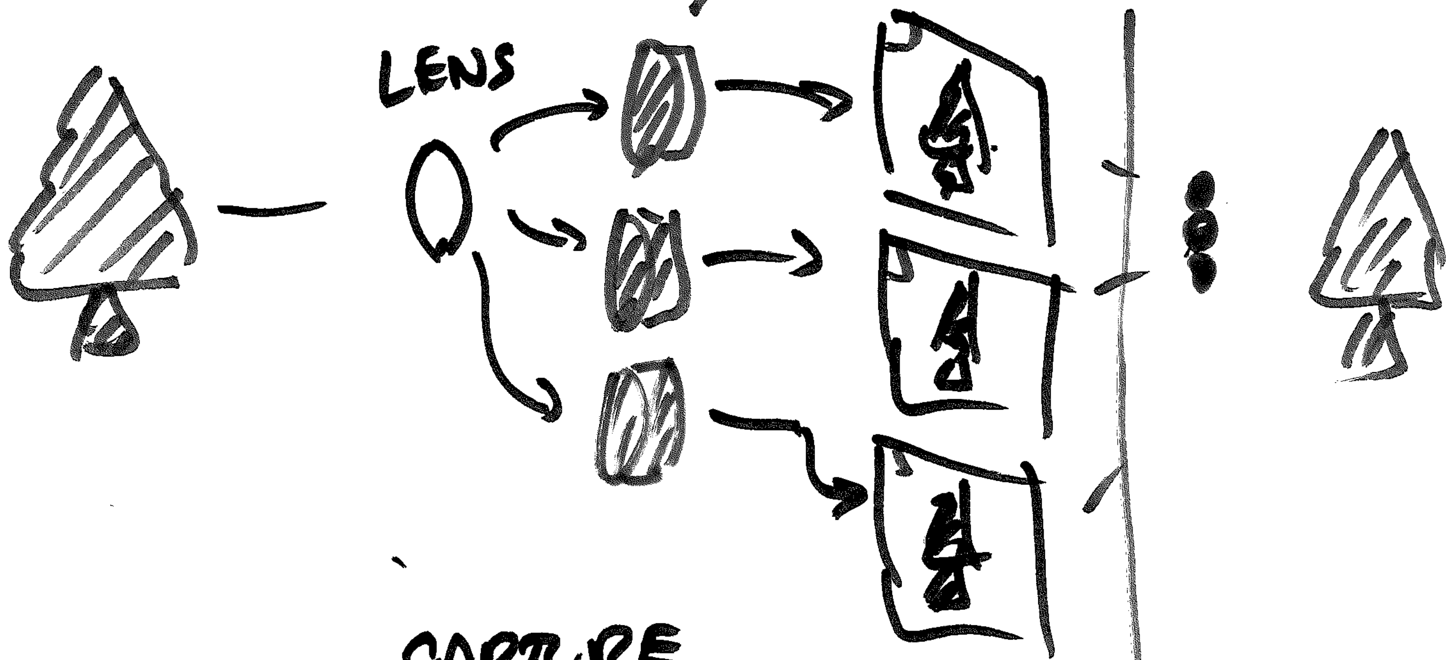
RED ∈ [0, 255]
- 0 — NO RED
- 255 — MAX (SATURATED) RED

GREEN ∈ [0, 255]

BLUE ∈ [0, 255]

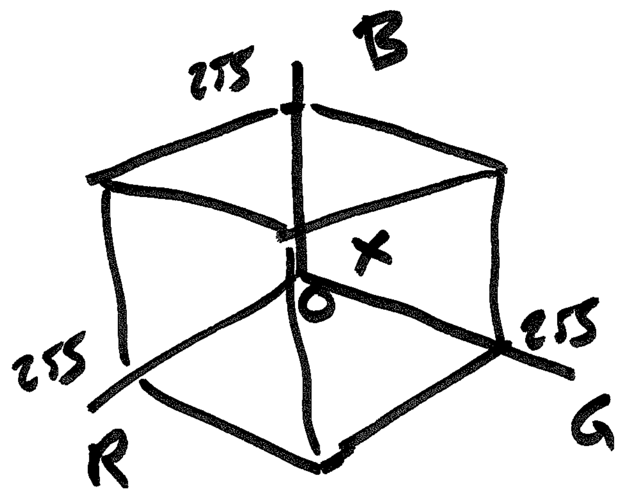
University of Idaho

IMAGE FORMATION

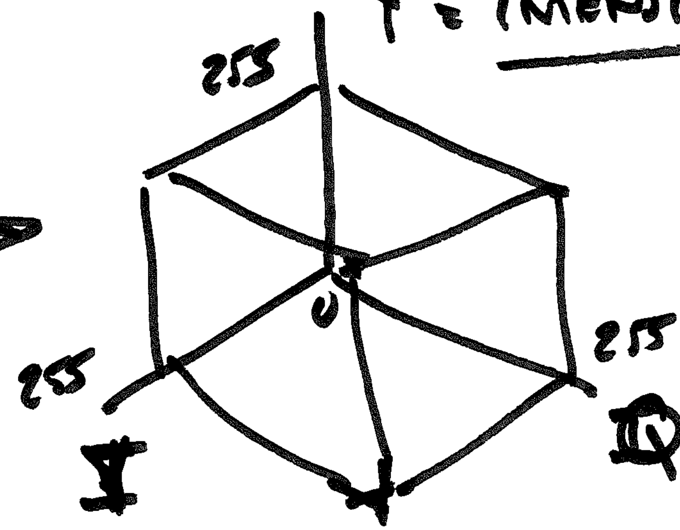


RGB

University of Idaho



EACH COLOR IS A POINT IN RGB SPACE



(I, Q) COLOR OR "HUE" COMPONENT

(R, G, B) → (Y, I, Q)

↳ DISPLAY B & W

University of Idaho

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} .299 & .587 & .114 \\ .596 & -.274 & -.322 \\ .211 & -.523 & .312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

3×3 matrix by 1×3 vector –

on every pixel

4500 × 3500 = 15,750,000 pixels

BUT WAIT! THIS IS VIDEO: 50 FRAMES/SEC.

