

HW
(COST)
 ↙ ↘
 \$COST REAL ESTATE (SPACE)
 ENERGY

$$\Delta t = 2 \times \frac{DT}{\text{PROCESSING IN} \rightarrow \text{OUT}}$$

WORST
CASE

$$DT_{AND} + DT_{OR} + 2 \times \cancel{DT_{NOT}} +$$

→ 123 456 | 912
 → 123 , 456 | 912

$$D = \underbrace{(\overline{A} \cdot \overline{B} \cdot C)}_{\text{PRODUCTS}} \underbrace{+}_{\text{SUM OF}} \underbrace{(A \cdot \overline{B} \cdot \overline{C})}_{\text{PRODUCTS}} + \dots$$

$$E =$$

$$\overline{A} \cdot B \cdot C$$

+

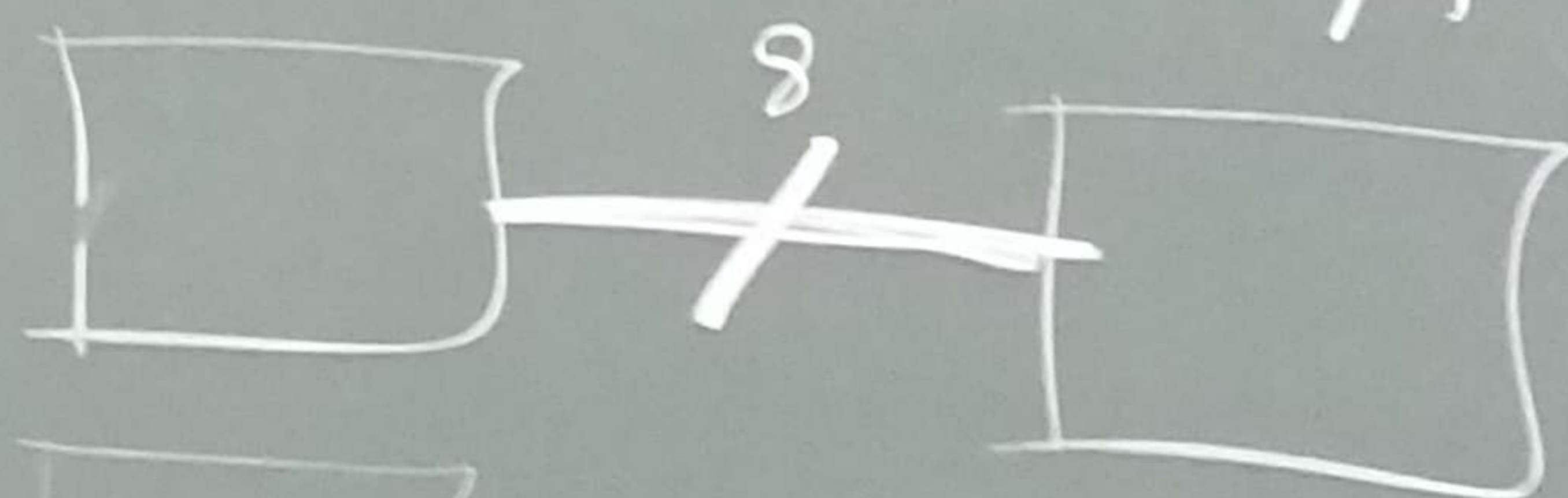
$$A \cdot \overline{B} \cdot C$$

+

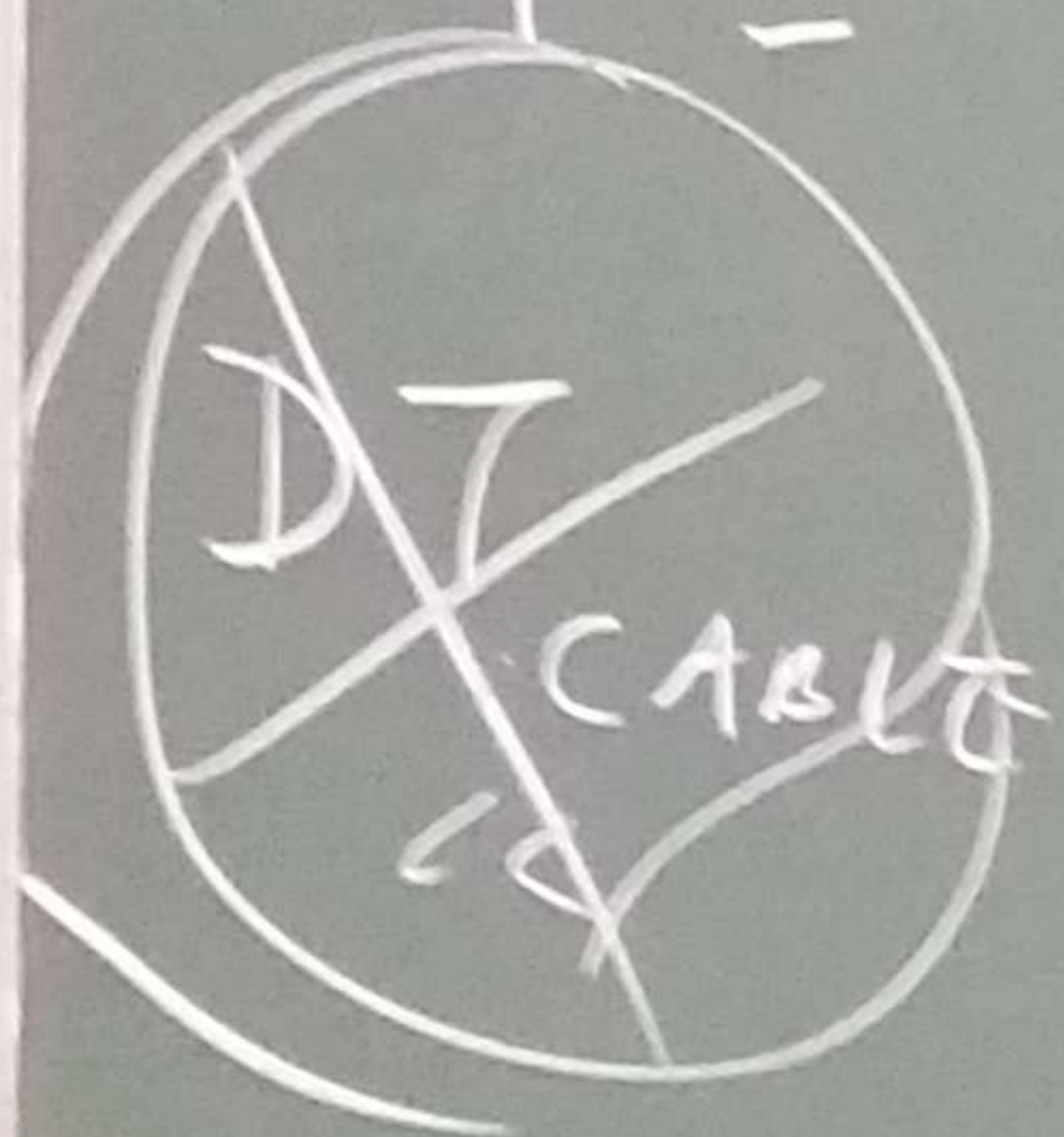
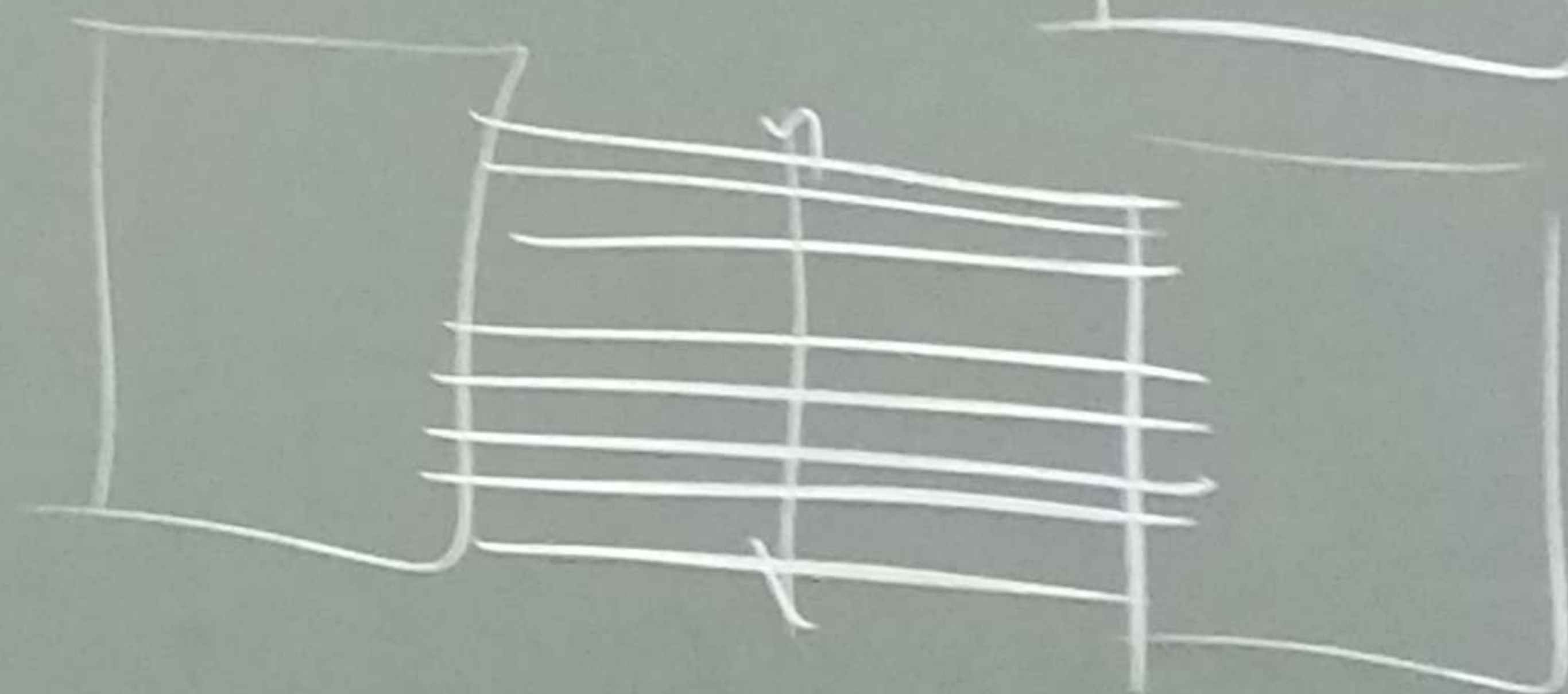
$$A \cdot B \cdot \overline{C}$$

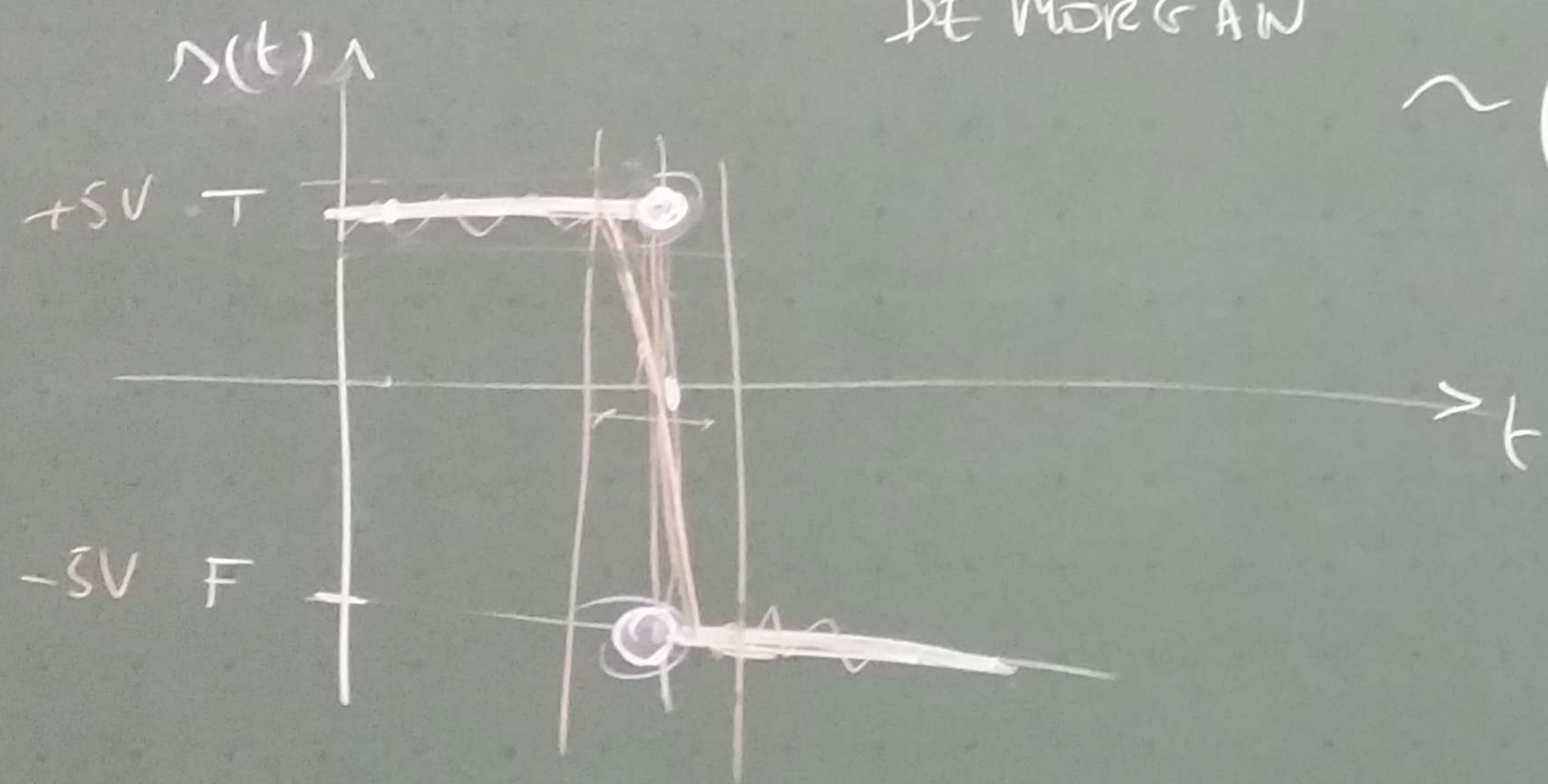
$$F =$$

$$A \cdot B \cdot C$$



"BUS"





