

CS 451 / 551 / ECE 541

ADVANCED
COMPUTER ARCHITECTURE

SESSION no. 3

University of Idaho

WEB SITE CORRECTION

www2.cs.uidaho.edu/ncs451_551

underscore! →

DATA TYPES

bits = {0, 1}

0 = 0V

1 = 2.5V

• STABILITY, REPEATABILITY

0 = '0' 1 = '1' 2 = '10'

DATA

\$, pixels, sound level, text

Numbers (2's complement)

Integers - unsigned or signed
addresses, indices quantities

FLOATING POINT

101.100 - FIXED POINT

↑ binary point

$$2^2 2^1 2^0 \cdot 2^{-1} 2^{-2} 2^{-3} 2^{-4}$$

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 + 0$$

$$= 4 + 1 + \frac{1}{2} = 5\frac{1}{2} = 5.5$$

ANALOGY

$$1.23 \times 10^2 = 123$$

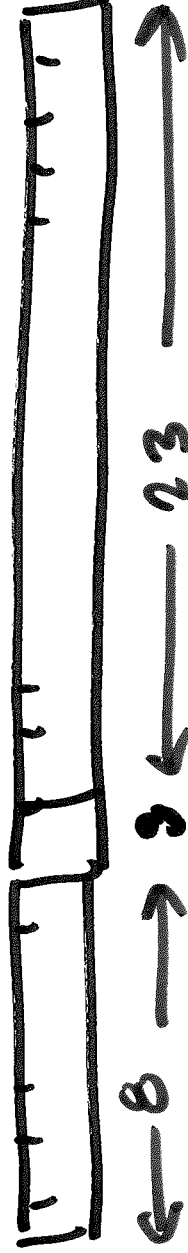
· ADDING · MULTIPLICATION

WHY? RANGE!

$$.0000000123 = 1.23 \times 10^{-9}$$

$$123,000,000,000 = 1.23 \times 10^{11}$$

EXP S SIGNIFICAND (MANTISSA)



$$10^{11} \times 2^{11^2}$$

SCIENTIFIC, TECHNICAL
MODELING, ...

LOOK UP IEEE 754

SINGLE

DOUBLE

- Wikipedia
- Text resources Appendix J

Range for single & double?

How do we add?

How do we multiply?

How do we represent "out of range"?

UNICODE - INTERNATIONAL.

16 bits

All major scripts

8 bit subset = ASCII

PIXEL 8 bits

BLACK

0
0₂

0000000

WHITE

255
2⁸-1

1111111

PERMANENT

RGB

COLOR

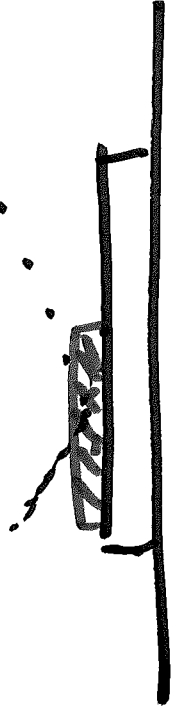
RED
8 bits

GREEN
8 bits

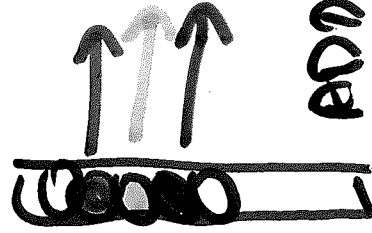
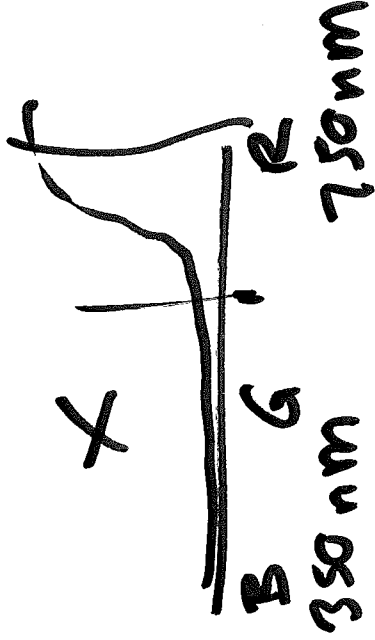
BLUE
8 bits