15,750,000 × 50 = 785×10^6 pixels/sec.

"STREAMING"

IN SISD:

Image: scan columns 0-3499

Image: scan rows 0-4499

EACH PIXEL

Y: SCAN ROW 0 & RGB VECTOR

SUM OF PRODUCTS

I: SCAN ROW 1 & RGB VECTOR

S.O.P.

Q: SCAN ROW 2 : RGB VECTOR

S.O.P.

MANY INSTRUCTIONS!

MUST BE FETCHED, DECODE, ..., WB

University of Idaho

4 LOOP COUNTERS

9 MULTIPLIES, 9 ADDS

AS A VECTOR MACHINE

DEFINE VECTORS

RGB - 3×3

ROW0 - 3×1

ROW1 - 3×1

ROW2 - 3×1

Y ~~Y~~ = $RGB \times ROW0^T$

I ~~G~~ = $RGB \times ROW1^T$

Q ~~B~~ = $RGB \times ROW2^T$

INNER PRODUCT = ONE INSTRUCTION

VECTOR INSTRUCTIONS

- VECTOR LOAD & STORE
- INNER PRODUCT
- VECTOR × SCALAR PRODUCT
- FEWER FETCHES
  - FASTER
  - LOWER POWER CONSUMPTION

EXAMPLES:

INTEL MMX   MULTIMEDIA EXTENSIONS

AVX

ARM: NEON

ADVANCED SIMD

MPE: MEDIA PROCESSING ENGINE

TYPICALLY: VECTOR LENGTH = 64

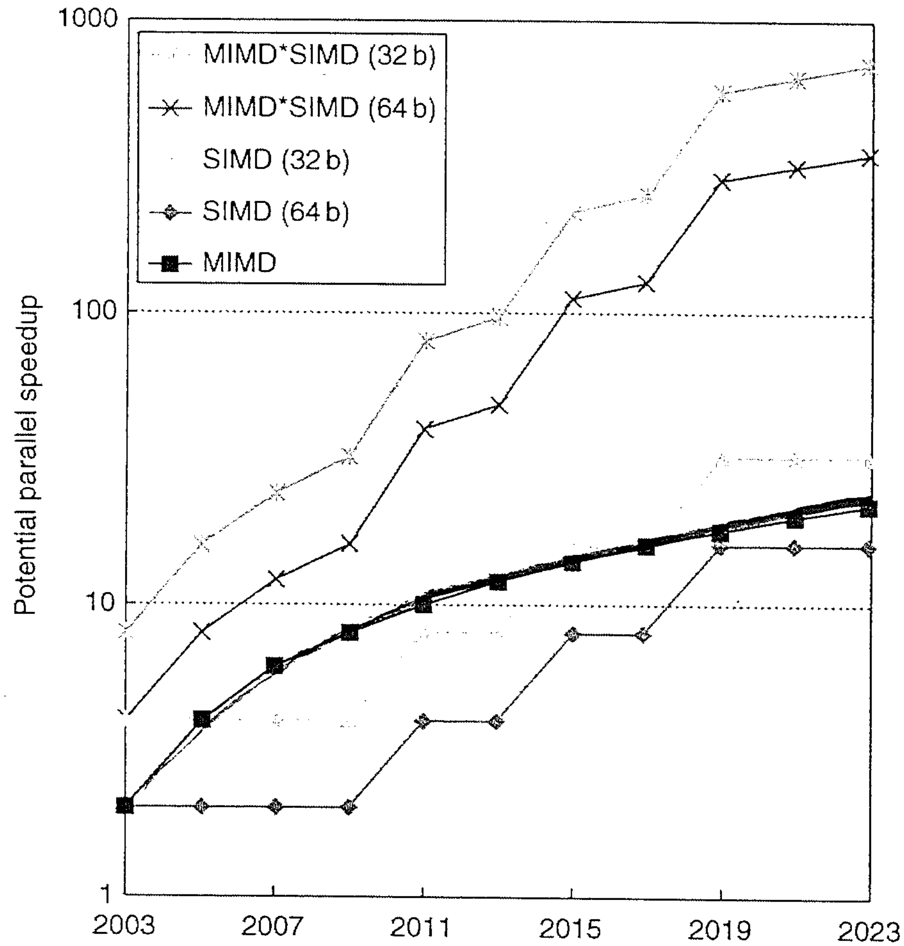


Figure 4.1 Potential speedup via parallelism from MIMD, SIMD, and both MIMD and SIMD over time for x86 computers. This figure assumes that two cores per chip for MIMD will be added every two years and the number of operations for SIMD will double every four years.