MATLAB

DE MORGAN

$$\sim (A + B) = (\sim A) \cdot (\sim B)$$

$$\sim (A \cdot B) = (\sim A) + (\sim B)$$

$$w = \text{sqrt}(z) \quad \text{where} \\ w = z$$

$$f_t: \mathbb{N} \rightarrow \mathbb{N}$$

$$f_p: \mathbb{N} \rightarrow \mathbb{N} \cup \{ \text{DON'T KNOW} \}$$

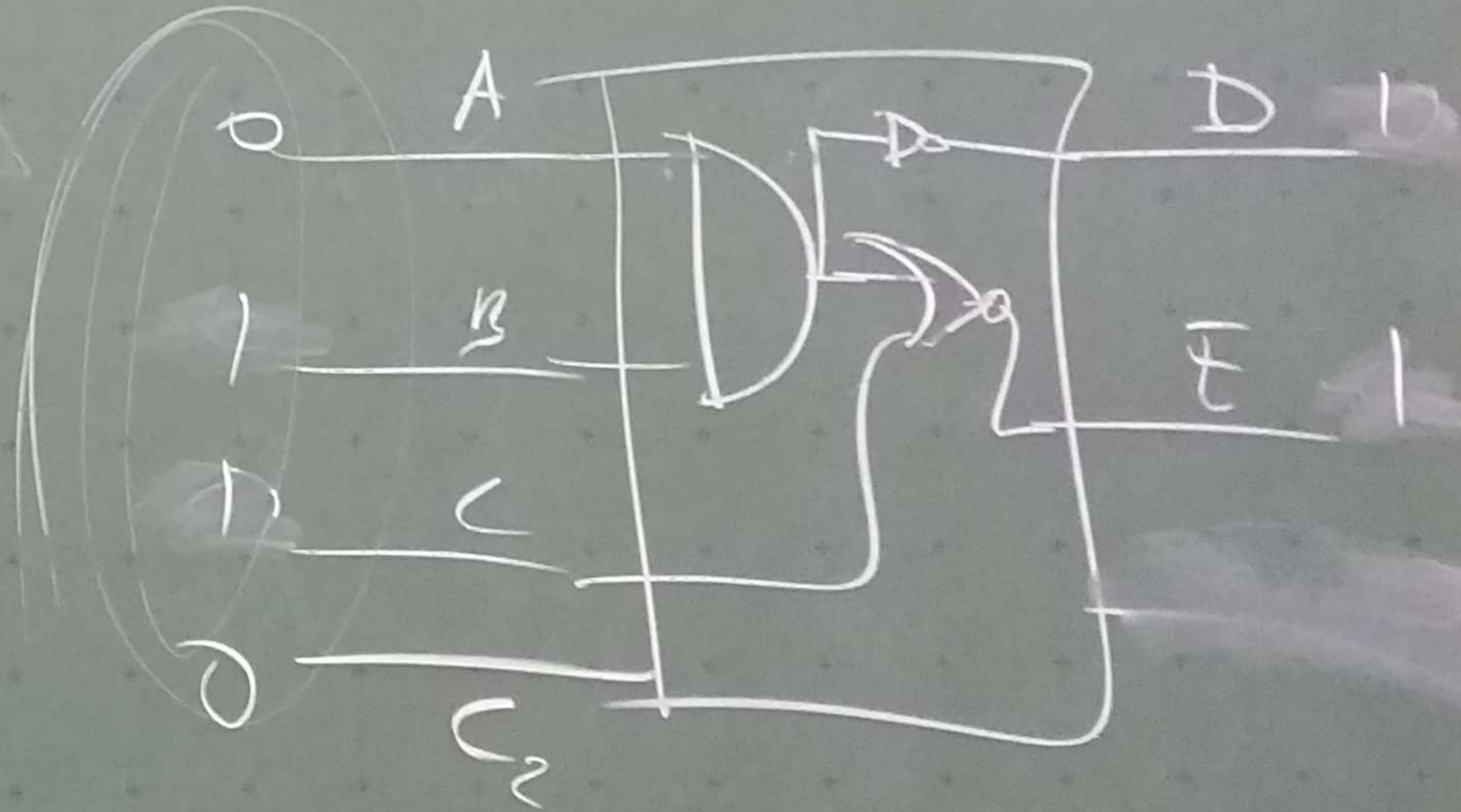
$$(a \cdot b) + (c \cdot d)$$

$\sim$
 $\dot{!} =$

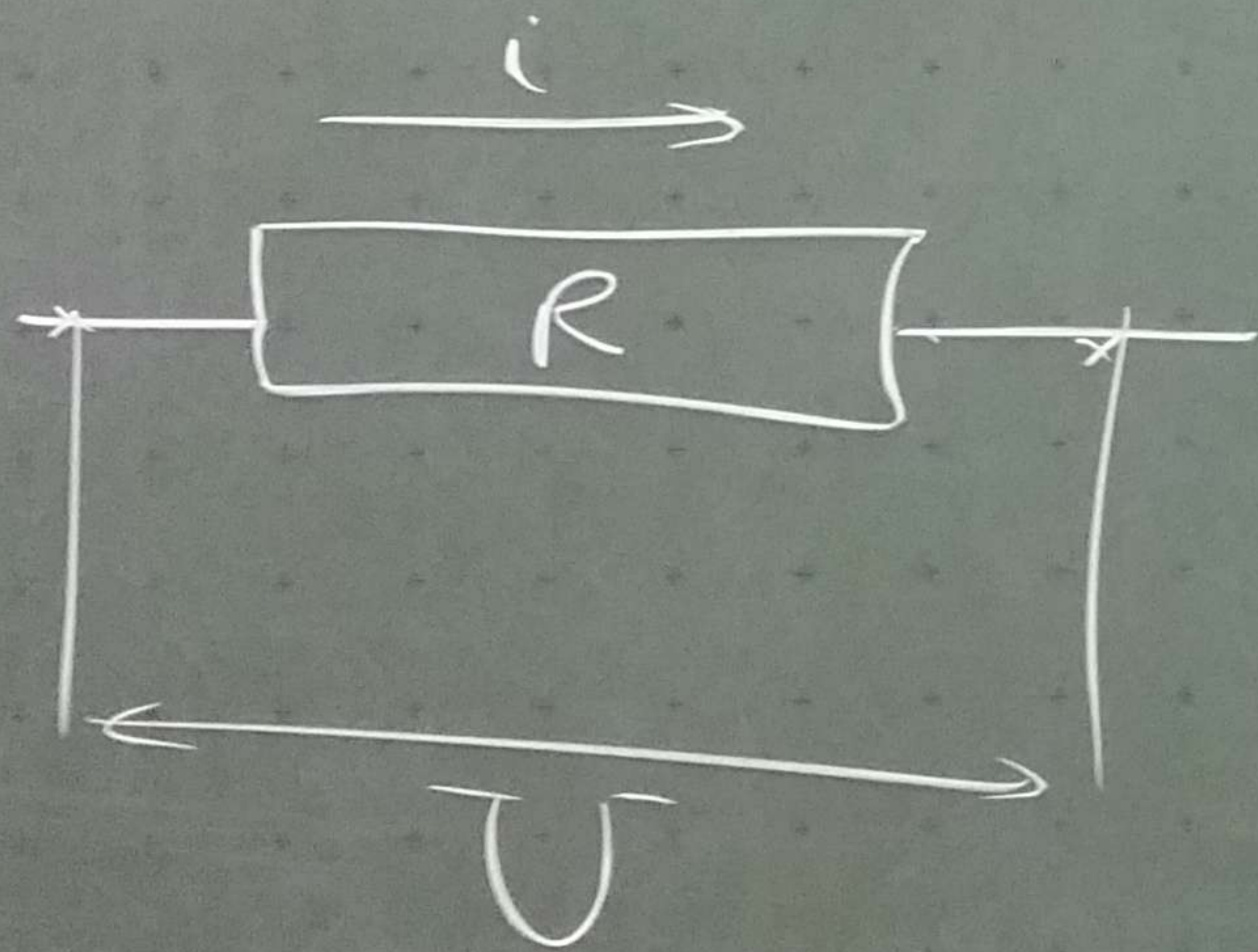
$\neg \vee \wedge \checkmark$

1	2
2	4
3	8
4	16
5	32

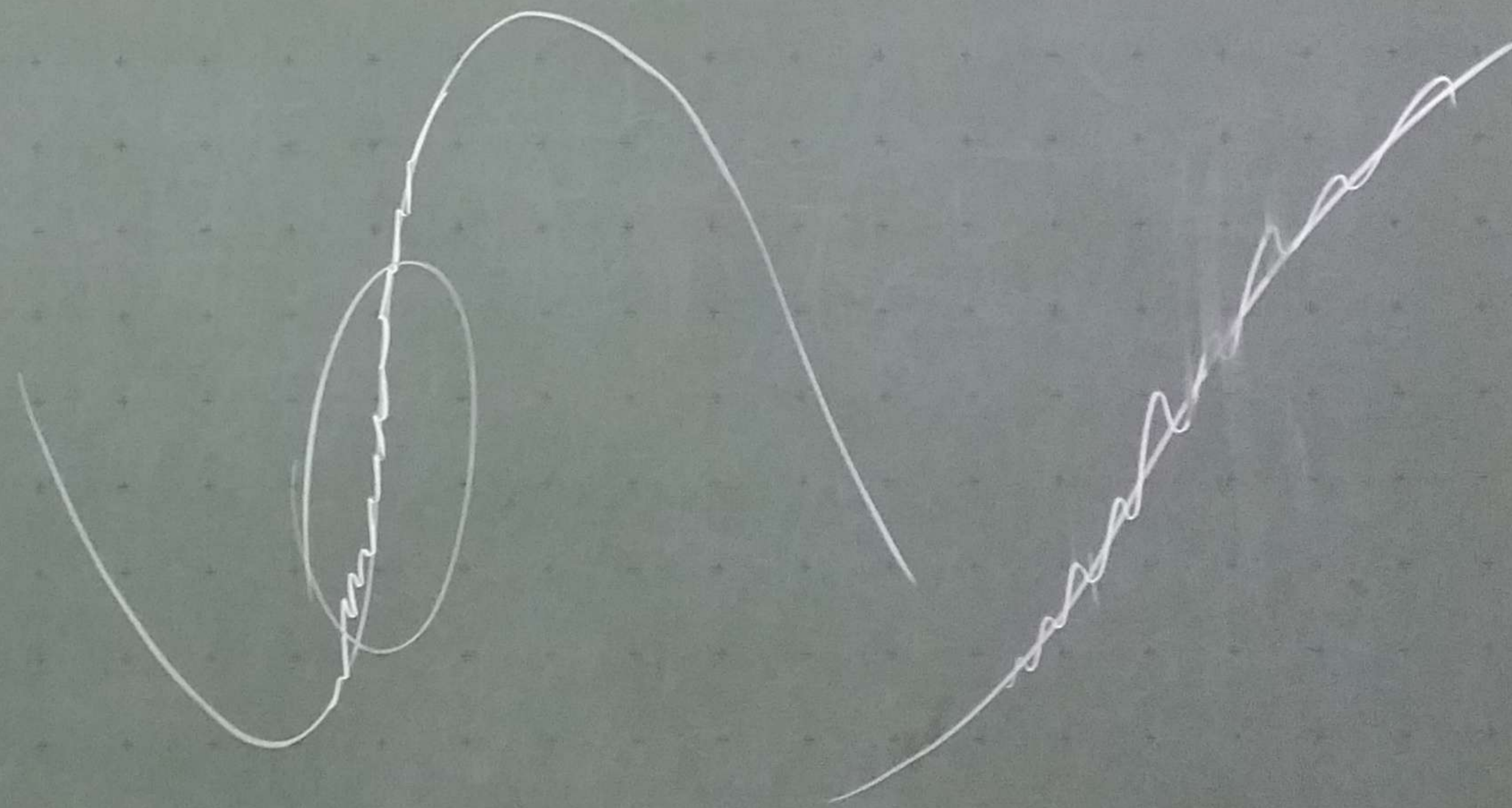
N 2^N



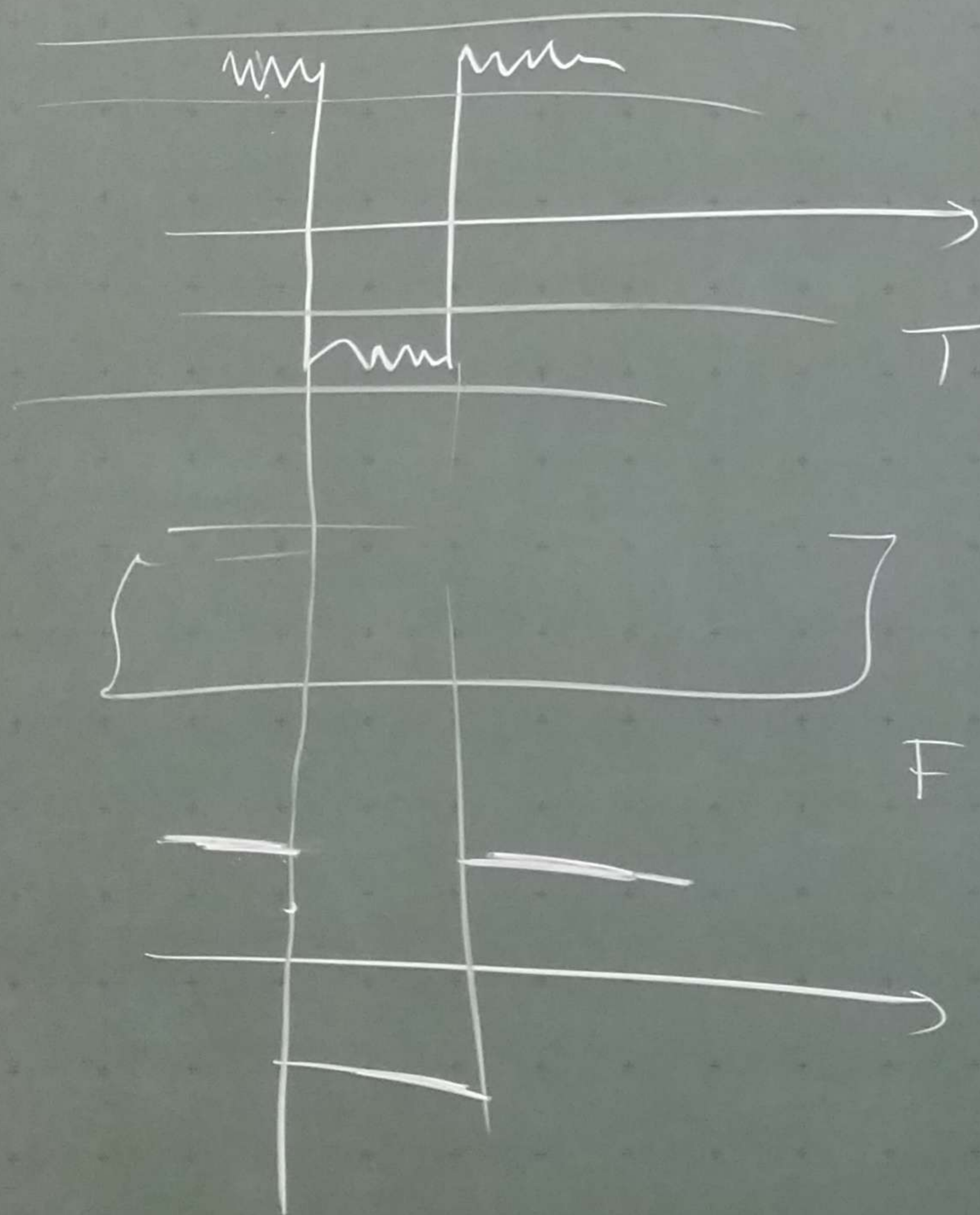
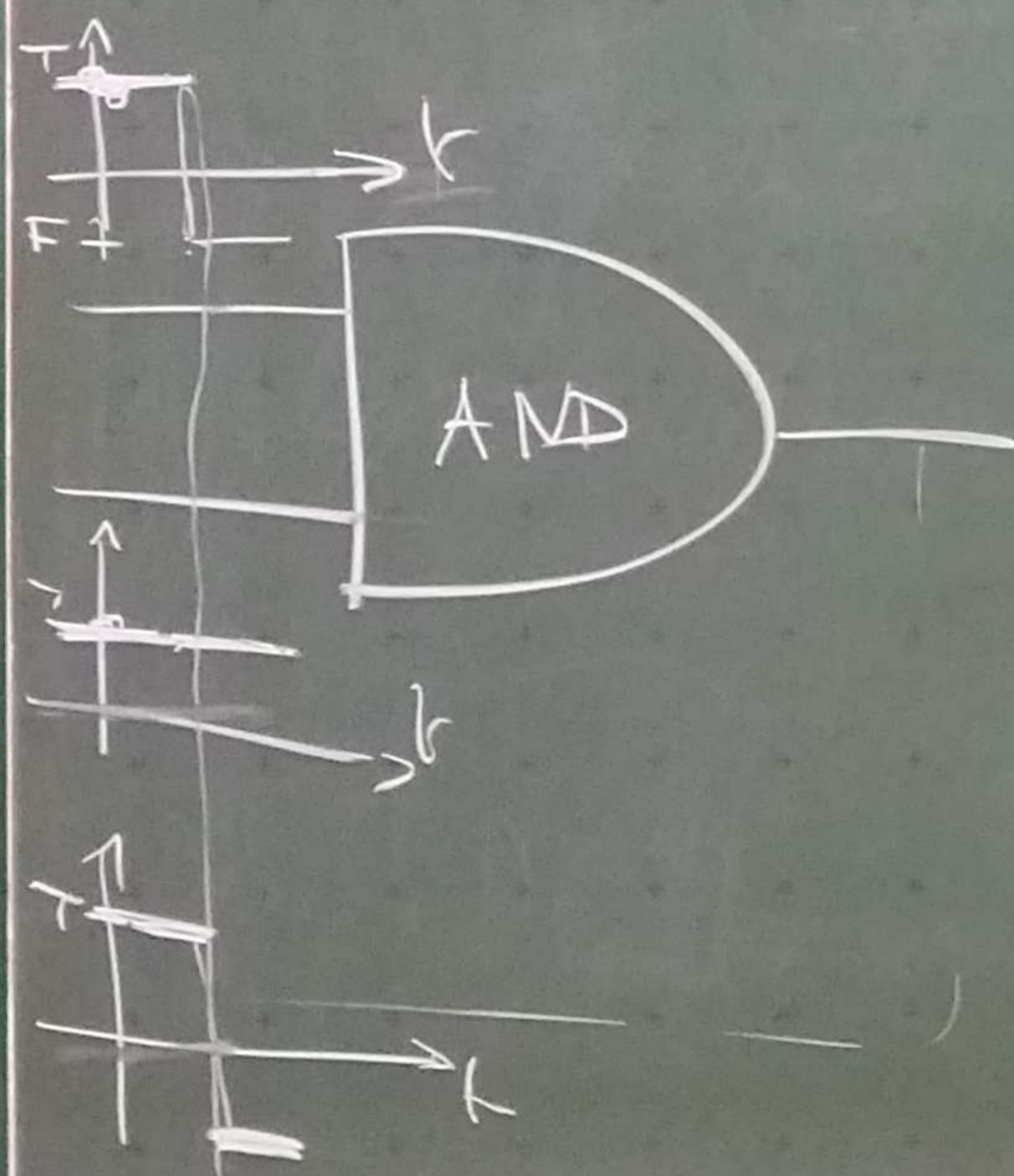
$$U = \overset{\text{!}}{\underset{\text{!}}{R}}(T) \cdot i$$



REPRODUCEERBAAR

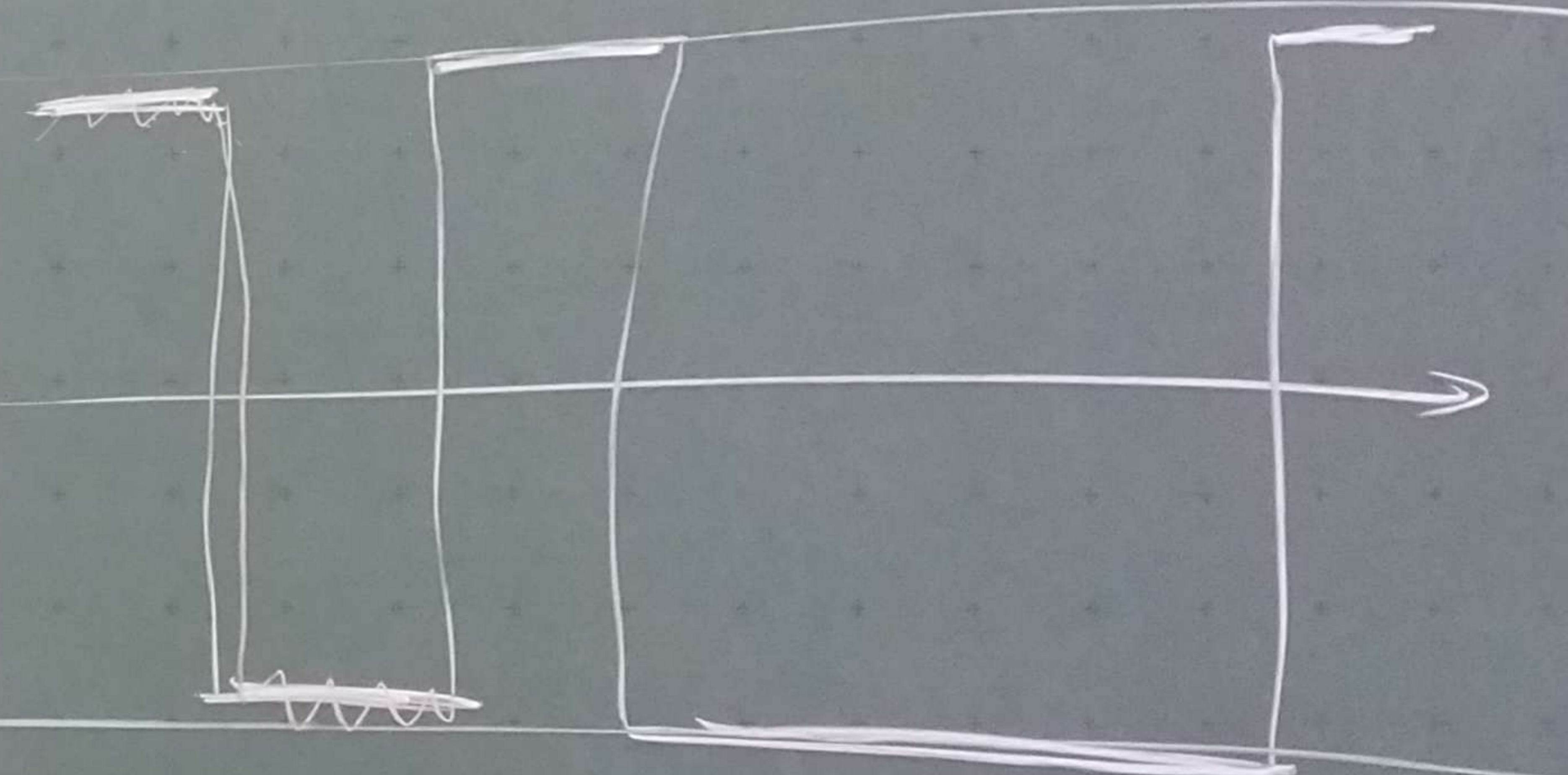


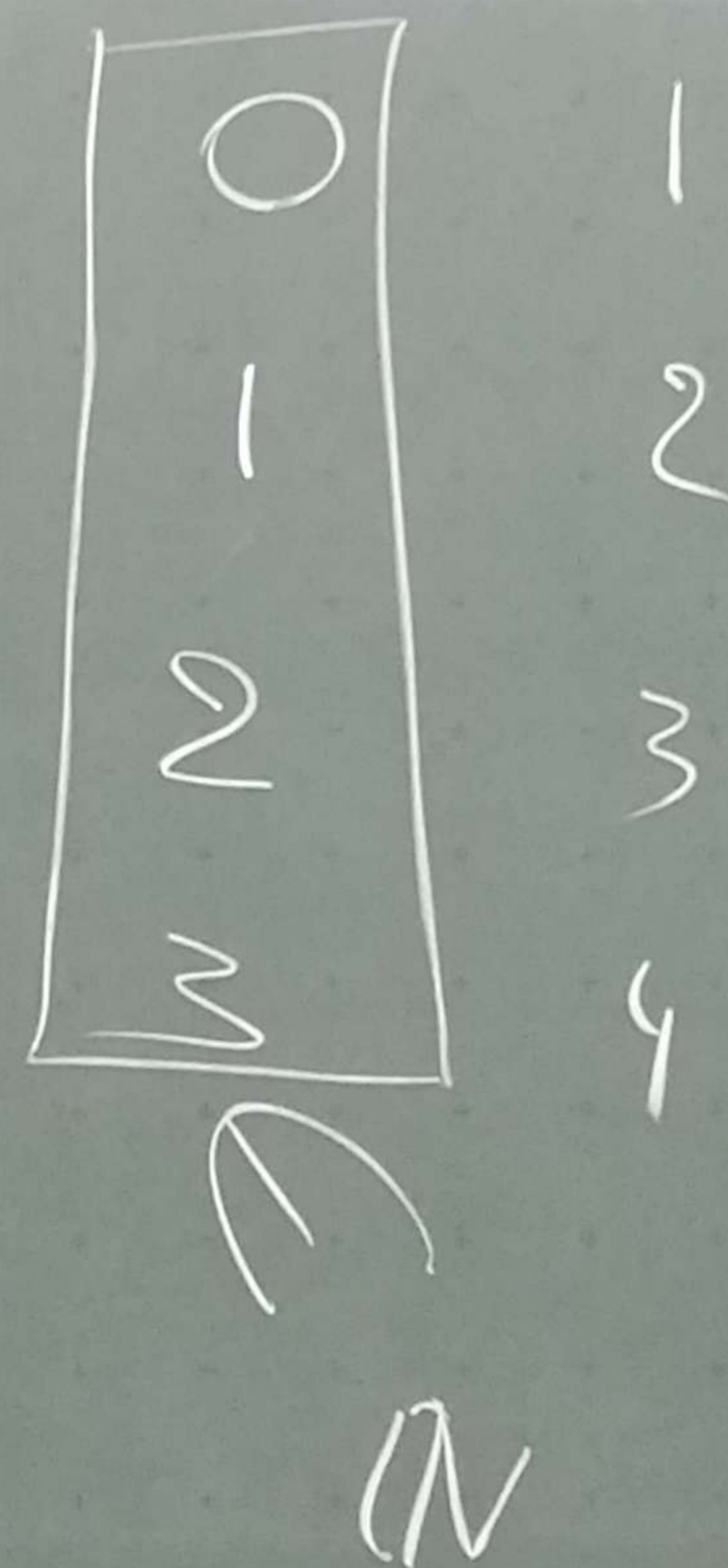
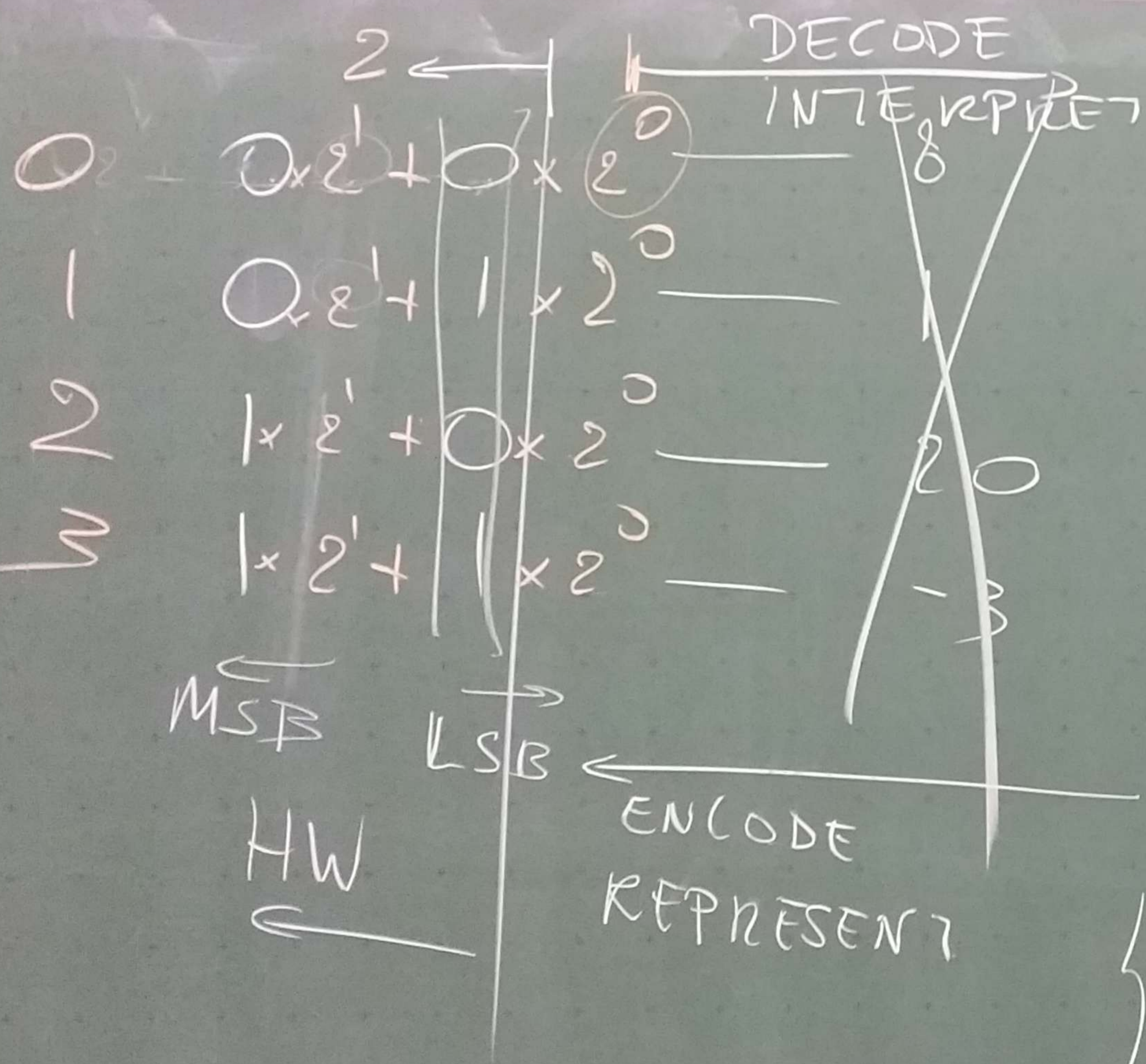
x AND y



TW ①

F ② ③





$$\{0, 1, 2, 3\} \subseteq \mathbb{N}$$

01001₂

$N \text{ bits} \rightarrow 2^N$

$$X = \underbrace{X_{n-1} 2^{n-1} + X_{n-2} 2^{n-2} + \dots + X_0 2^0}_{n \text{ bits}}$$

$\rightarrow 0, 1, \dots, 2^N - 1$

k

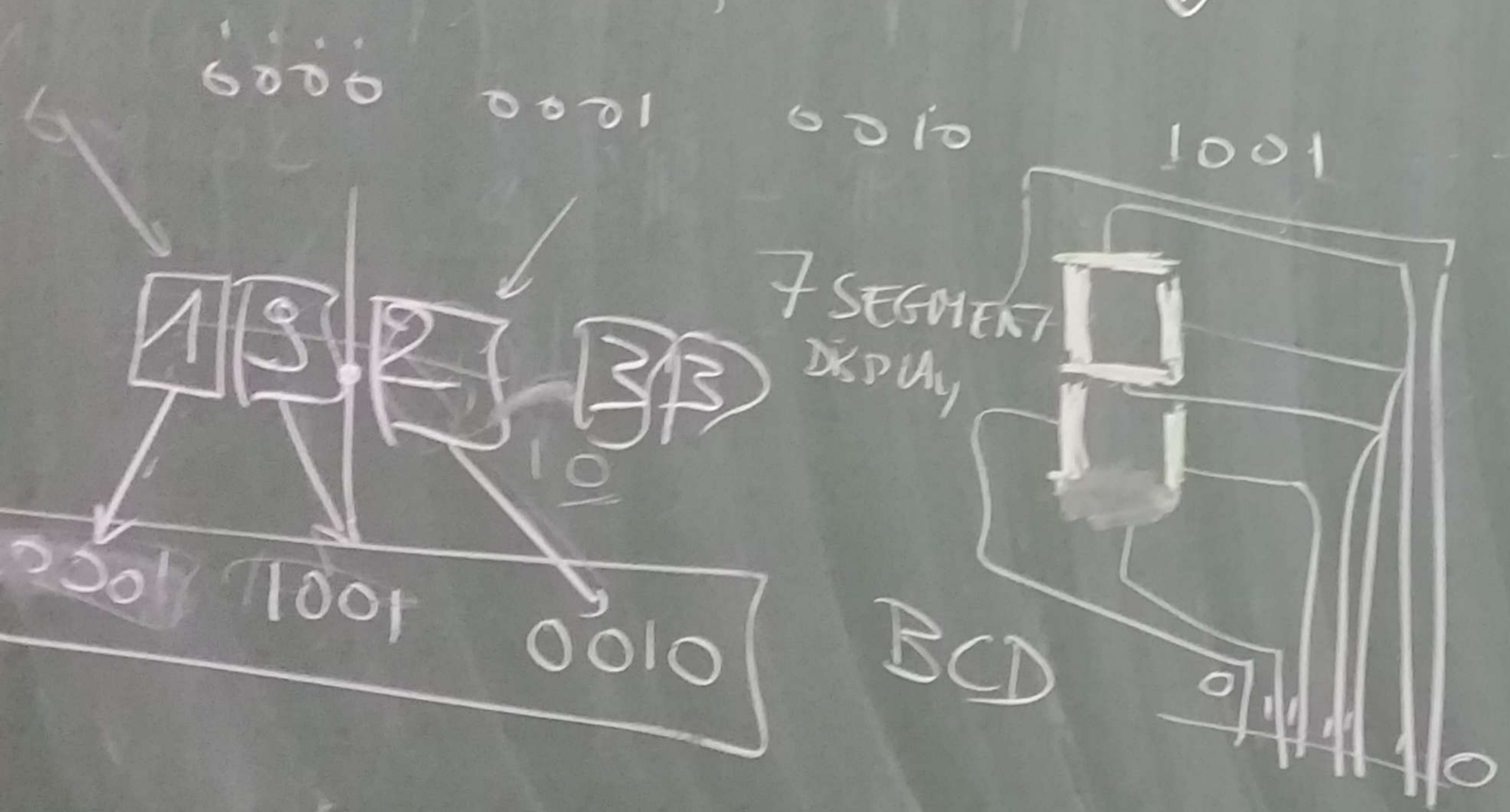
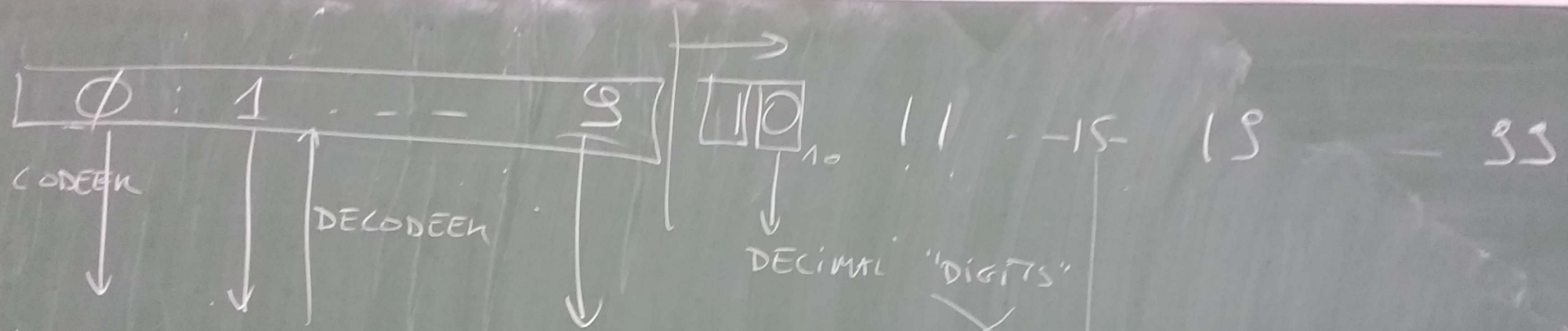
01001₂

$$1001 = 0 \times 10^4 + 1 \times 10^3 + 0 \times 10^2 + 0 \times 10^1 + 1 \times 10^0$$