C = minimum amount (of red)
255 = max. amount (of red)

255 →



SUBSTRACTIVE



ADDITIVE



ADDITIVE

RGB

MEMORY MODEL

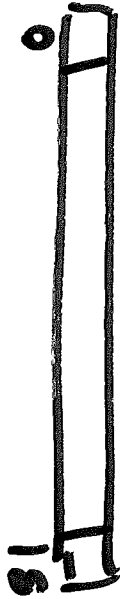
· PRINCIPLE OF LOCALITY OF REFERENCE

MEMORY: ~~THE~~ THE NEXT MEMORY

ACCESS (INSTRUCTION OR
DATA) IS LIKELY TO BE NEAR
THE LAST ACCESS.

TEMPORAL: ~~AND~~ A DATA ITEM THAT
HAS BEEN ACCESSED IS ~~VERY~~
LIKELY TO BE ACCESSED AGAIN
SOON.

$2^{32} - 1$



32 BIT
DATA WORDS

32 BIT
ADDRESSES

$$2^{32} = 4,294,967,296$$

WORDS

$$1K = 2^{10} = 1024$$

$$2^{32} / 2^{10} = 2^{22}$$

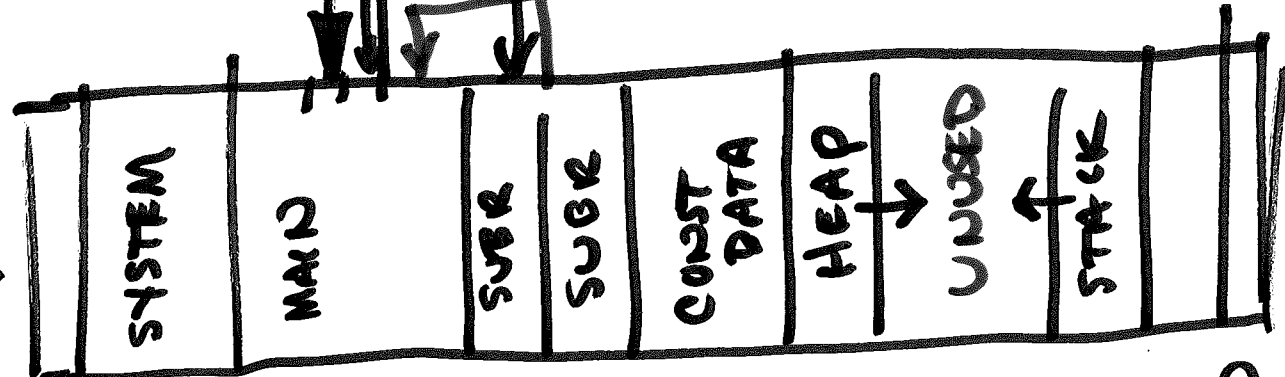
$$1048,576 \text{ KW}$$

$$1M = 2^{20}$$

$$2^{32} / 2^{20} = 4096 \text{ MW}$$

$$1G = 2^{30} = 4 \text{ GWORDS}$$

ADDRESS SPACE



$2^0 = 0$

University of Idaho

HEAP = DYNAMIC MEMORY ALLOCATION

C: malloc (siz) - allocate

free() - free

C++: new (siz)

delete()

Java: new (siz)

"Garbage collection"

