2.

(d.) IEEE 754 FP. infinity (single precision) 32 −9 23

fraction exponent = 1111 1111

$\boxed{0\ 1111\ 1111\ 00000000000000000000000}$

1 8

(e) 5bits, register의 개수. 32개

(f)

2 | 100
2 | 50 … 0
2 | 25 … 0
2 | 12 … 1
2 | 6 … 0
2 | 3 … 0
1 … 1

0.1100100

0110 0100.

1's complement 1001 1011.

2's complement 1001 1100.

(g). align (exponent) → add significands → normalize, check overflow/underflow → round off.

(h). $\boxed{127,\ 1023}$

(i).

single digit to the left of the decimal point

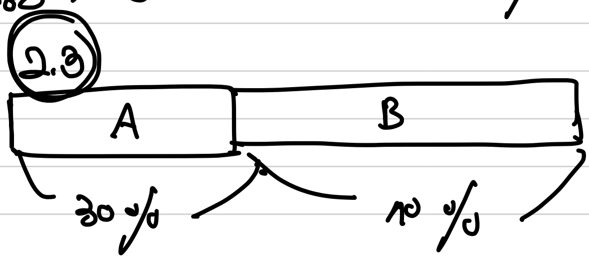
(j)

(Fortran Ada.) ⟷ C, Java

#3.

Quad-core processor. 2.3Ghz.

성능 → 시간에 반비례 clock cycle time



시간

performance에 반비례

$\frac{25}{19}$

↓

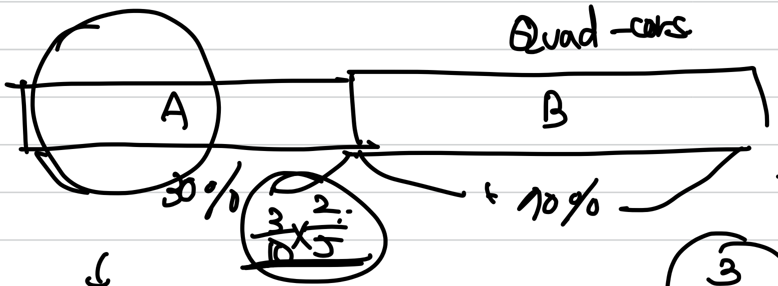
향상시키려면.

19) 25 (1.3 1...
19
60
57
30
19
110

1.32배

(b)

Quad-core

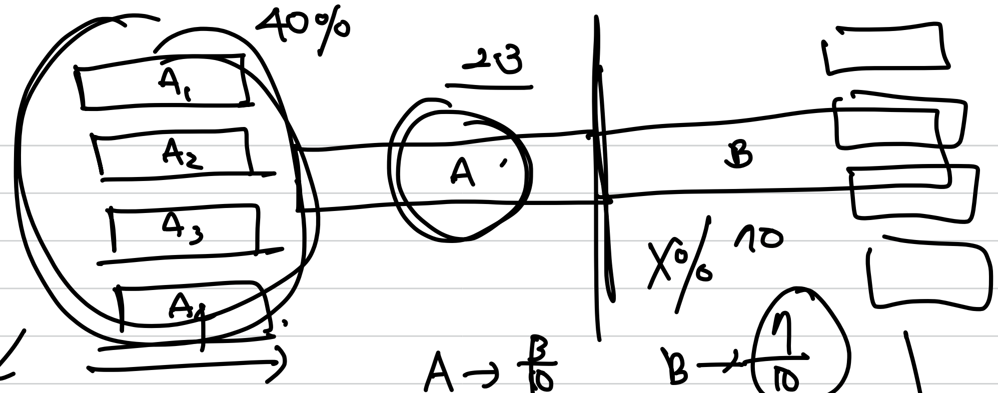


$\frac{3}{25}$ 부분.

A코어의 40%는. 병렬처리 가능.