— x — x — x —

$$3.75_{10} \rightarrow 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$$

1001.11₂



1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

$$\begin{array}{r} 12 \uparrow \\ 12 \uparrow \times 2 \downarrow \end{array} \begin{array}{c} 011 \\ 0110 \end{array} = \begin{array}{c} 3_{10} \\ 6_{10} \end{array}$$

$$\begin{array}{r} 12_{10} \\ 1200_{10} \end{array}$$

$$2^{32} = 2$$

$$2 + 10 + 10 + 10$$

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

4 1024 1024 1024

$$\begin{array}{c} 1 \\ 2 \\ 1 \\ 2 \end{array}$$

$$\begin{array}{c} 1111111111111111 \\ \text{FFFFF} \end{array}$$

DECONT

DECONT

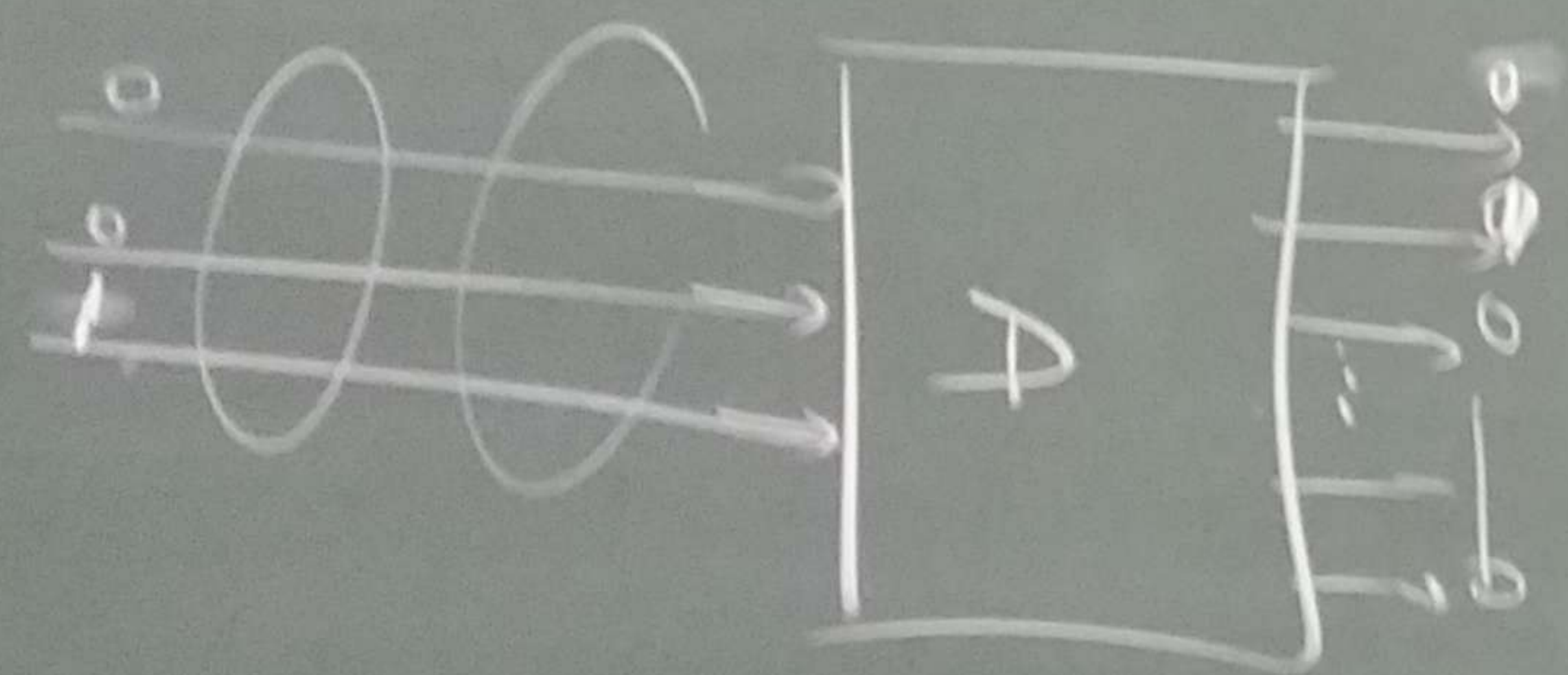
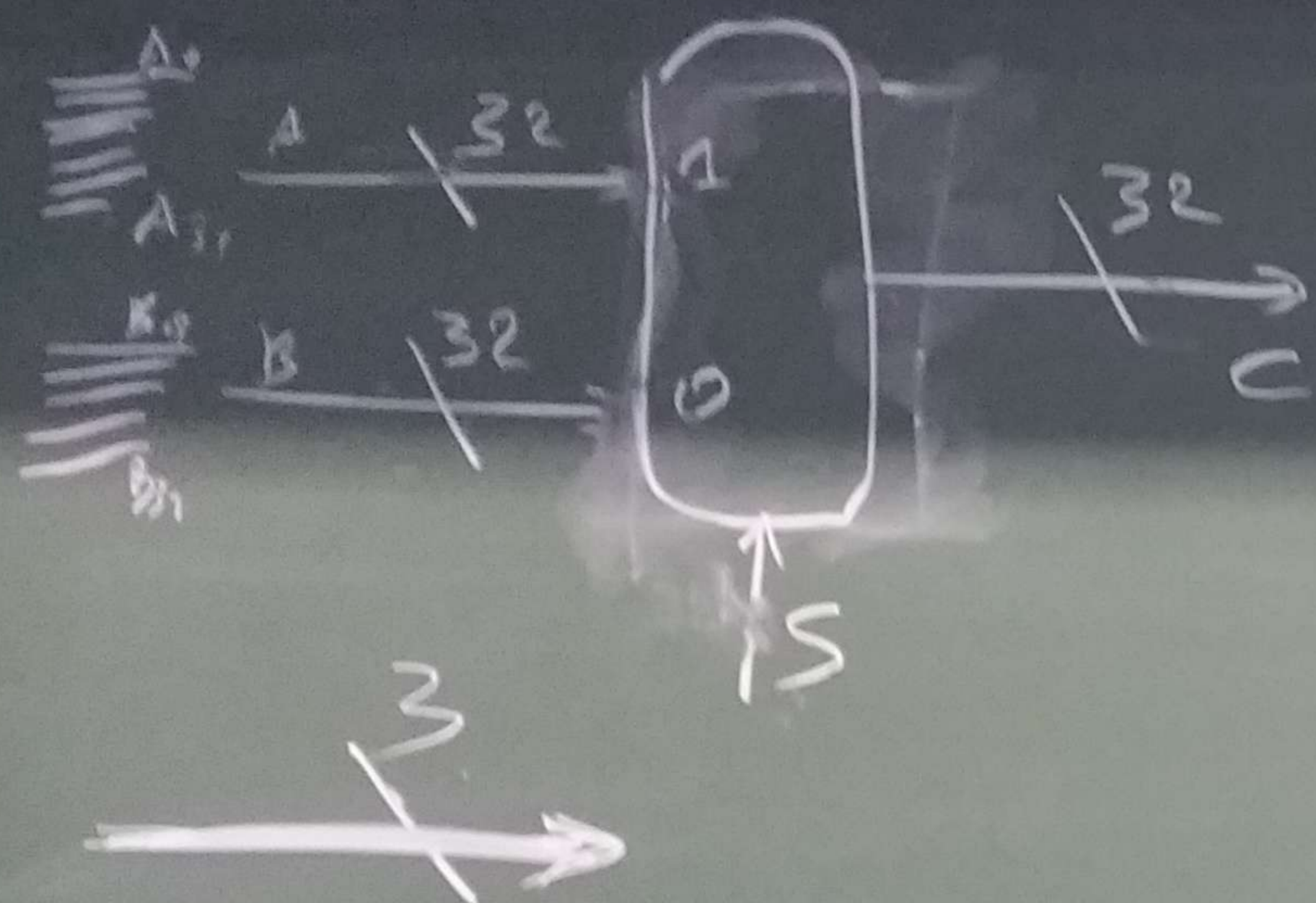
3 1 F

$3 \times 16^2 + 1 \times 16^1 + 15 \times 16^0$

$$123.9 \overline{9}_{10}$$

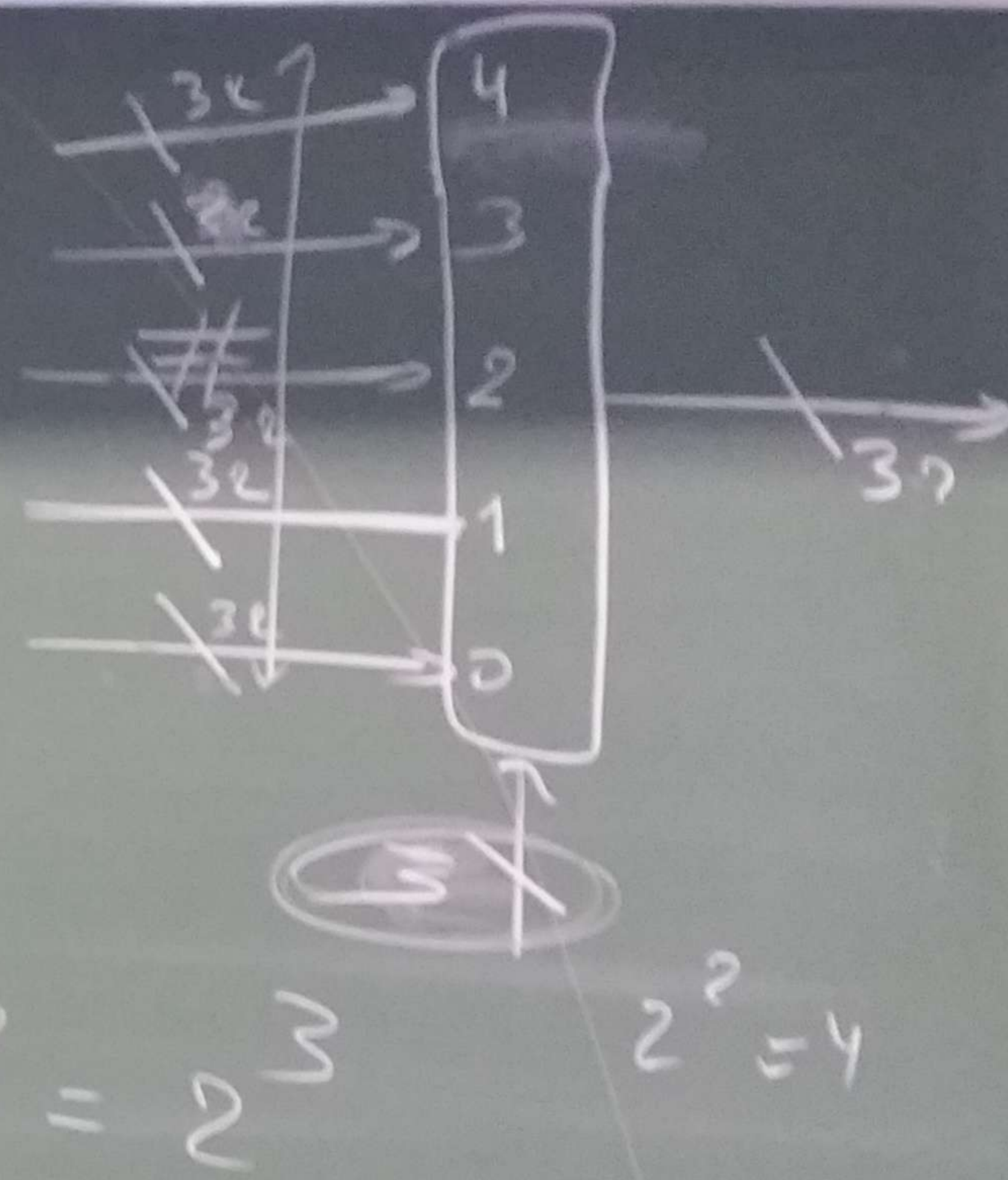
$$\begin{array}{r} 0.1_2 \\ 0.01_2 \\ 0.001_2 \end{array} \quad \begin{array}{r} 0.5_{10} \\ 0.25_{10} \\ 0.125_{10} \end{array}$$

0.2



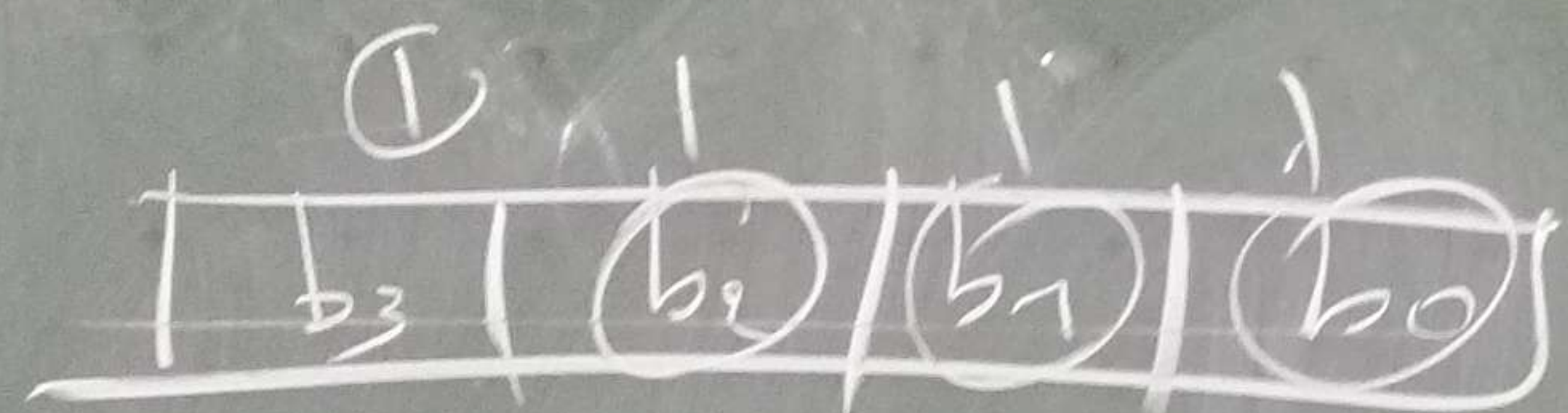
DECODE

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7



$$8 = 2^3$$

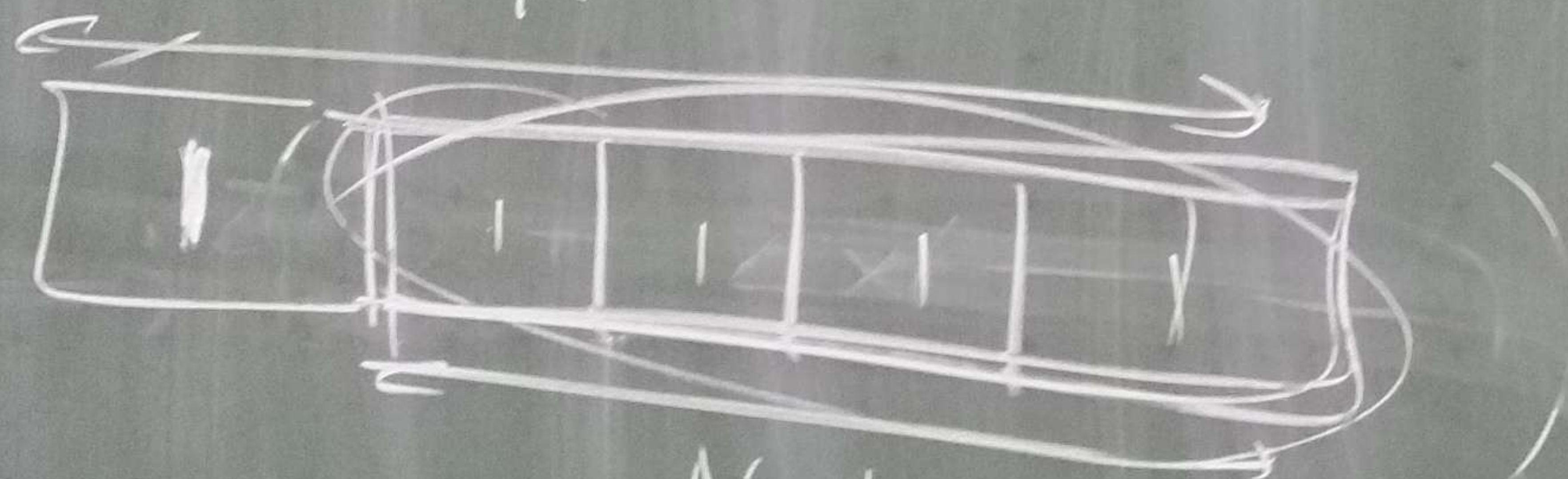
$$2^2 = 4$$



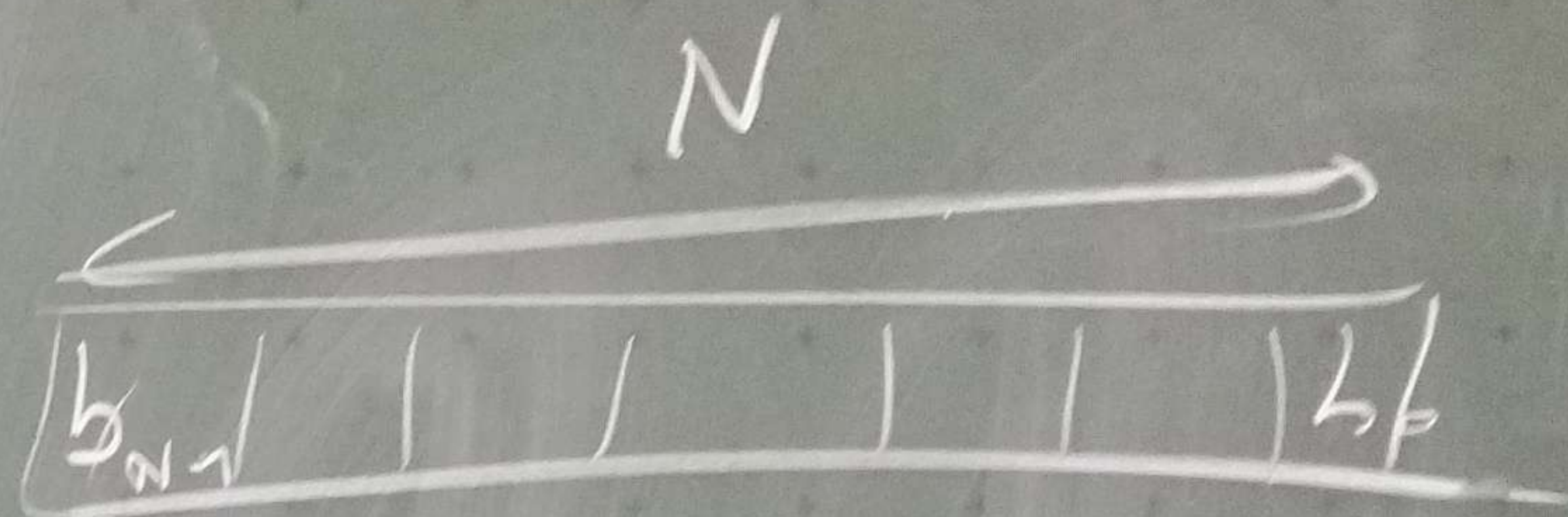
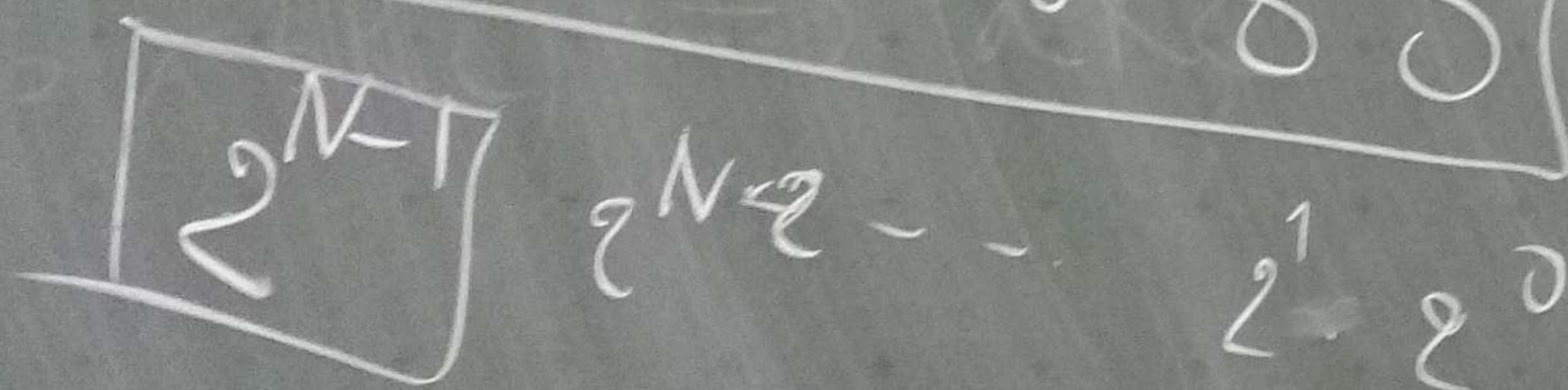
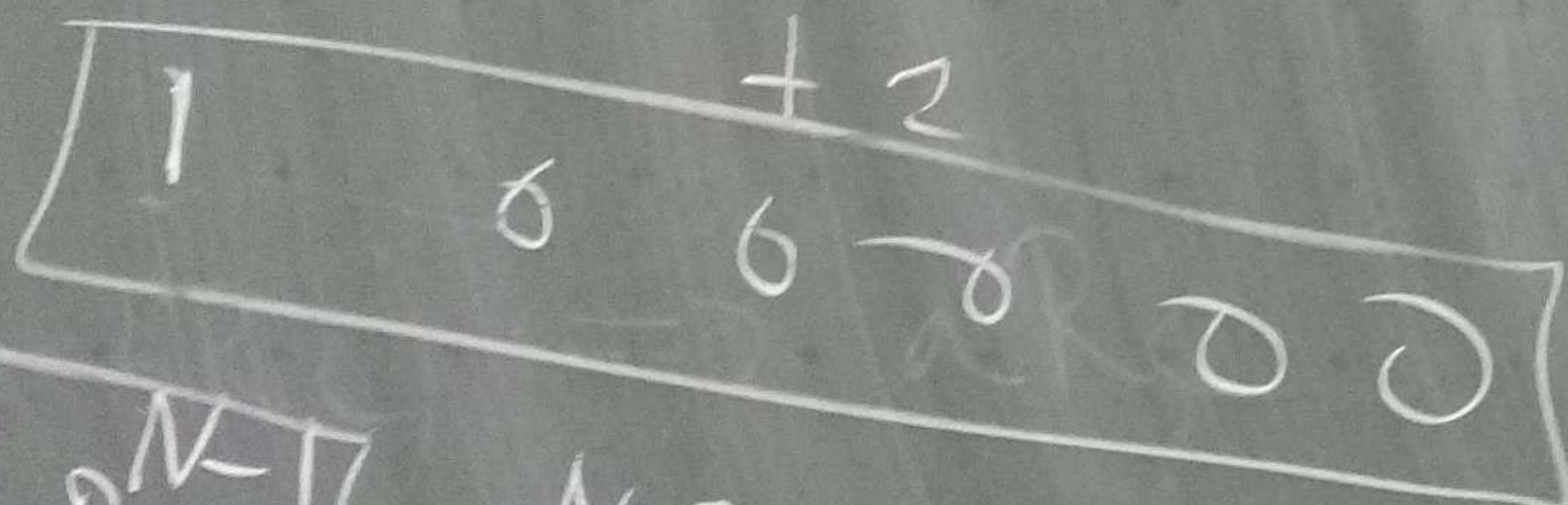
$$2^3 \quad 2^2 \quad 2^1 \quad 2^0$$