LOCALITY

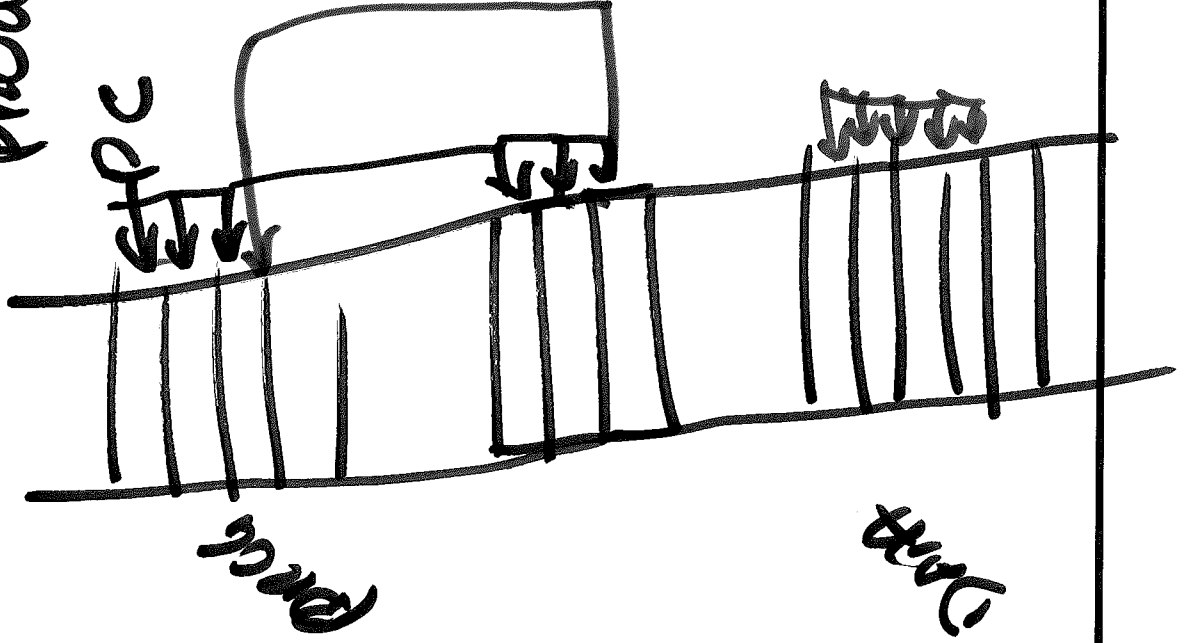
PROG EXECUTION

EPC

LOOPS

LOOP THRU ARRAY

MEMORY REFERENCES
TEND TO CLUSTER
IN ADDRESS SPACE.



HIGHER PERFORMANCE

THROUGHPUT

~~LATENCY~~

LATENCY

POWER CONSUMPTION

SIZE

COST

1.8 PERFORMANCE GOALS (METRIC)

EXECUTION TIME time()

wall clock (real) time -

processor time { user
system -

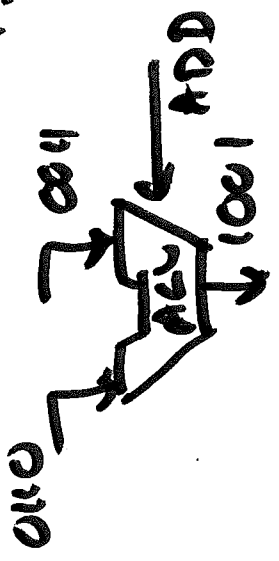
CPU time ~~from~~ processor

CYCLE TIME PER INSTRUCTION

• Not all instructions same number of clock cycles

EX ADD

0110	(6)
0011	(3)
<hr/>	
1001	(9)



MULTIPLY

0110	(6)
× 0011	(3)
<hr/>	
0110	
0000	
0000	
0000	
0010010	
<hr/>	
2 ⁴ + 2 ¹	= 1(18)

METRICS - PERFORMANCE

MIPS Mega Instr / Sec.

PROCESSOR A 100 MIPS -

.. B 150 MIPS -

BENCHMARKS - STANDARD TEST PROGRAMS

SUITES

WHETSTONE, DAREY STONE, GAUSSIANS

SPEC - STANDARD PERFORMANCE EVALUATION CORPORATE