4개 → 병렬하면. 가속X

아프리카 X →



X %

$\frac{X}{100}$

$A \to \frac{3}{10}$

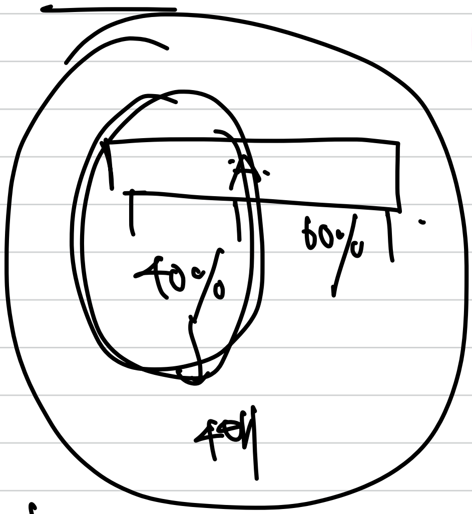
$B \to \frac{7}{10}$

A의 $\frac{2}{5}$ A 전체 50

병렬처리 $\frac{3}{10} \times \frac{2}{5} = \frac{3}{25}$

병렬처리 X $\frac{3}{10} \times \frac{3}{5} = \frac{9}{50}$

암호의 병렬



0.28

오답중 병렬로 재야

$\frac{3}{5}T + \frac{\cancel{2}^{1}}{5}T \times \frac{1}{\cancel{4}_{2}}$

$= \frac{3}{5}T + \frac{1}{10}T$

$= \frac{7}{10}T$

$2.3 \times \frac{3}{10} \times \frac{7}{10}$

$+ 2.3 \times \frac{7}{10} \times \square$

$\frac{2}{7}$배

$7 \overline{)2.0}$ 0.28

$\frac{50}{25}$

$\cancel{2.3} \times \left(\frac{21}{100} + \frac{7}{10}X \right) = \frac{7}{25}$

$\overline{\cancel{2.3}}$

$X = \frac{55}{70} = \frac{11}{14}$

$\frac{1}{4}$ x $(1-x)$

$\frac{1}{4}X + 1 - X =$

$\frac{21 + 70X}{100} = \frac{16}{100}$

B

$\cancel{T} \cdot x \times \frac{1}{4} + \cancel{T}(1-x) = \frac{11}{14}\cancel{T}$

$70X = 55$

28.57%

$$\frac{x+4-4x}{4} = \frac{11}{14}$$

$$44 = 14(-3x+4)$$

$$-3x+4 = \frac{22}{7}$$

-3

$$3x = \frac{6}{7}$$

$$\frac{6}{21} \quad \frac{2}{7}$$

#4.

Convert $\underline{-5.3125}$

$\underline{-1.010101 \times 2^{2}}$

$\underline{2+127}$

$= \underline{129}$

(b)

$1000\ 0001$

$1\ 1000\ 0001\cdot 0101010000000000000000$

$\frac{2}{7}$ 100

10A 23

12

$\boxed{C0AA0000}$

$$7\overline{)200}\ \ 28.57$$

14

60

56

40

35

50

49

1

(c)

→ 실제 정렬하려는 상황을 생각해보면.

음수와 양수와 크기가 관련될 수 있기 때문..?

r52 r51

5.

0x 00148493 → $\boxed{\text{addi x9, x9, 1}}$

imm | rs1 | | | x9
0000 0000 0001 | 0100 | 1 000 | 0100 | 1001 | 0011 → addi.
x9 | | x1 | | 9

0x 40578833

rs2 | rs1 | | x16
0100 000 | 0 0101 | 0111 | 1 000 | 1000 | 0011 | 0011
x5 | x15 |

$\boxed{\text{sub x16, x15, x5}}$

0x 0F053483

ld → lmmed.

rs2 | rs1 | | rd
0000 1111 | 0000 | 0101 | 0 011 | 0100 | 1000 | 0011
x16 | x10 | func3 | x9 | store

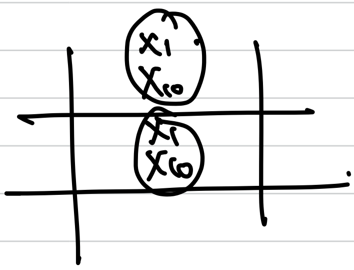
16 32 64 128

192+48
=240

$\boxed{\text{ld. x9 240(x10)}}$

6.

(a). $\boxed{\text{bge. x5, x0, L1}}$



(b)

fact(2) → jalr 몇번 3번

(c) There are two ld instructions at:

→ "factorial 위에서, 밑 마지막이니까.

x1 →
x10 →) 사용한 ~~경우~~ X

(d). line 12

#8. for (n=0; n<m; n++){

if (myarray[n]<0){

} myarray[n]=0;

x_{11} ← base of myarray

x_6 ← n.

x_7 ← m.

double word.

```
main:
    addi x6, x0, 0        // n=0 초기화
Loop:
    blt x6, x7, Exit      // n<m 이면. Exit
    slli x28, x6, 3       // x8 = n×8
    add x29, x11, x28     // x29 ← address of myarray[n]
    ld x30, 0(x29)        // x30 = myarray[n]
    bge x30, x0, L1       // myarray[n] < 0 → L1
    sd x0, 0(x29)         // myarray[n] = 0
    addi x6, x6, 1        // n++ → 해당 안하면 계속 loop
    jal x0, Loop          // goto Loop
L1:
    addi x6, x6, 1.       // n++
    jal x0, Loop.         // goto Loop
Exit:
    jalr x0, 0(x1)
```

beq x11, x0, L1 → "very long jump"

UCL

lui x9, L1_upper20
addi x9, x9, L1_lower20
bne x11, x0, L2
jalr, x0, 0(x9)
L2:

synchronize 등

책 저자

David A. patterson
John L HENNESSY

100

0110 0100
1001 1011