$$-(8) + (4 + 2 + 1) = -1$$

N = 7



N-1



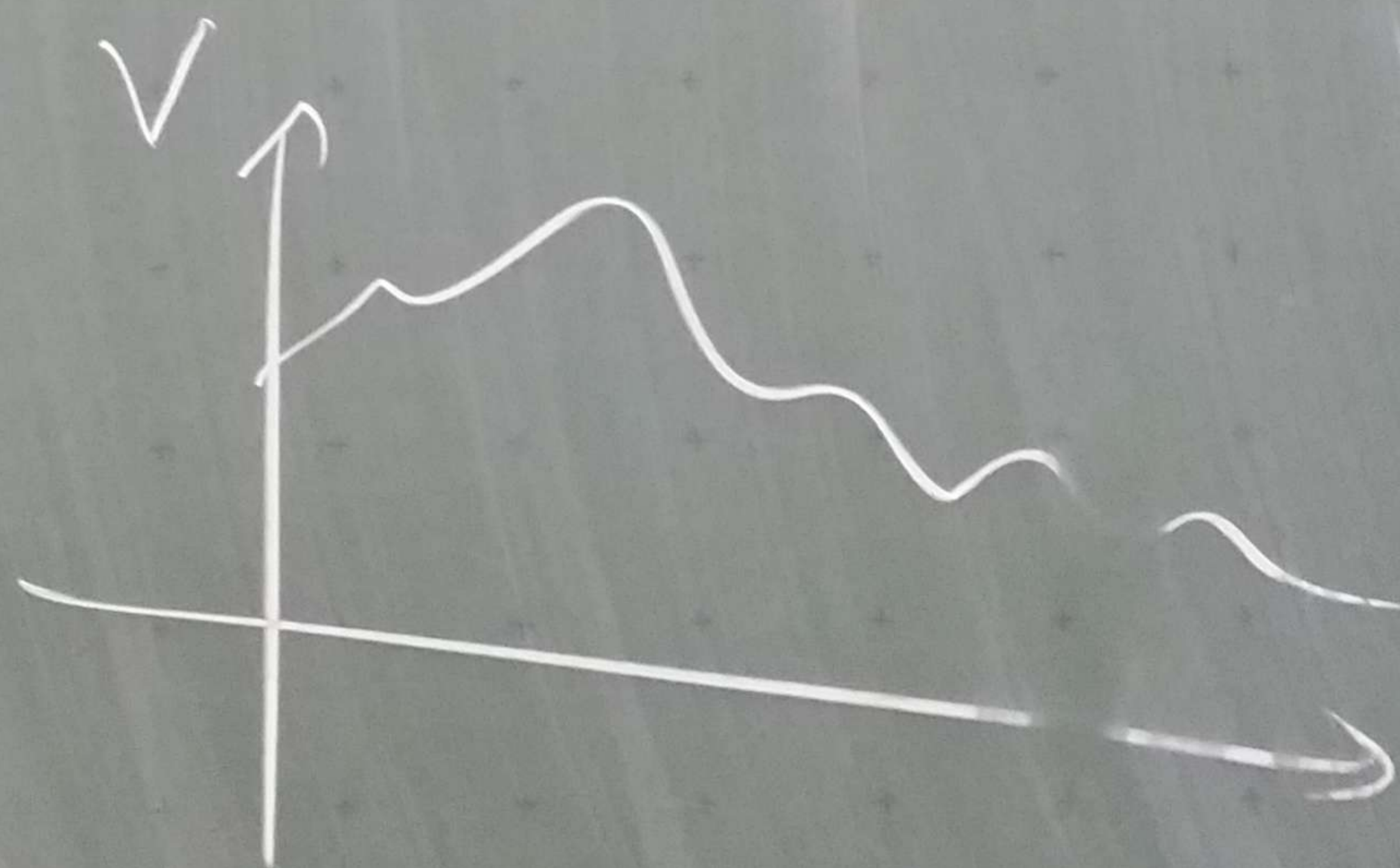
2's COMPLEMENT

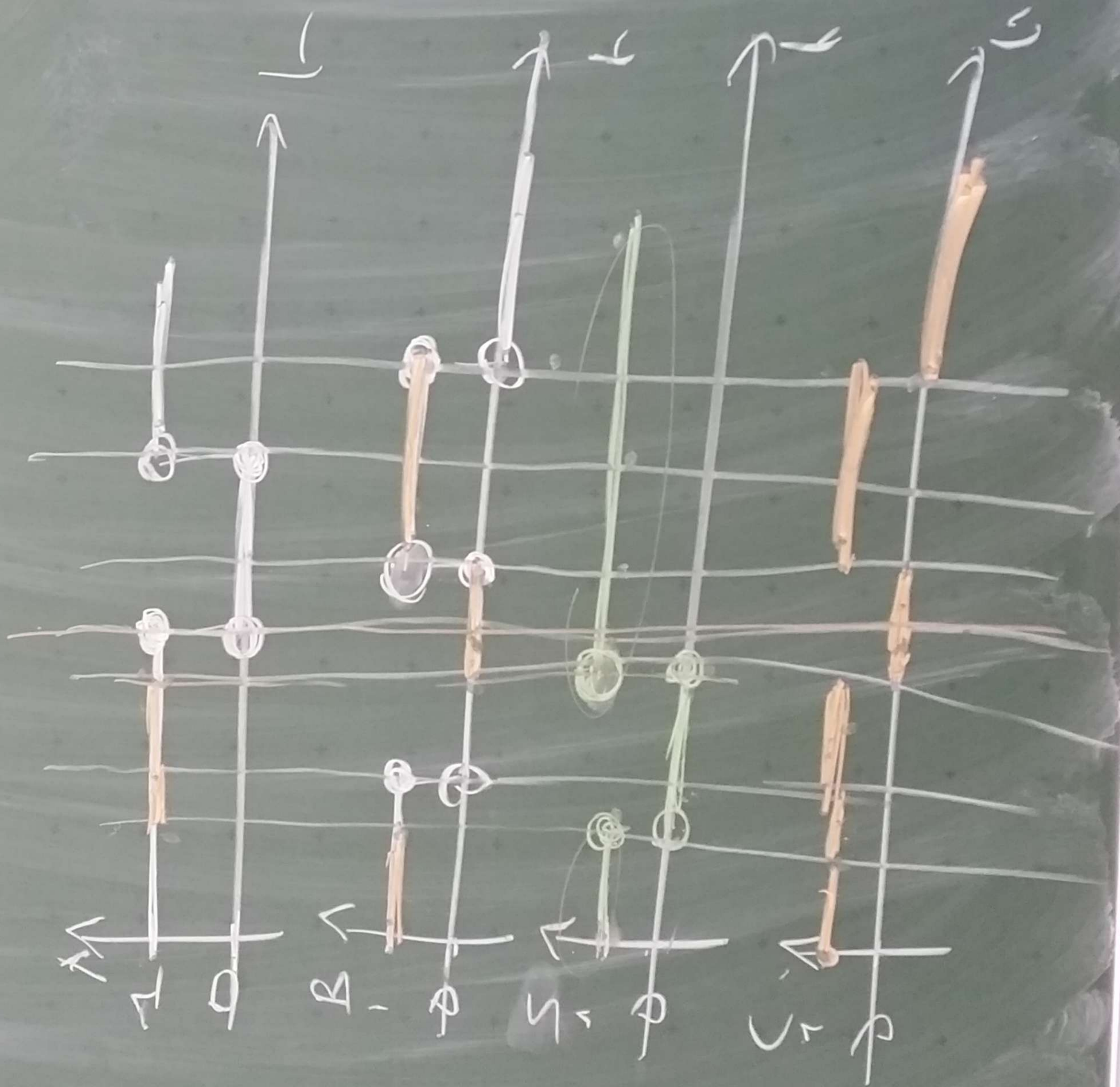
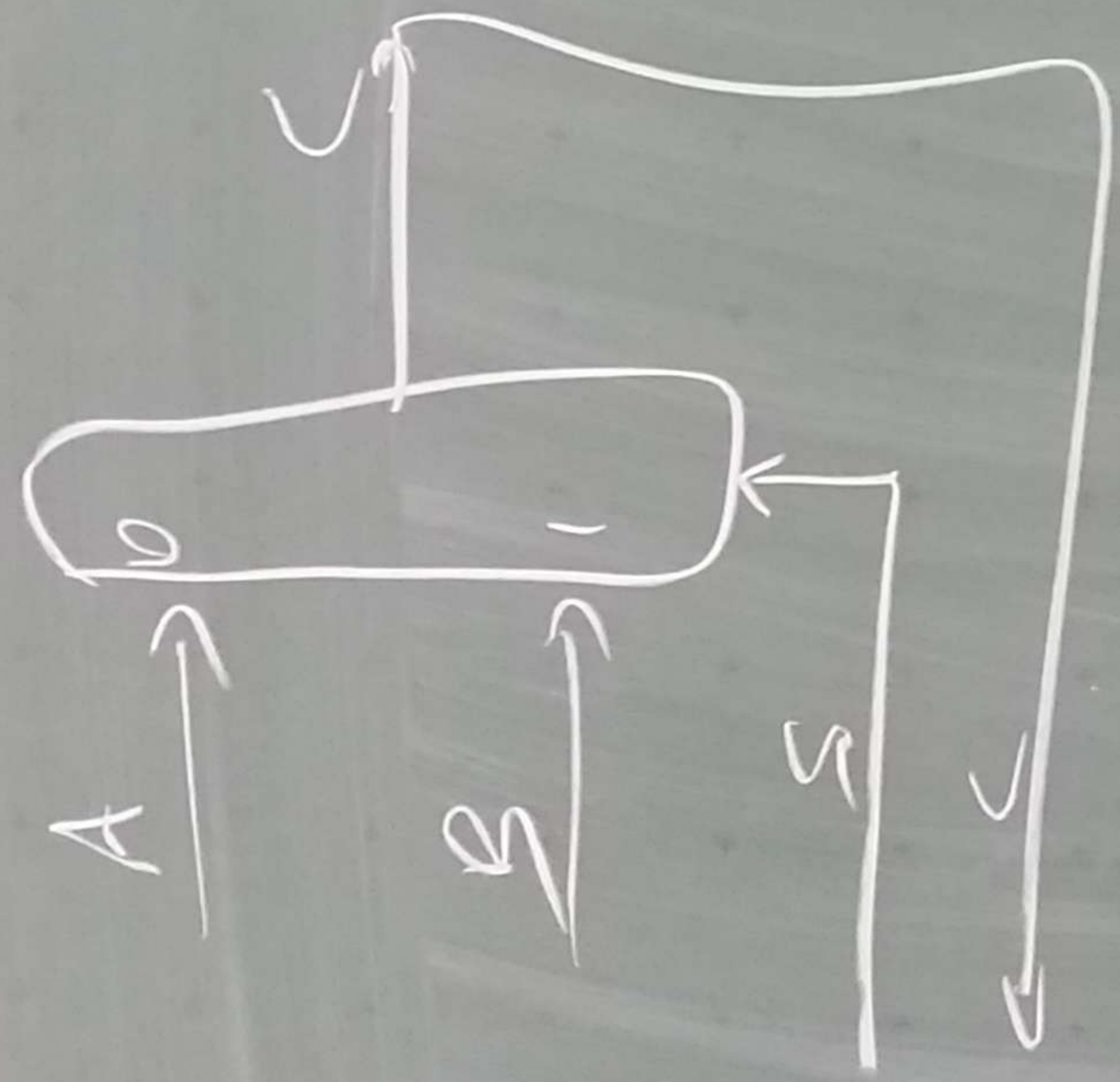
$$2^{N-1} - 1$$

$$- \cancel{b_{N-1}}^{\cancel{N-1}} z + \cancel{b_{N-2}}^{\cancel{N-2}} z^{-1} - \dots + \cancel{b_1}^{\cancel{1}} z + \cancel{b_0}^{\cancel{0}} z^0$$

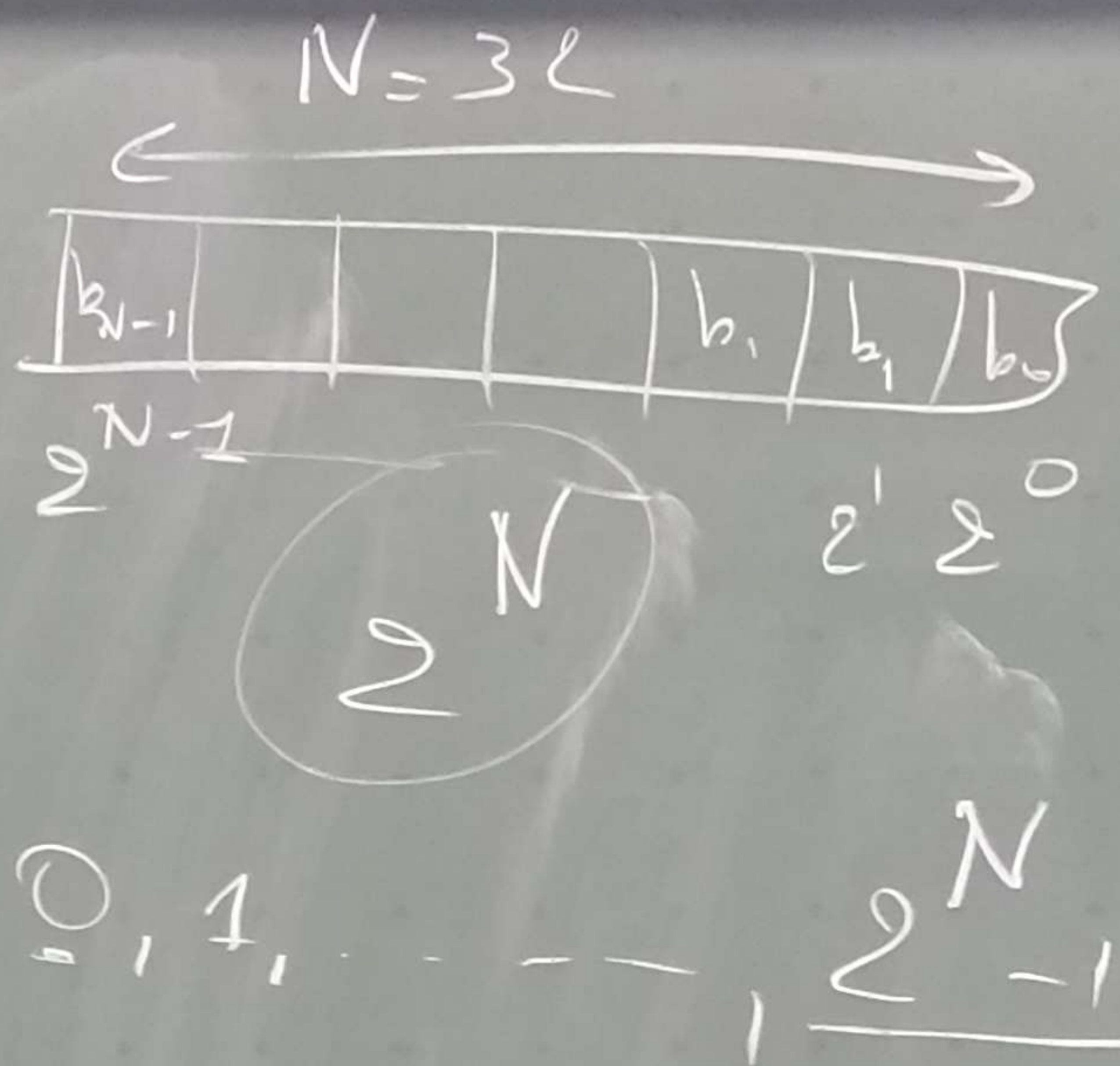
$$- \frac{2^{N-1}}{2^{N-1}} +$$

$$\frac{2^{N-1}}{2^{N-1}} - 1$$





IN



"RANGE"



-3

- A.
- B.
- C.
- D. BIASED

SIGNED MAGNITUDE
1'S COMPLEMENT
2'S COMPLEMENT

$$\begin{array}{r}
 00000011 \\
 11111100 \\
 00000001 \\
 \hline
 11111101
 \end{array}$$

2's COMPLEMENT

$$\begin{array}{l}
 x = +3 \\
 \bar{x} \\
 + 1 \\
 - 3
 \end{array}$$

$$\begin{array}{r}
 x \quad 1101110_2 \\
 + \bar{x} \quad 0010001_2 \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 x + \bar{x} = \\
 \underbrace{111111}_2 \\
 - 1
 \end{array}$$

$$\begin{array}{r}
 x + \bar{x} + 1 = -1 \\
 -x \quad \quad \quad -1 \quad +1 \\
 \quad \quad \quad -2
 \end{array}$$

$$\bar{x} + 1 = -x$$

$$A < B$$

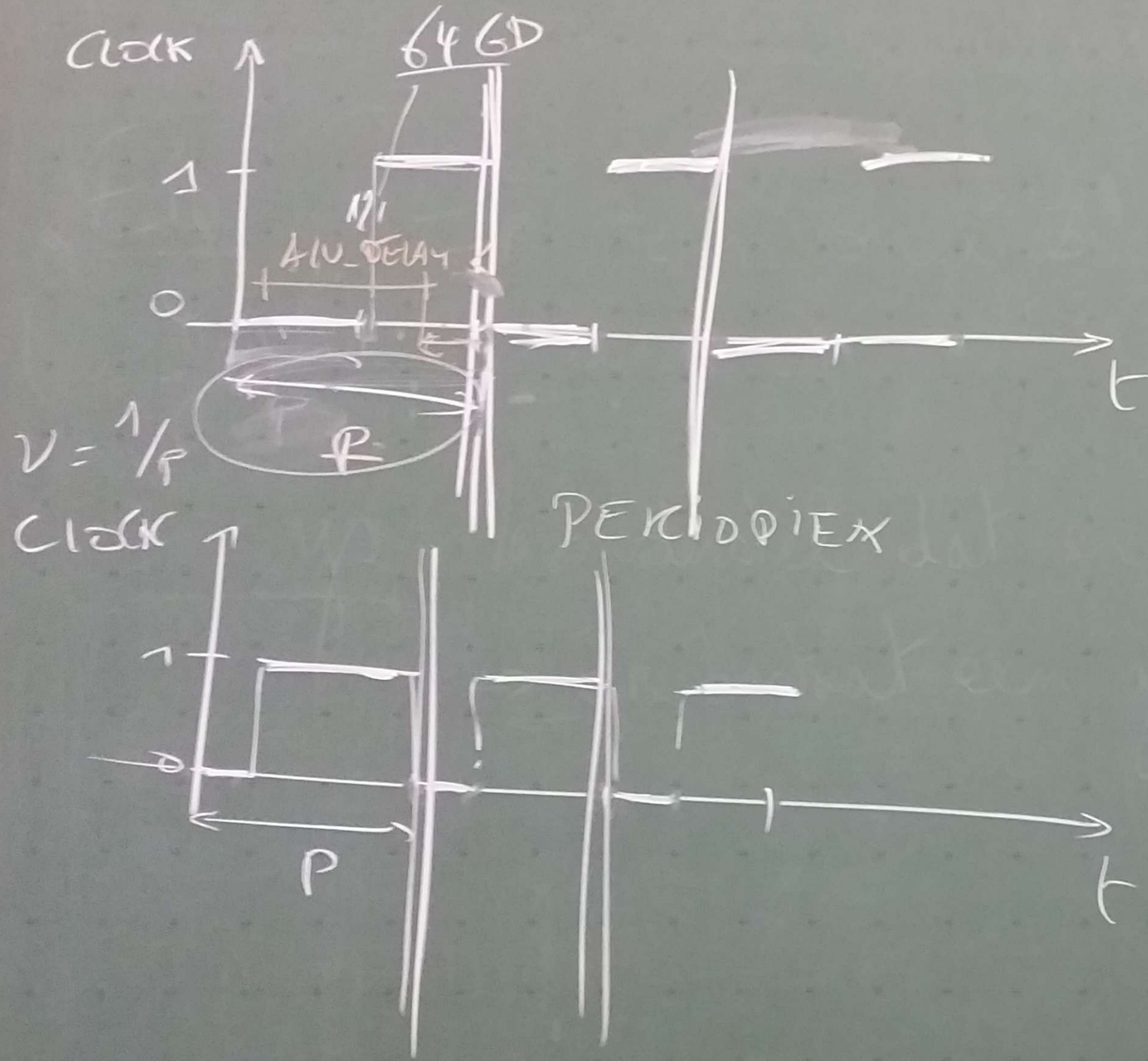
$$A - B < \cancel{B} - \cancel{B}$$

$$\begin{array}{l}
 A - B = A + (-B) \\
 A + B = A + (\bar{B} + 1)
 \end{array}$$

U...

P's

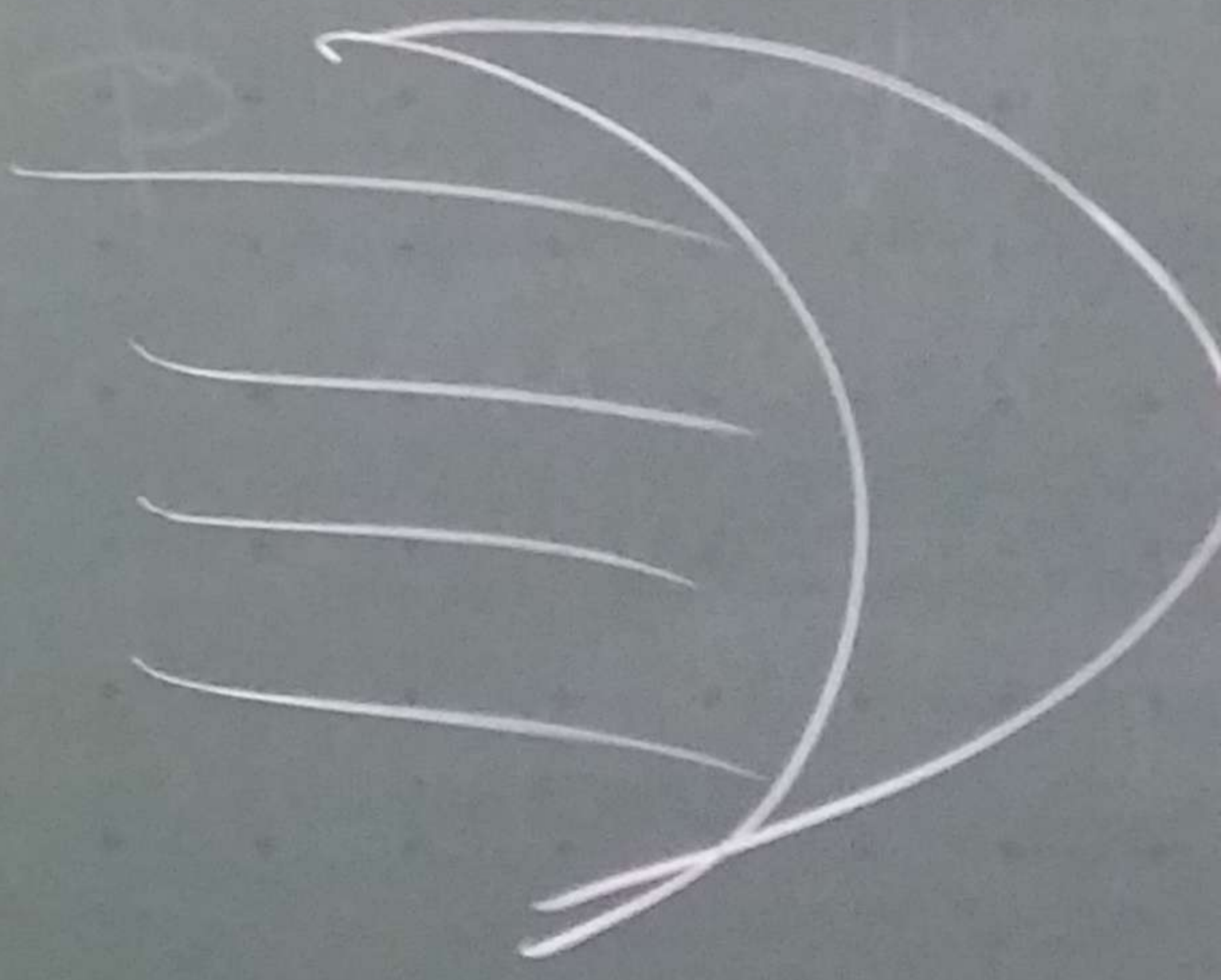
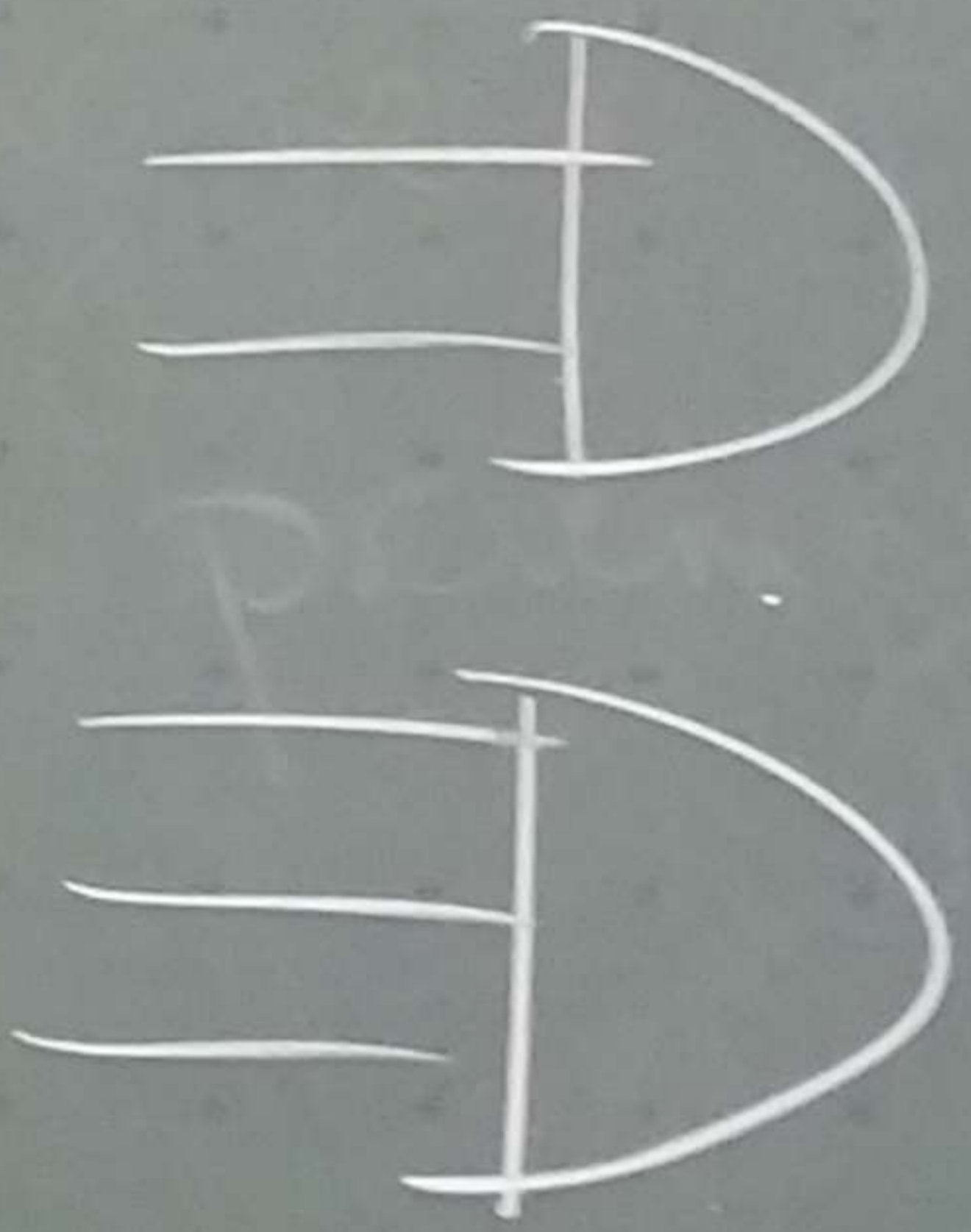
10 AM



$V = 1/P$

2 GD

$7 + 1 = 8$



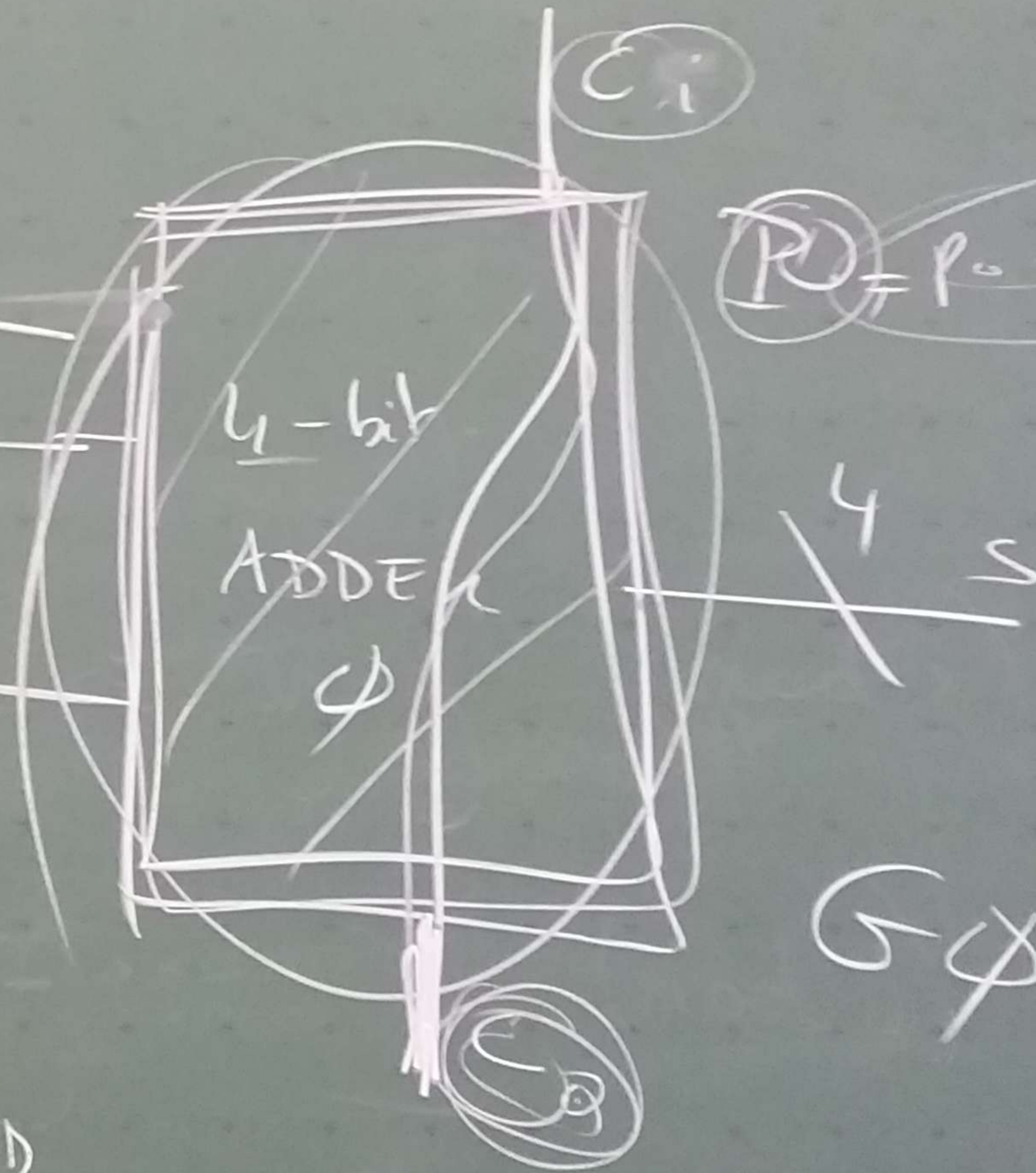
$$P \quad \text{AND OR} \quad 1 + 4 = 5$$

$$G \quad 3 + 1 = 4$$

$$4 + 6 = 10$$

$$LS \quad a \quad 4$$

$$b \quad 4$$



$$P = p_0 \cdot p_1 \cdot p_2 \cdot p_3$$

$$G \quad \phi$$

$$14 + (4 \times 19) \text{ GATES}$$

$$AY \quad C_4 = 3 \text{ G-D}$$

$$C \quad 7 + 1$$

$$\begin{matrix} 10 \text{ AND} \\ 4 \text{ OR} \\ 14 \end{matrix}$$

AND OR
 $1 + 4 = 5$

G $3 + 1 = 4$
 $4 + 6 = 10$

AND $4 + 4 = 2$

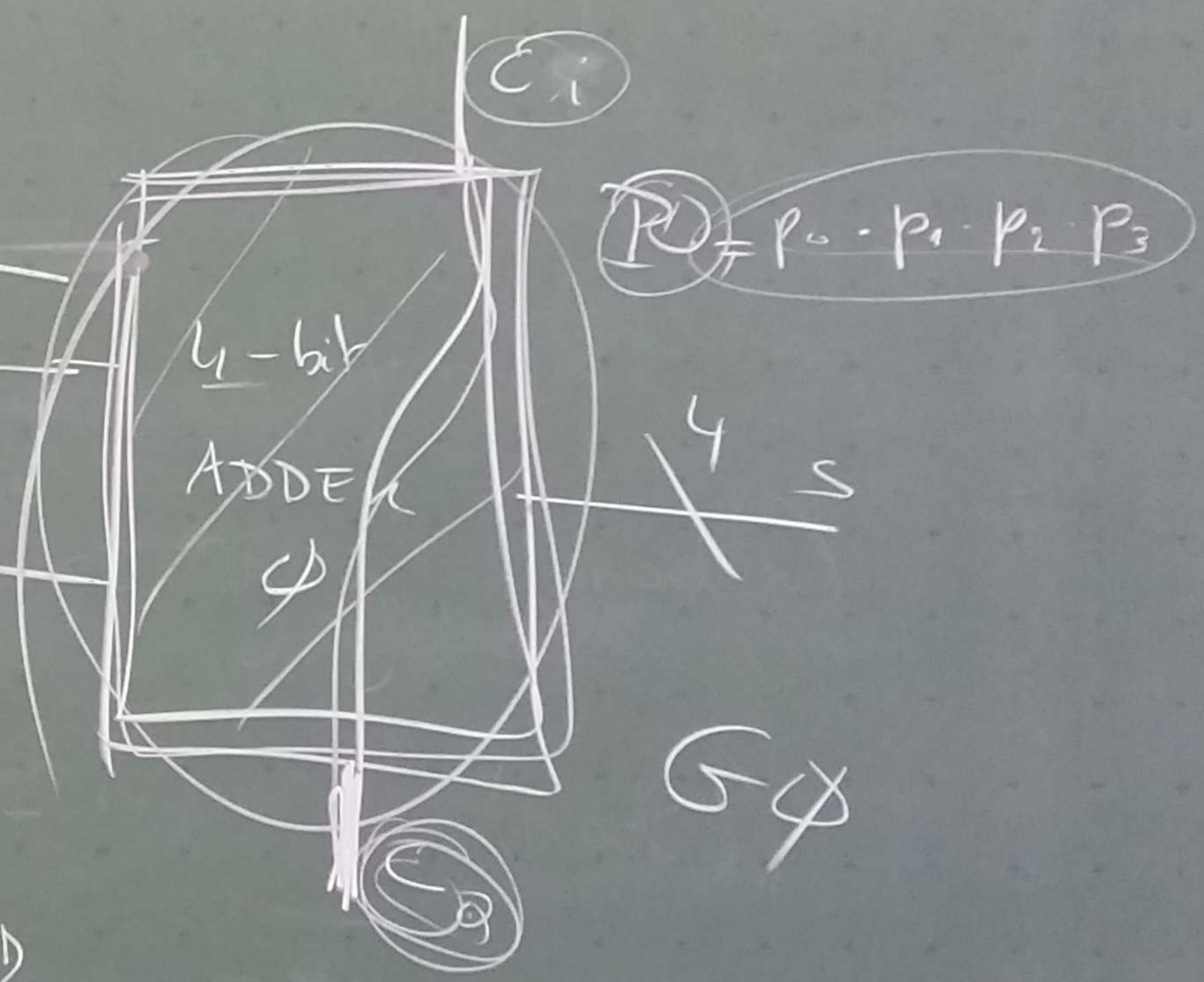
OR $1 + 10 = 11$

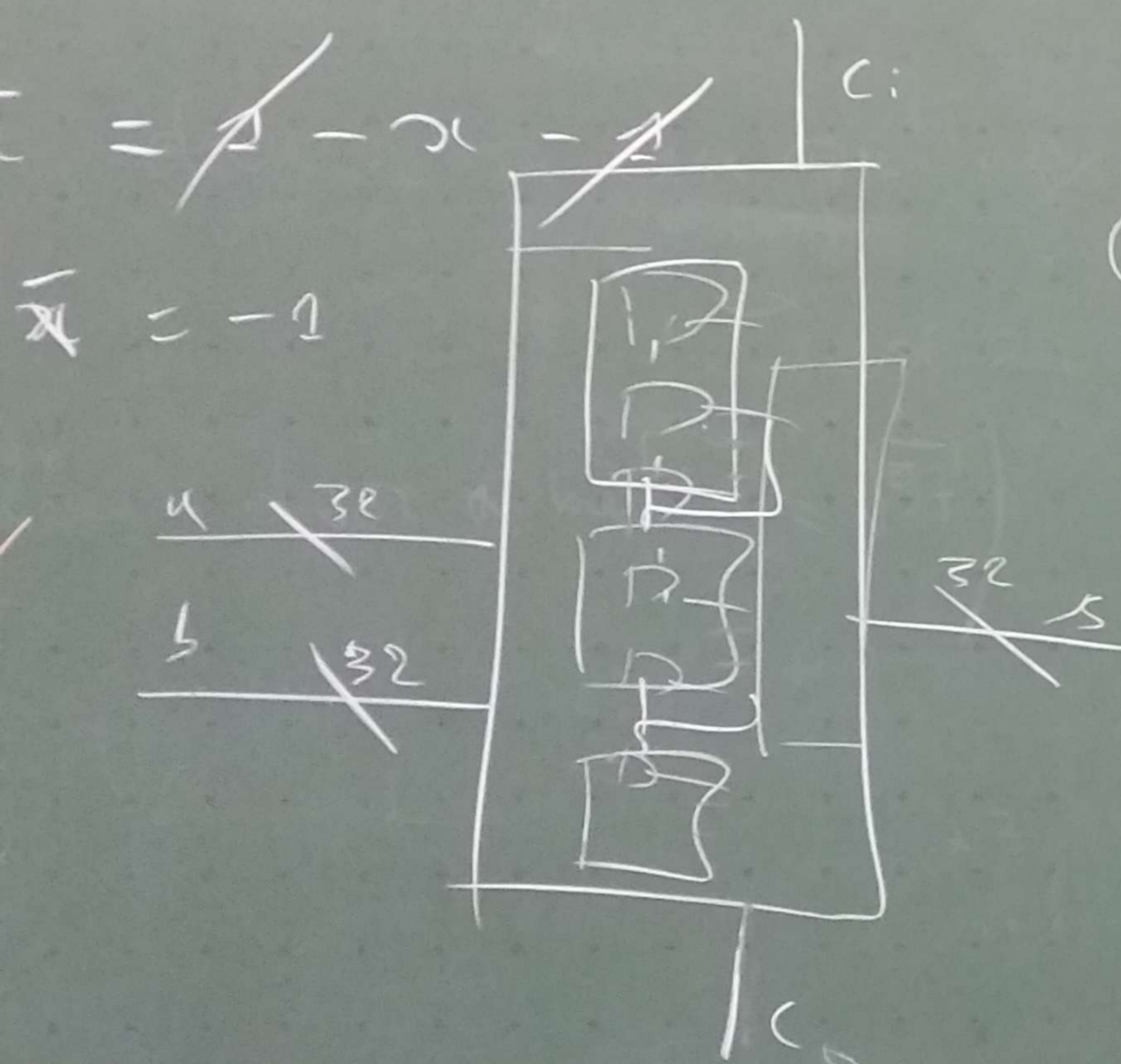
$14 + (4 \times 11) = 58$ GATES

DELAY $C_1 = 3$ GD

P's
 $1 + 1 = 2$
 G's $7 + 1 = 8$

10 AND
 4 OR
 14





$$a_b < b_b$$

$$(a-b) < 0$$

- 16 b.r
- RIPPLe (Atrary)
 - $D = 32 \text{ GD}$
 - # GATES
 - CARRY LOOK-AHEAD
 - 4b, 4x
 - $D = 32 \text{ GD}$
 - INFINTIE HW
 - $D = 2 \text{ GD}$

AND $4 + 4 = 2$

OR $1 + 10 = 11$

$14 + (4 \times 18) \text{ GATES}$

DELAY $C_4 = 3 \text{ GD}$

$\begin{matrix} 1 & 1 \\ + & 1 \\ \hline 1 & 1 \end{matrix}$
 P's
 G's

$7+1$

AND OR

$1 + 4 = 5$

$3 + 4 = 7$

$4 + 6 = 10$

$15 \quad 0 \quad 4$

$5 \quad 4$

$10 \quad 1$

14

IN

unsigned int

BCD

BINARY

Z

signed int

SIGNED MAGN

1's COMPL

2's COMPL

BIASED

IEEE

Q
RUL

fixed point

float/double
32 64

+∞
-∞
NaN



$$\begin{array}{r} \text{EXP} \\ \textcircled{37} \\ \textcircled{112} \textcircled{34} \times 10 \\ \hline \textcircled{33} \end{array} \quad \begin{array}{l} E1 \\ < \\ E2 \end{array}$$

$$8.333 \times 10$$

$$\begin{array}{r}
 11 \\
 \textcircled{111} \\
 + \textcircled{111} \\
 \hline
 \textcircled{110} \\
 \hline
 \end{array}
 \begin{array}{r}
 -1 \\
 -1 \\
 -2
 \end{array}$$

$$\begin{array}{r}
 010 \\
 + 101 \\
 \hline
 111
 \end{array}$$

~~0000~~

X

0 0 0 1

0 0 1 0

0 0 1 1

⋮

~~1 1 1 1 1~~

X

$+\infty$

$-\infty$

$N \in N$

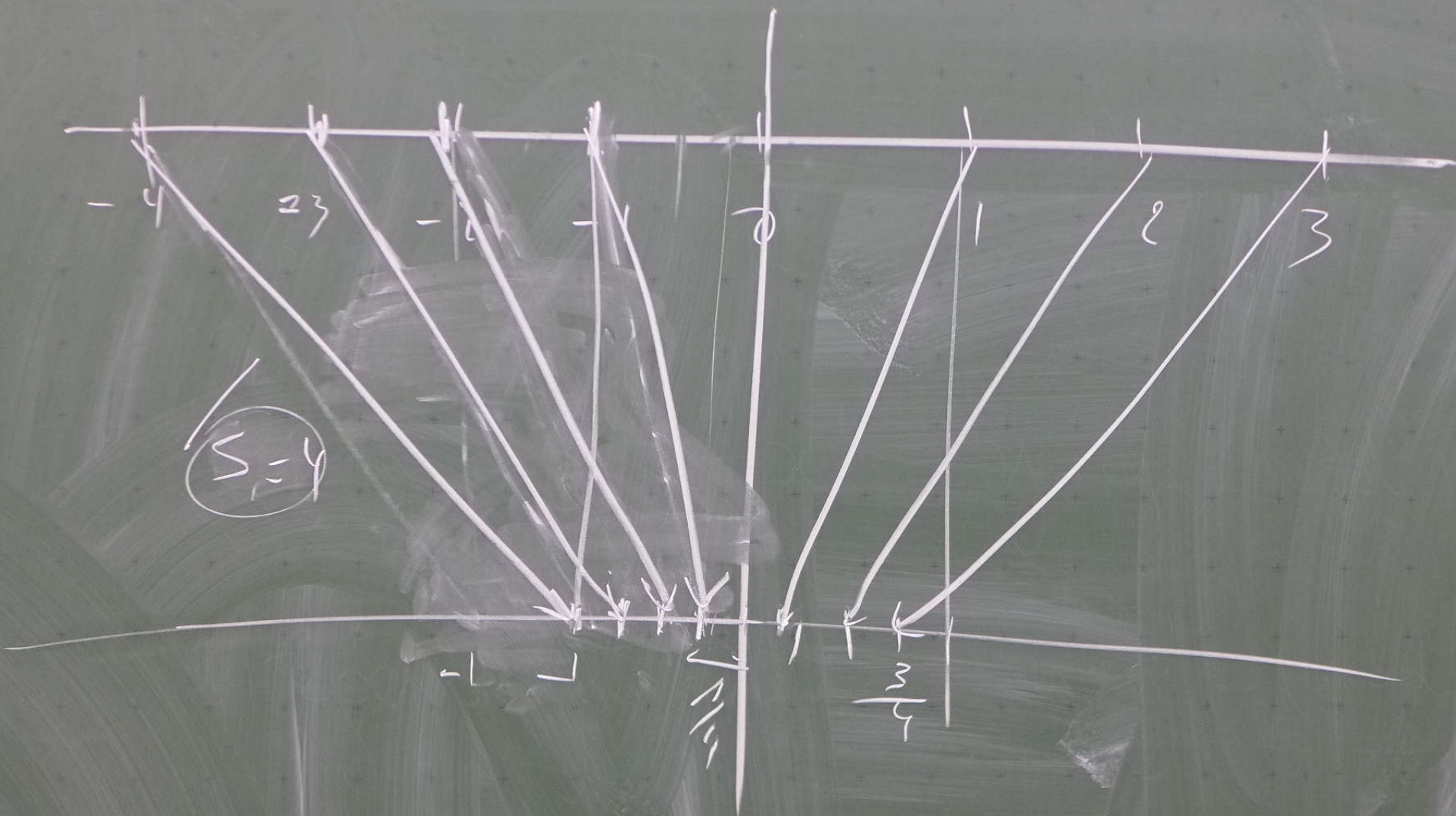
$$\begin{array}{rcl}
 a_{fip} & \longrightarrow & a_i / s_i \\
 \times b_{fip} & \longrightarrow & b_i / s_i \\
 \hline
 \end{array}$$

$$(a_{fip}) \times b_{fip}$$

$$\boxed{
 \begin{array}{l}
 a_i \times b_i \\
 \hline
 s_i^2
 \end{array}
 }$$

$$\begin{array}{r}
 \begin{array}{c} 2 \\ \textcircled{010} \end{array} \longrightarrow \begin{array}{c} 0 \\ \hline 010 \end{array} \\
 \times \begin{array}{c} \textcircled{001} \\ 1 \end{array} \quad \swarrow 5 \quad \begin{array}{c} 0 \\ \hline 000 \end{array} \\
 \hline
 \times \textcircled{010}
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{c} \textcircled{0.5}_{10} \\ 0.75_{10} \end{array} \\
 \searrow \\
 \begin{array}{c} \textcircled{0.5}_{10} \\ \hline 4 \end{array} \neq \underline{\underline{0.125_{10}}}
 \end{array}$$



$$\text{VALUE} \left(\overset{\text{FIXED POINT}}{\underset{S=100}{\text{"000"}}} \right) = \frac{\phi}{\Sigma} = \phi$$

$$\text{VALUE} \left(\overset{\text{FIXED POINT}}{\underset{S=100}{\text{"999"}}} \right) = \frac{999}{\Sigma} = \frac{999}{100} = 9.99$$

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

$$\begin{array}{rcl}
 \begin{array}{|c|} \hline "100" \\ \hline "001" \\ \hline "101" \\ \hline \end{array} & \begin{array}{l} \xrightarrow{/S} \\ \\ \xrightarrow{/S=4} \end{array} & \begin{array}{l} 100_2 \quad (1_{10}) \\ 001_2 \quad (0.25_{10}) \\ 101_2 \quad (1.25_{10}) \end{array}
 \end{array}$$

$\in \mathbb{Q}$

$$\begin{aligned}
 a_{\text{fp}} &= a_i / S \\
 + \quad b_{\text{fp}} &= b_i / S \\
 \hline
 a_{\text{fp}} + b_{\text{fp}} &= (a_i + b_i) / S
 \end{aligned}$$

$$\text{VALUE}(\underbrace{\phi \phi \phi}_{\text{TV}}) = +\phi$$

$$\text{VALUE}(\underbrace{\phi \phi \phi}_{\text{Exc 4}}) = -4$$

$$\text{VALUE}_{\text{FIXEDP}}\left(\begin{array}{c} \text{" 1 1 1 " } \\ S=4 \\ = 2^2 \end{array}\right) =$$

$$\begin{array}{c} \times 2^0 \quad \times 2^{-1} \quad \times 2^{-2} \\ \swarrow \quad \swarrow \quad \swarrow \\ | \quad | \quad | \\ 1 \quad 1 \quad 1 \\ \quad \quad 2 \\ 1.75 \\ 10 \end{array}$$

$$N=3$$

$$S=4$$

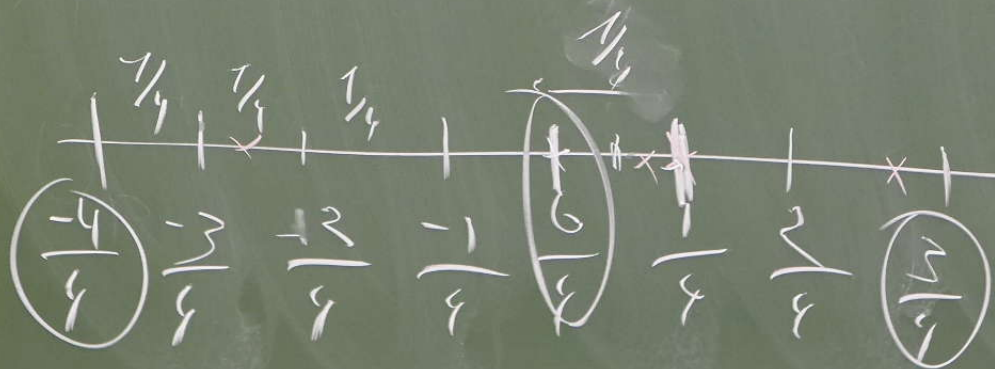
SMALLER

$$\frac{2^{N-1}}{S}$$

LARGEST

$$\frac{2^{N-1}-1}{S}$$

$$\text{DISTANCE} = \frac{1}{S} - 1$$



$$\textcircled{+} \textcircled{0.12345} \times \textcircled{10}^{-12}$$

LARGE

$$b = 10$$

$$\Delta = 5$$

$$\boxed{W} = 55$$

$$\boxed{m} = -99$$

SMALLER

SMALLER

RANGE

LARGES

SMALLER

12

LARGEST REPR = $0.99999 \times 10^m + (1 - b^{-1}) \times 10^m$

SMALLEST REPR

SMALLEST REPR = $0.10000 \times 10^m = \frac{1}{b} \times 10^m$

RANGE = $\left[-\frac{(1-b^{-1})10^m}{2}, \frac{(1-b^{-1})10^m}{2} \right]$

LARGEST GAP = $0.00001 \times 10^m = \frac{1}{b} \times 10^m$

GAP $\approx$ $\frac{1}{2} (1-b^{-2}) 10^m$

SMALLEST GAP = $0.00001 \times 10^m = \frac{1}{b} \times 10^m$

199

10

b^m

b

$0.99999 = 1 - 1 \times b^{-1}$

$+ 0.00000$

$+ 1.00000$

$(b-1) \times b^{-1} + (b-1) \times b^{-2} + \dots + (b-1) \times b^{-m}$

$\sum_{i=1}^m b^{-i}$

$i=1$

b^{-1}

b^{-2}

b^{-3}

b^{-4}

b^{-5}

b^{-6}

b^{-7}

b^{-8}

b^{-9}

b^{-10}

b^{-11}

b^{-12}

b^{-13}

b^{-14}

$$+ \underline{11.0110} \times 2^{\textcircled{2}}$$

$$\underline{1101.10} \times 2^{\circ}$$

$$0.000 \times 2^{\textcircled{1}}$$



CANONISCH

$$\underline{A} \quad \cancel{0}.\underline{1}10110 \times 2^{\textcircled{4}} \quad \text{EXP} \quad \boxed{4}$$

$$\underline{B} \quad \cancel{1}.\underline{10110} \times 2^3$$

$$\boxed{1}.\underline{00000} \quad \text{EXP} \quad \boxed{0}$$

$$-(2^N - 1)$$

$$2^N - 1$$

$$\times 10^{-1}$$

$$0.0254 \times 10^3$$

'GANDNISAT'

$$0.0254 \times 10^{-1}$$

ASCII 7-bit

EXTENDED ASCII 8bit

EBCDIC 8bit

UNICODE

UTF-8
- UTF-16
- HTML

0 - 127
ASCII

10FFFF

FONT A A SANS SERIF
A L SERIF

$V=IR$ $\neq$ Ω

CHARACTER 2

Glyph M

UNIT OF ELECTRICAL RESISTANCE

UPPER CASE 1
GREEK OMEGA

CHARACTER 1 $\neq$ CHARACTER 2

LOWER CASE LATIN f

LOWER CASE LATIN i

Glyph FONT

LIGATURE

Glyph Ω



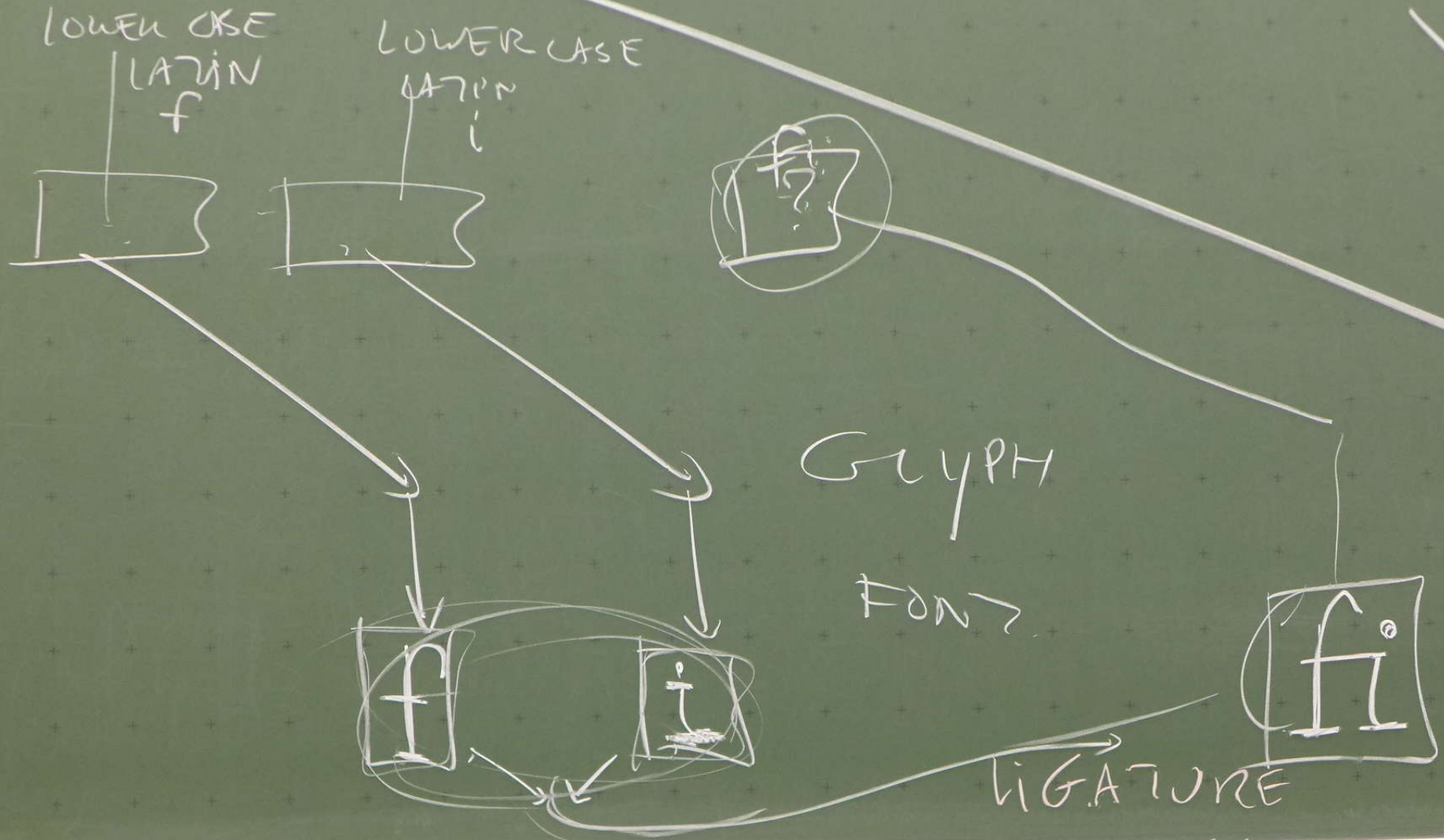
CHARACTER 2

CHARACTER 2

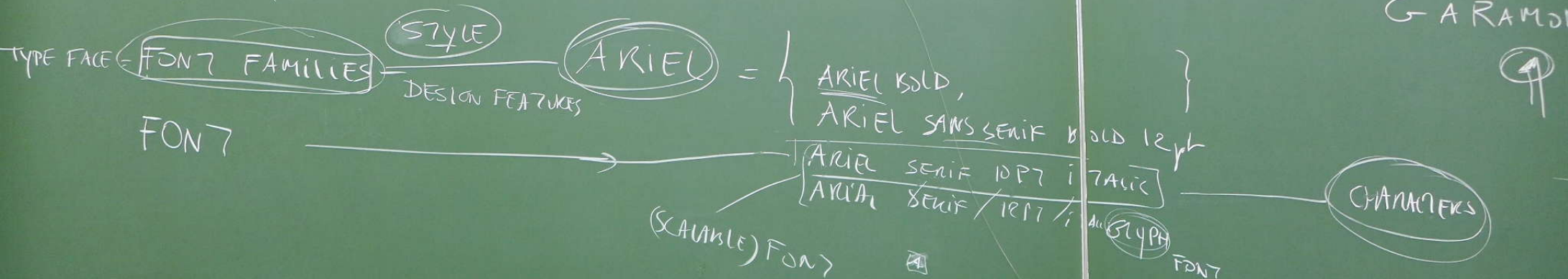
ГЛУХ

M

CHARACTER 1 $\neq$ CHARACTER 2



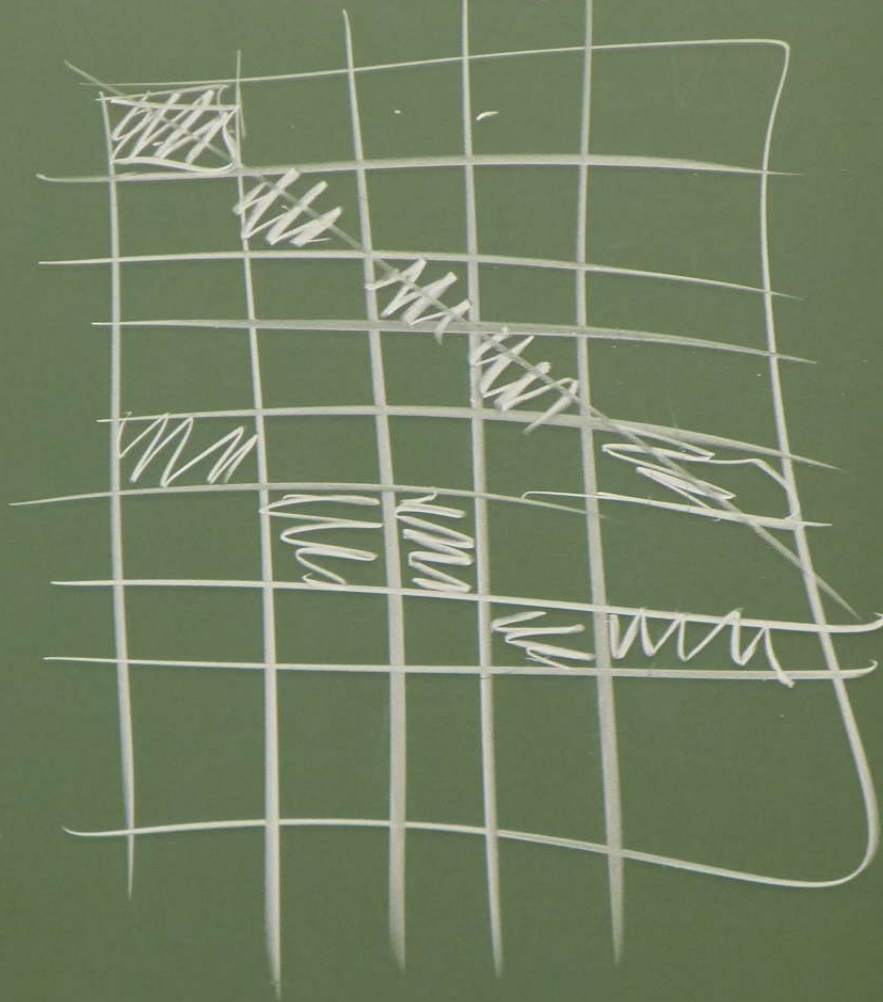
TYPE (-SETTING)



PRINT

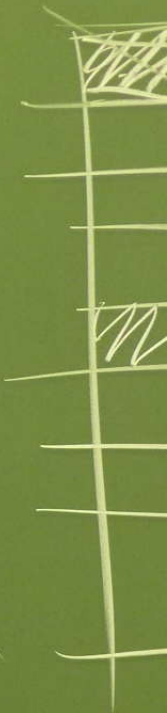
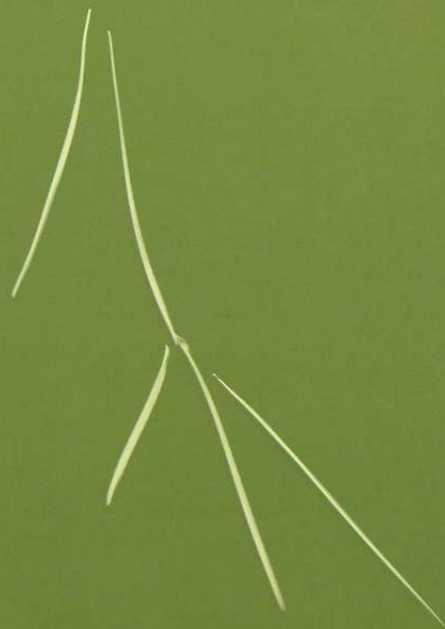
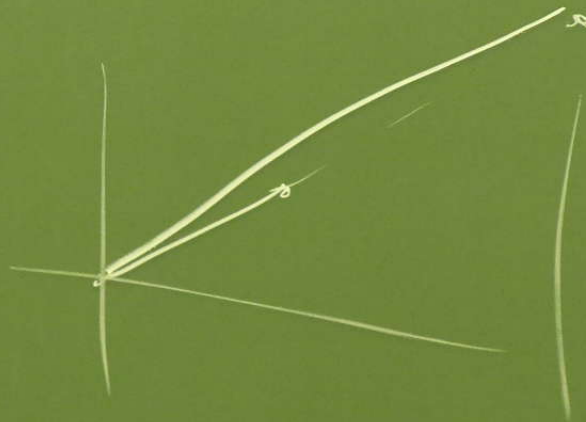
SCREEN

1600



G A RAMOND

1600



CHARACTERS

⚡ HIGH-LEVEL LANGUAGE CODE
C, PYTHON

$$A = (B + C) * D;$$

↓ COMPILER / IN GCC

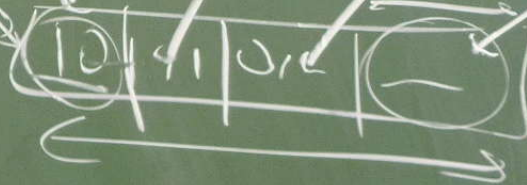
ASSEMBLY LANGUAGE CODE

ADD

$\$R4, \$R1, \$R3$

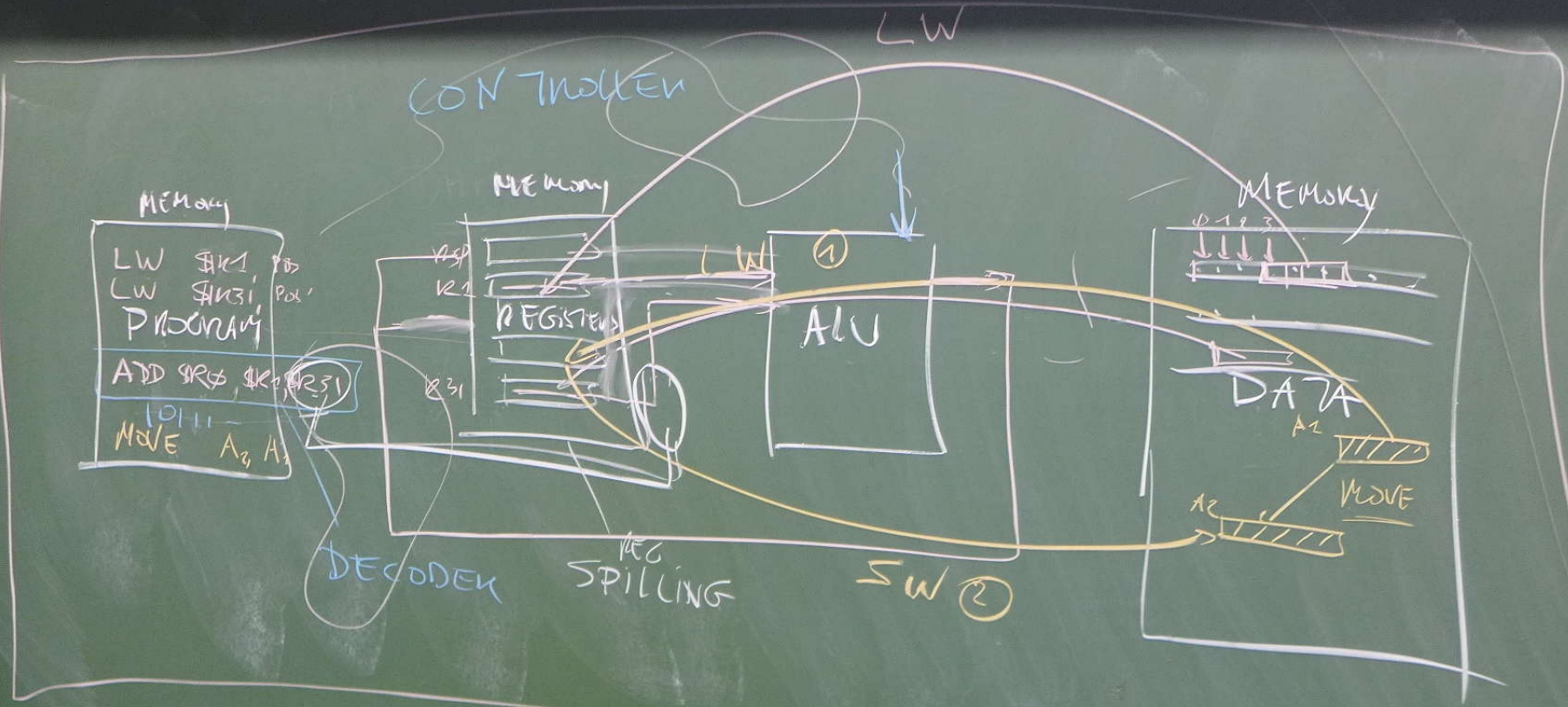
ASSEMBLER (COMPILER)

MACHINE CODE

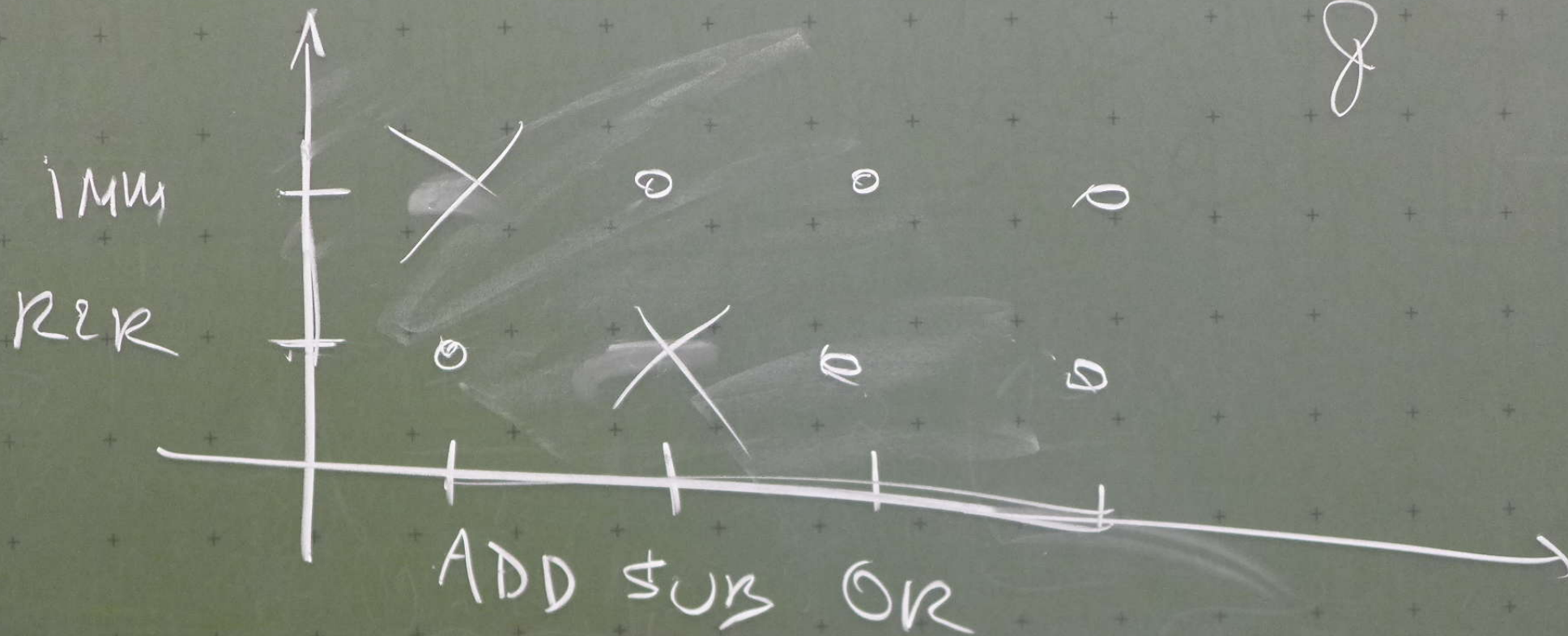


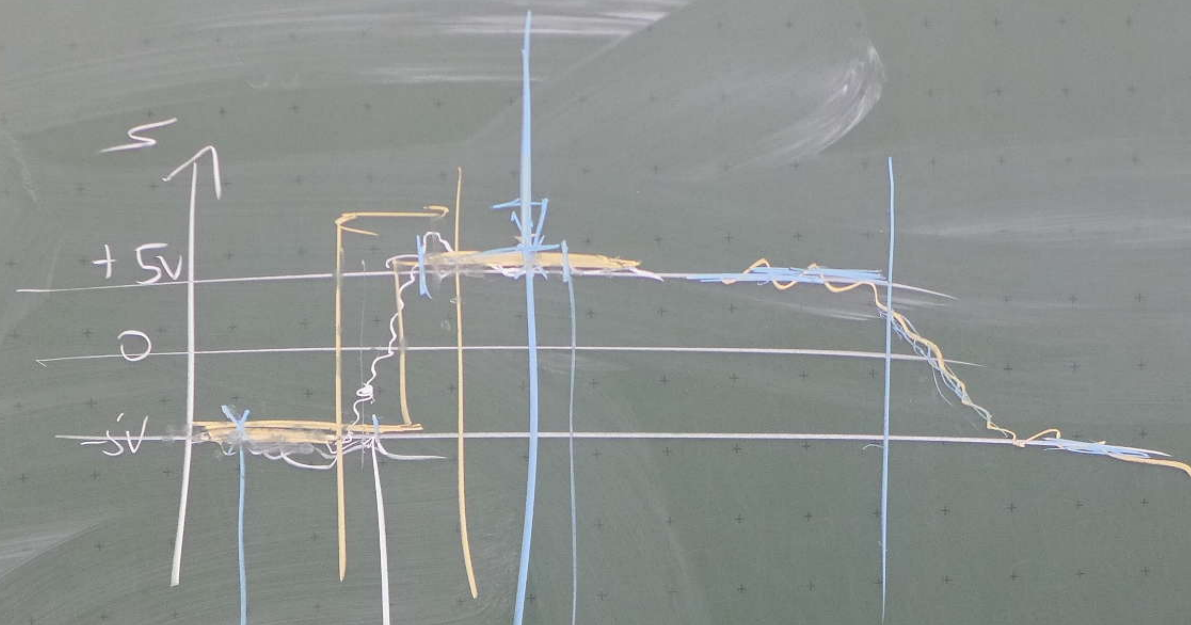
HARDWARE / DATA

DATA PATH



ADD R1, R2, R3





~~$x^2 + y^2 = 1$~~

$$N$$
$$\textcircled{2^b}$$



$$b = \lceil \log_2 N \rceil$$



EP

EP

RAM

R/W

ROM

RO

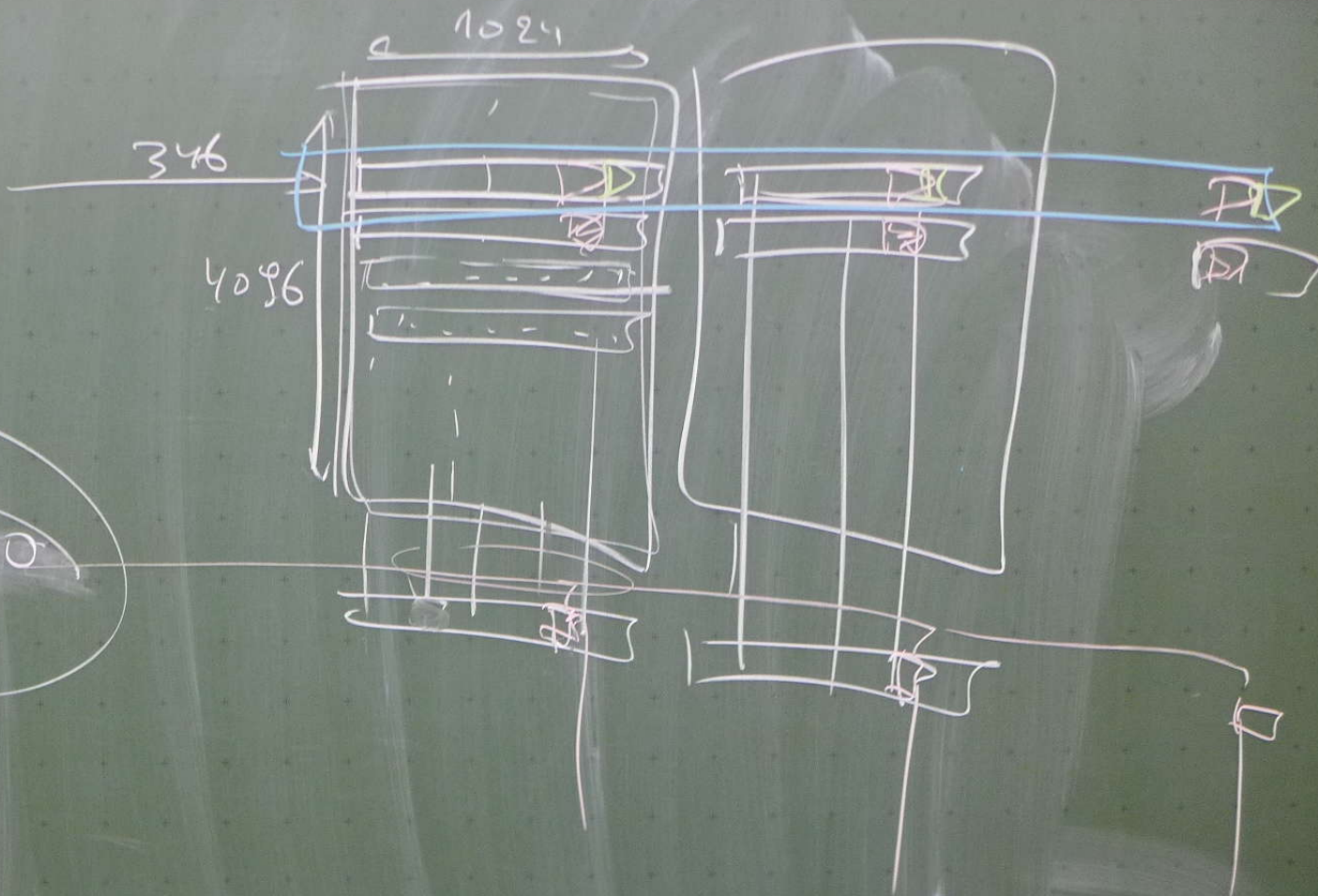
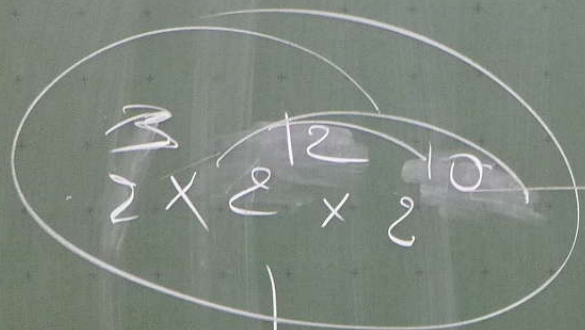
22

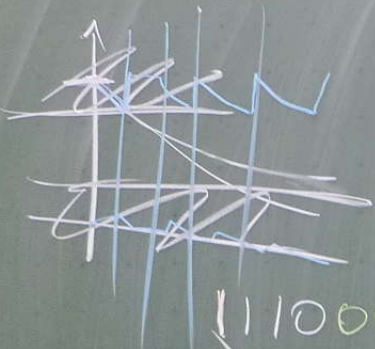
222

3 12 10
2 x 2 x 2

R/W

RO





10001101

RAM

EP

ROM

$\lceil \log_2 2^{21} \rceil$
 1110011, 0010011



0001011

2^{22}

6
A. AND \$R2, \$R2, \$ZERO

B.
NOT
AND

\$R3, \$R2

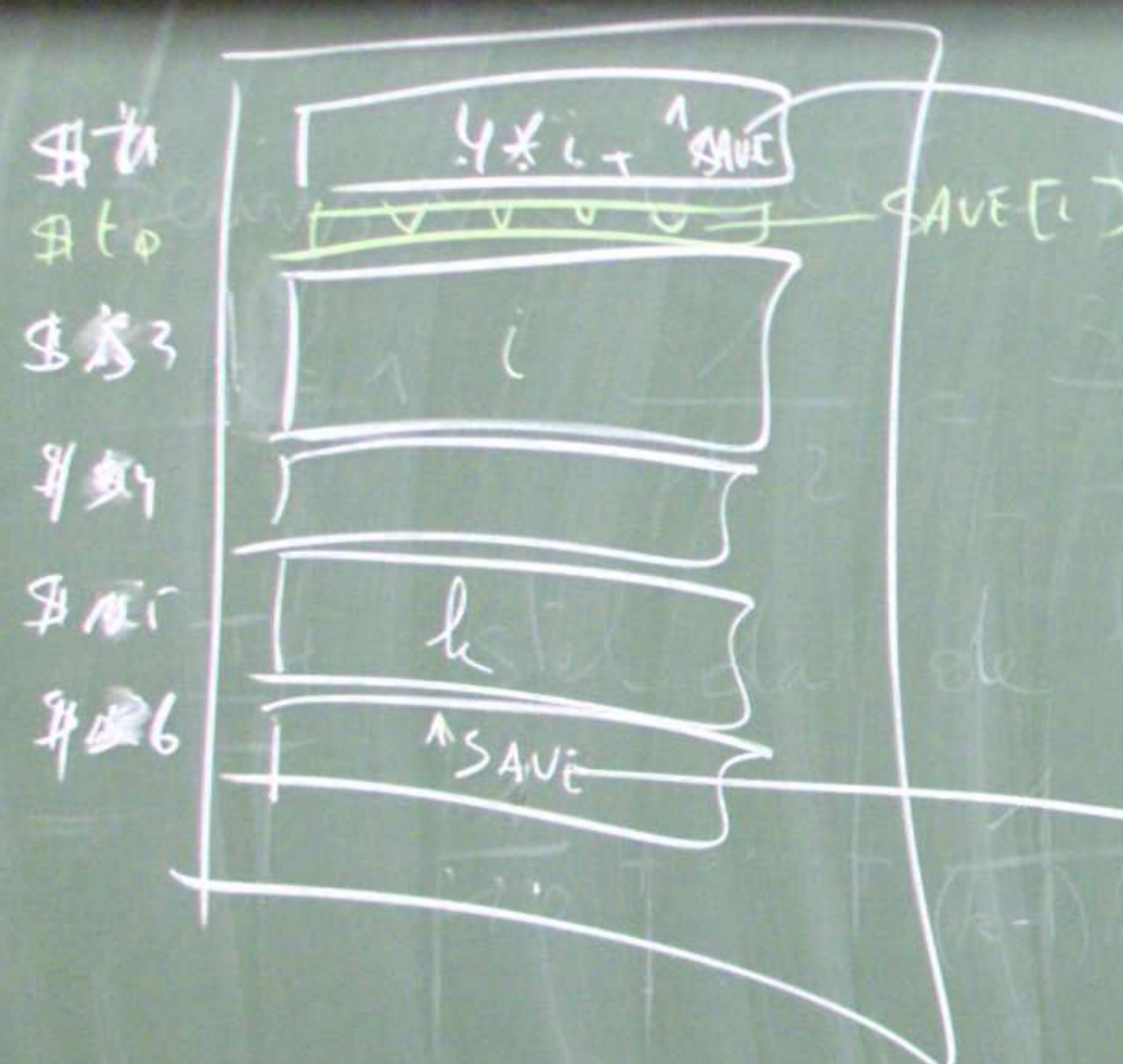
\$R2, \$R2, \$R3

1
2
3

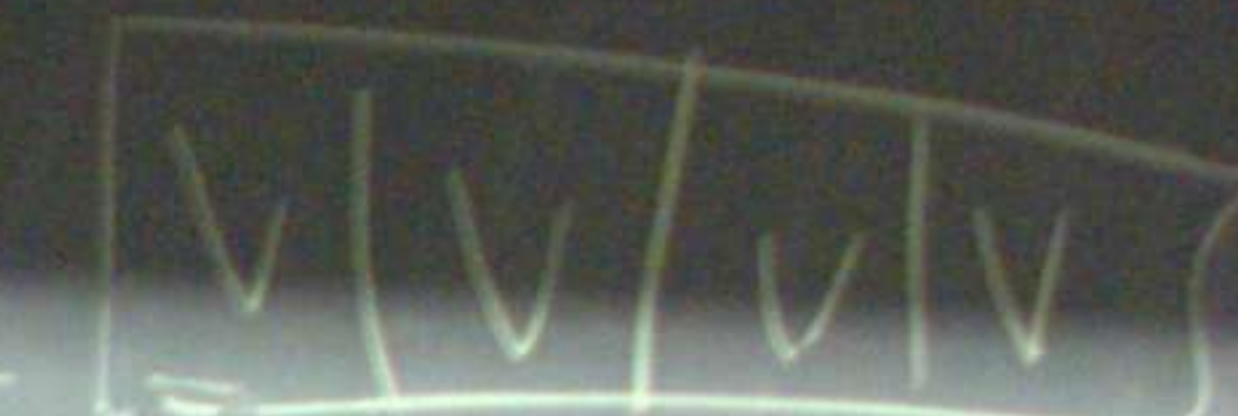
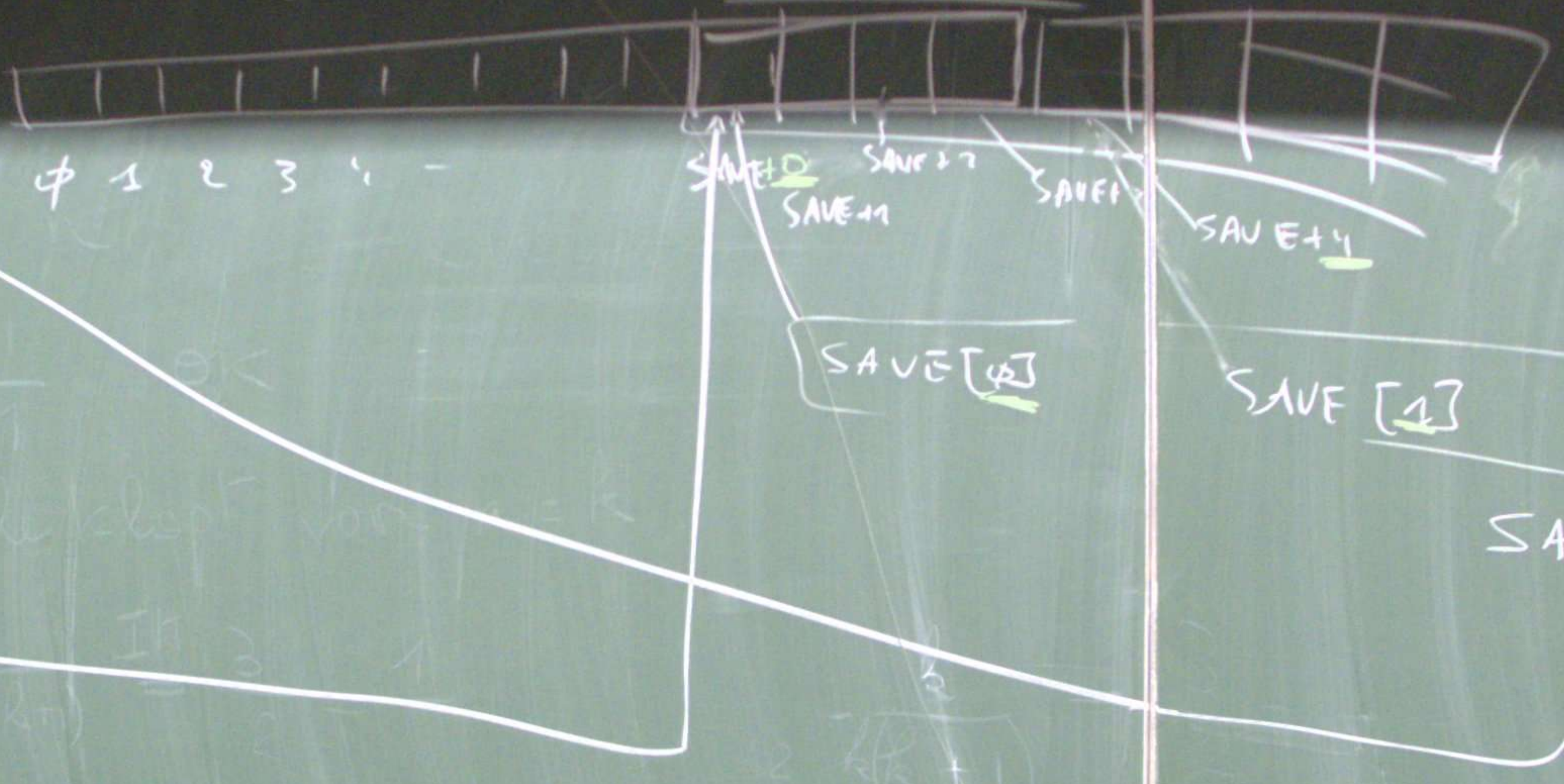
SET $\{R?, \emptyset$

OP	msg	U —————	o
6	5	12	

REG



DATA MEM SAVE



$\text{SAVE} + 4 \times i$

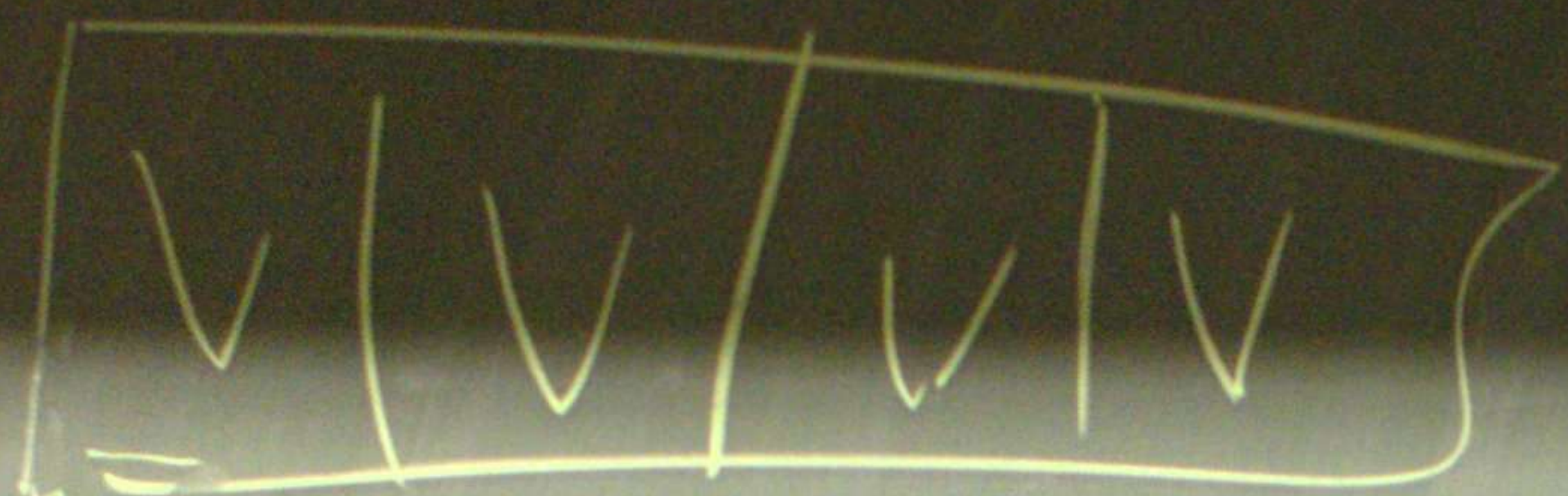
SAVE[i]

STRENGTH REDUCTION

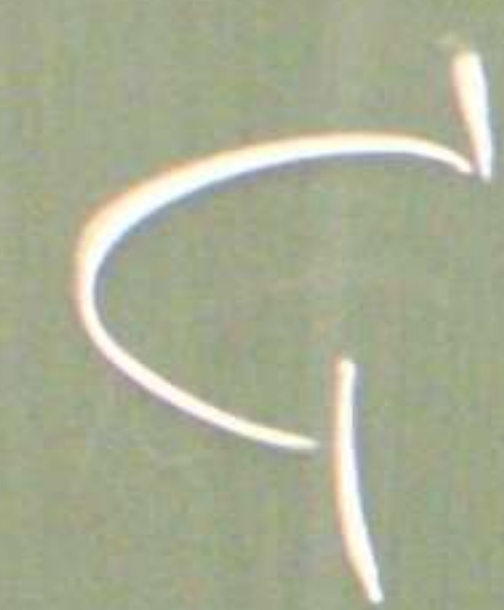
i	0	1	2	3	4	5
$4 \times i$	0	4	8	12	16	20
$\text{SAVE} + 4 \times i$	SAVE	$\text{SAVE} + 4$	$\text{SAVE} + 8$	$\text{SAVE} + 12$	$\text{SAVE} + 16$	$\text{SAVE} + 20$

← 4 ← 4 ← 4 ← 4

bft
bgt
:



$$\text{SAVE} + \underline{4 \times i}$$



SAVE[i]

STRENGTH REDUCTION

i	0	1	2	3	4	5	
4xi	0	4	8	12	16	20	
SAVE + 4xi	SAVE	SAVE + 4	SAVE + 8	SAVE + 12	SAVE + 16	SAVE + 20	
	← 4 → ← 4 → ← 4 → ← 4 →						

bgt

\$t1, \$t2, L

bgt

PSEUDO-

INSTR

add

\$a1, \$a2

bgt \$r1, \$r2, L

bgt PSEUDO-
INSTR

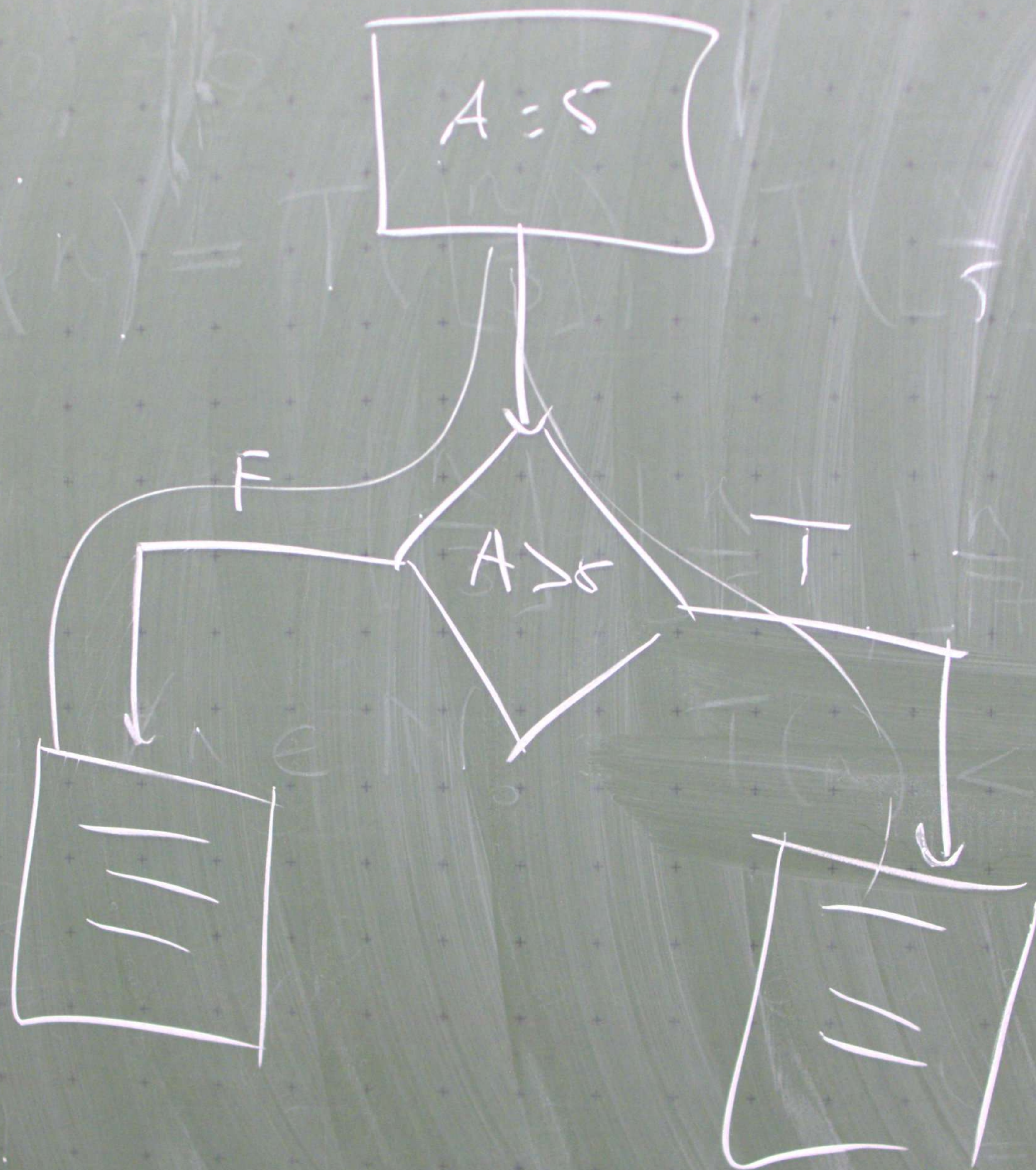
NOTH REDUCTION

add

\$a1, \$r1, \$r2

3 4 5

12 16 20



$\frac{11}{6}$

inductive

$i = 0$
 $j = -1$

if
else

$\left(\frac{i}{7} = j+1 \right)$

BB1

BB2

$j = a + Lxc$

BOEHM & JACOPINI

①

SEQ

②

IF

③

LOOP

PSEUDO-INSTRUCTION

BLT \$s1, \$s2, L

→ SLT

→ BNE

\$s1, \$s2, \$t4

\$t0, \$ZERO, L

ISA

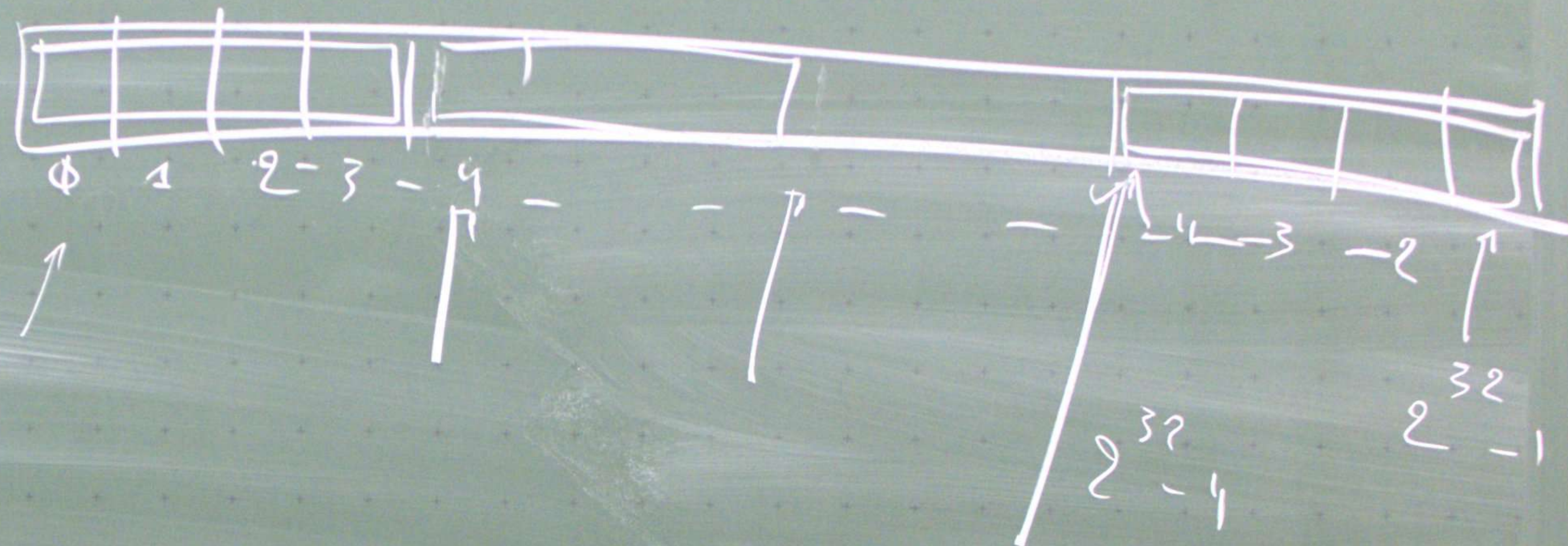
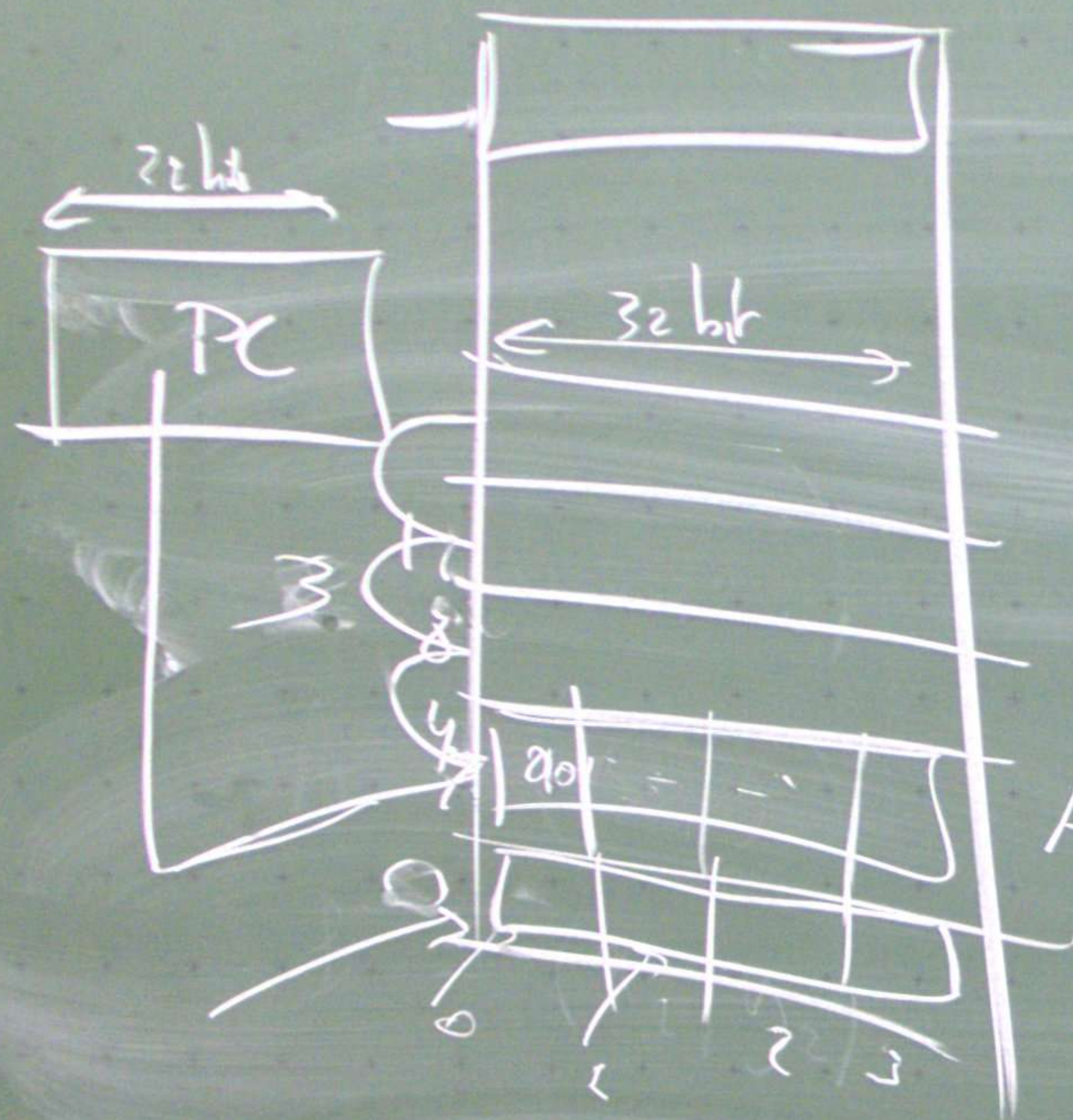
1 ϕ

5

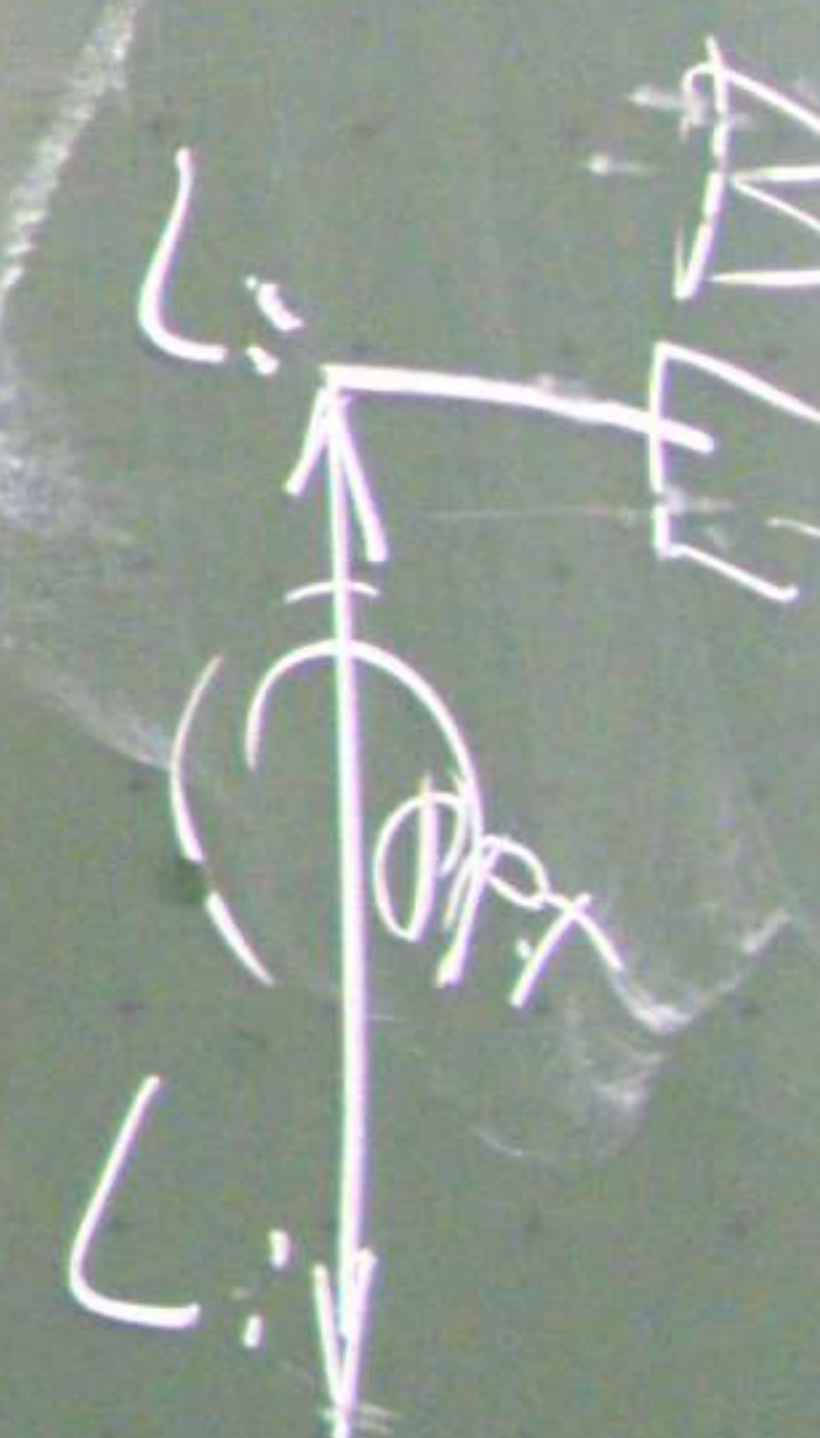
SN

L:

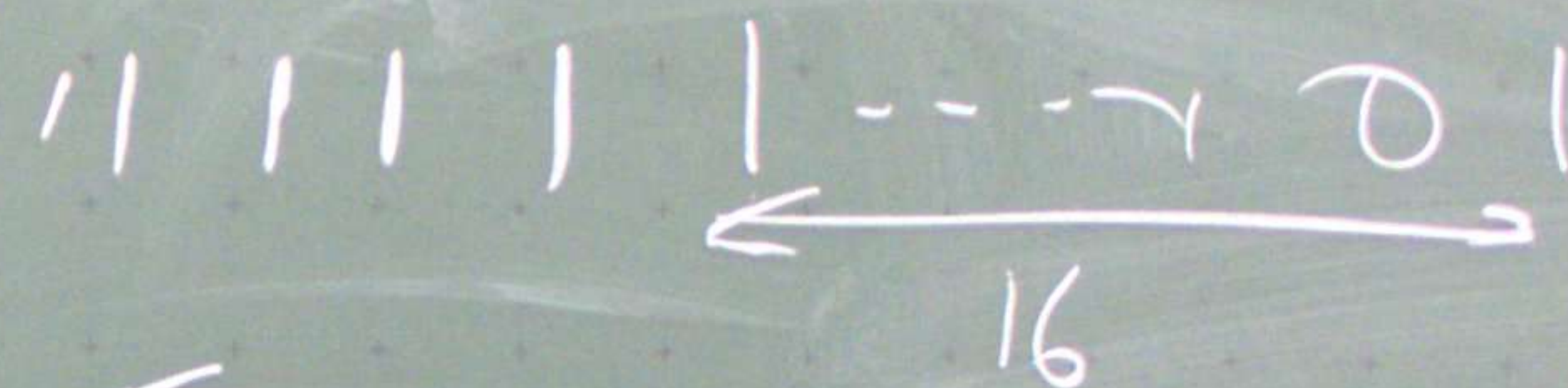
INST MEM



ADD \$12, \$0, \$2



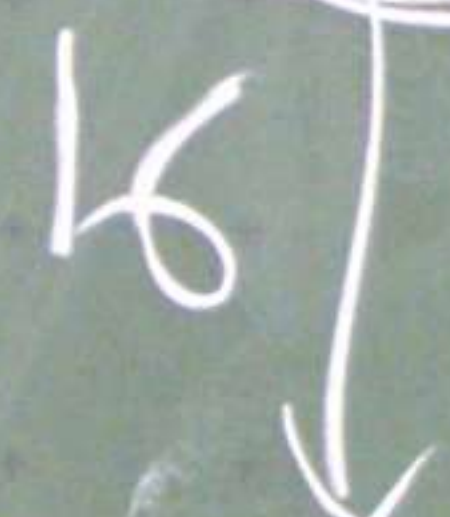
ADD, R5, 125, 125
 LW \$S1, 5(\$S2)



$$\cancel{\phi P}^{32}_{\text{NEW}} = \cancel{P}^{32}_{\text{OLD}} + 4$$

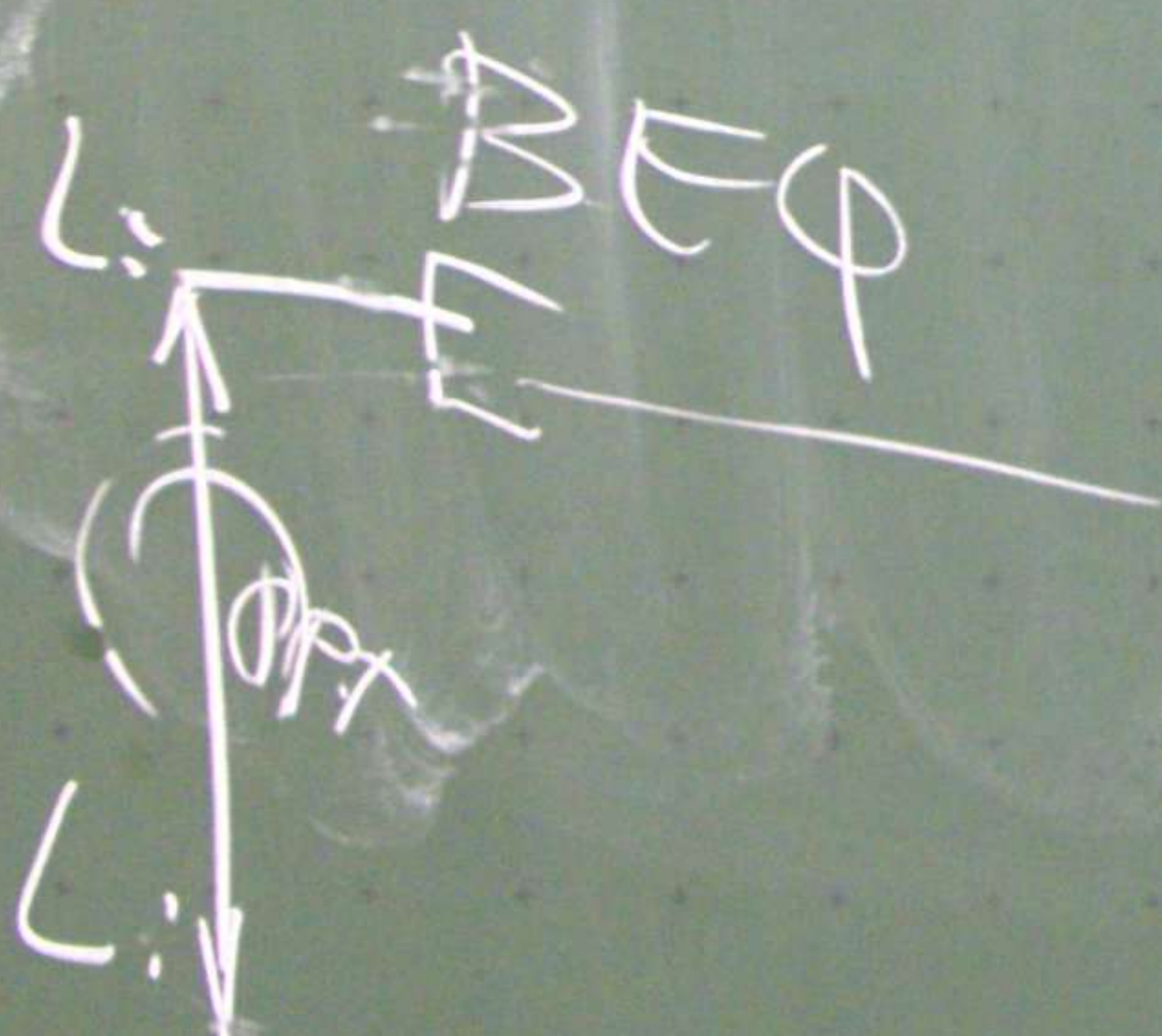
ST

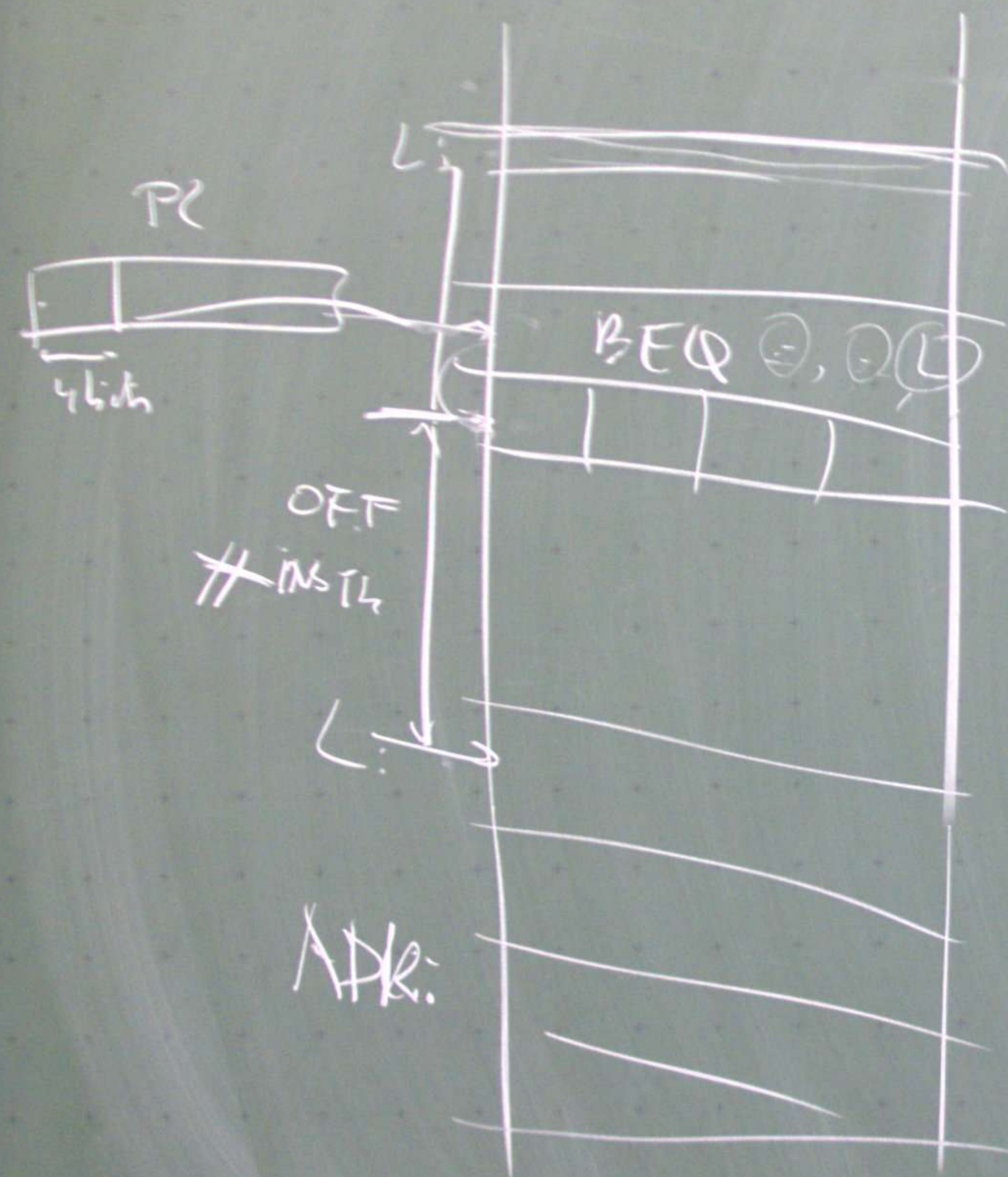
\$S1, 125, \$S2



32 bit

\$S1, \$S2, 125





INSTRUCTION MEM

32

LW $\$S1, \text{OFF}(\$S1)$
 ST $\$S1, \text{OFF}(\$S1)$
 J ADDR

opcode

000 | 00

←

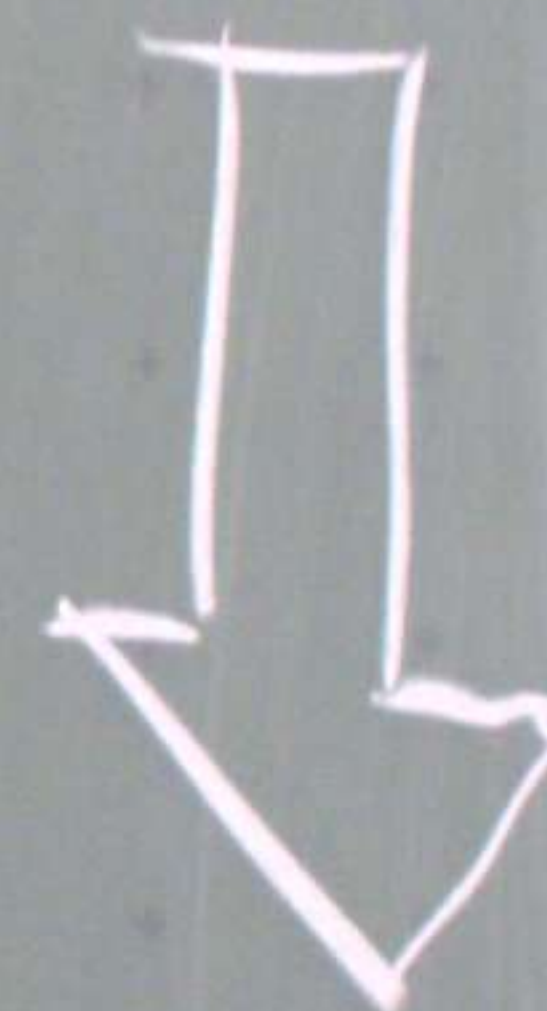
6 bits

←

26 bits

FROM PC [31..28]

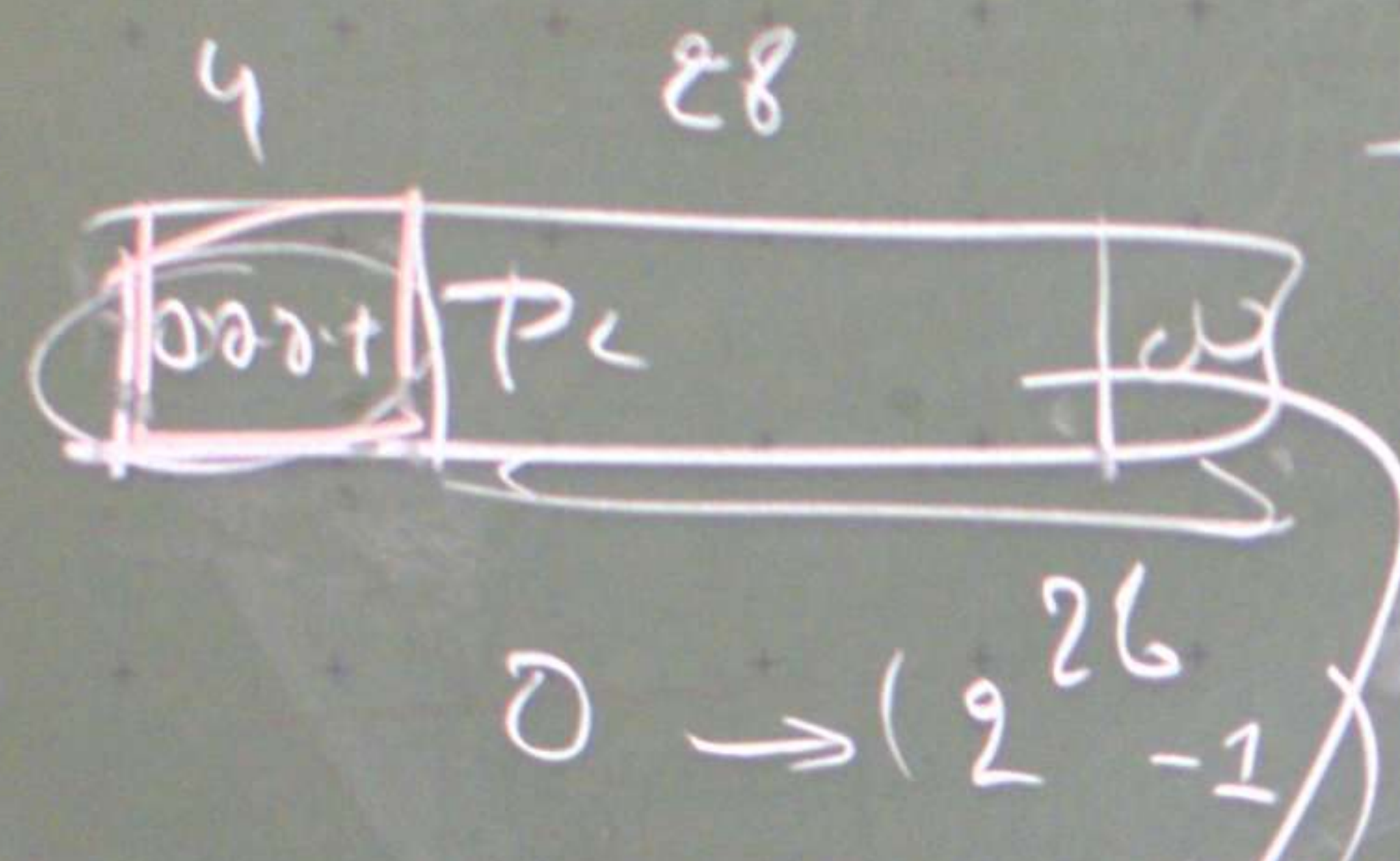
e_{26}



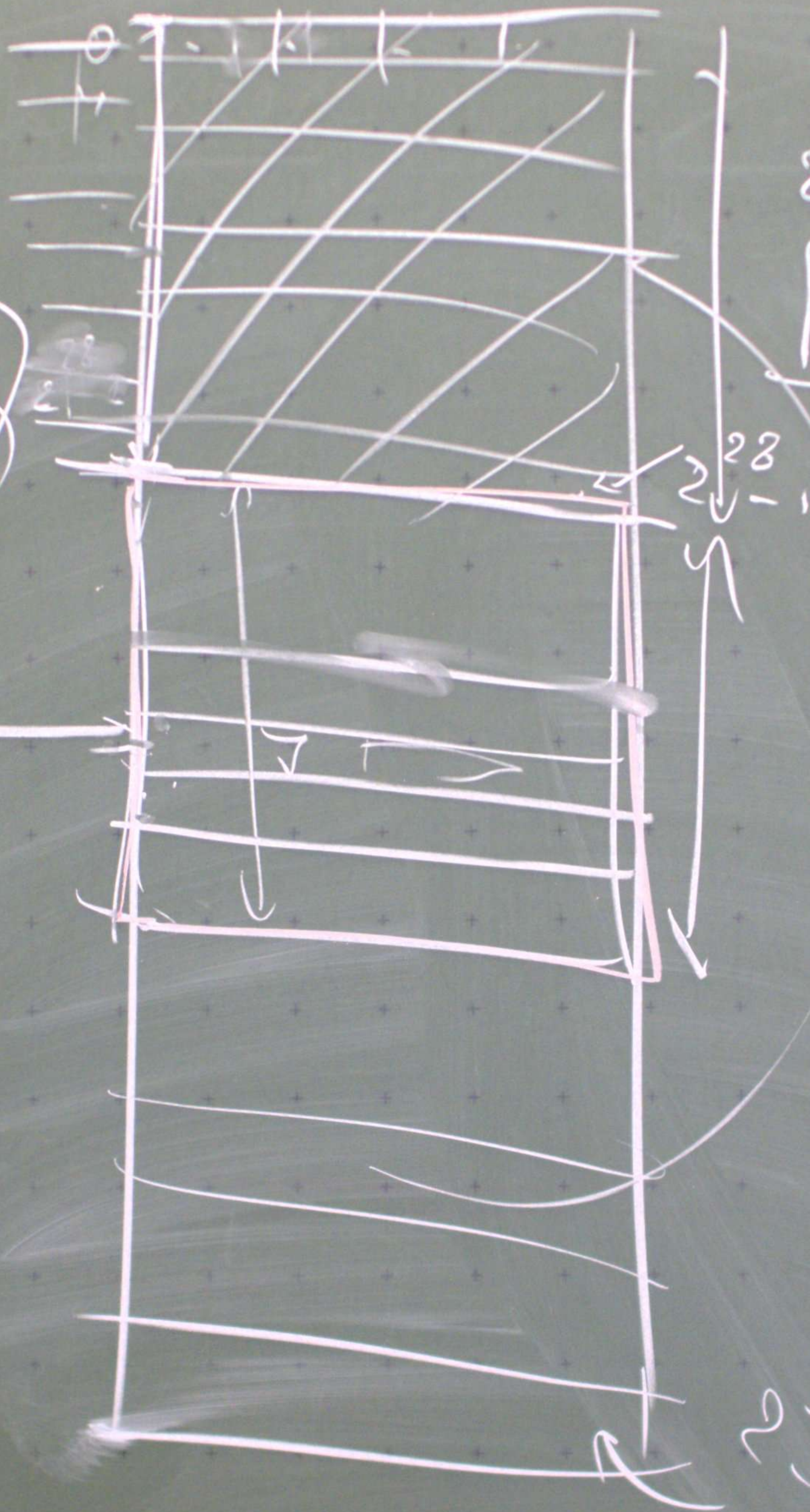
4 bits

00

J L



MEM
BLOCKS
24



28
2 28
By 747

26
2 26
INSTIR

22
2 22
INSTIR

25
2 25

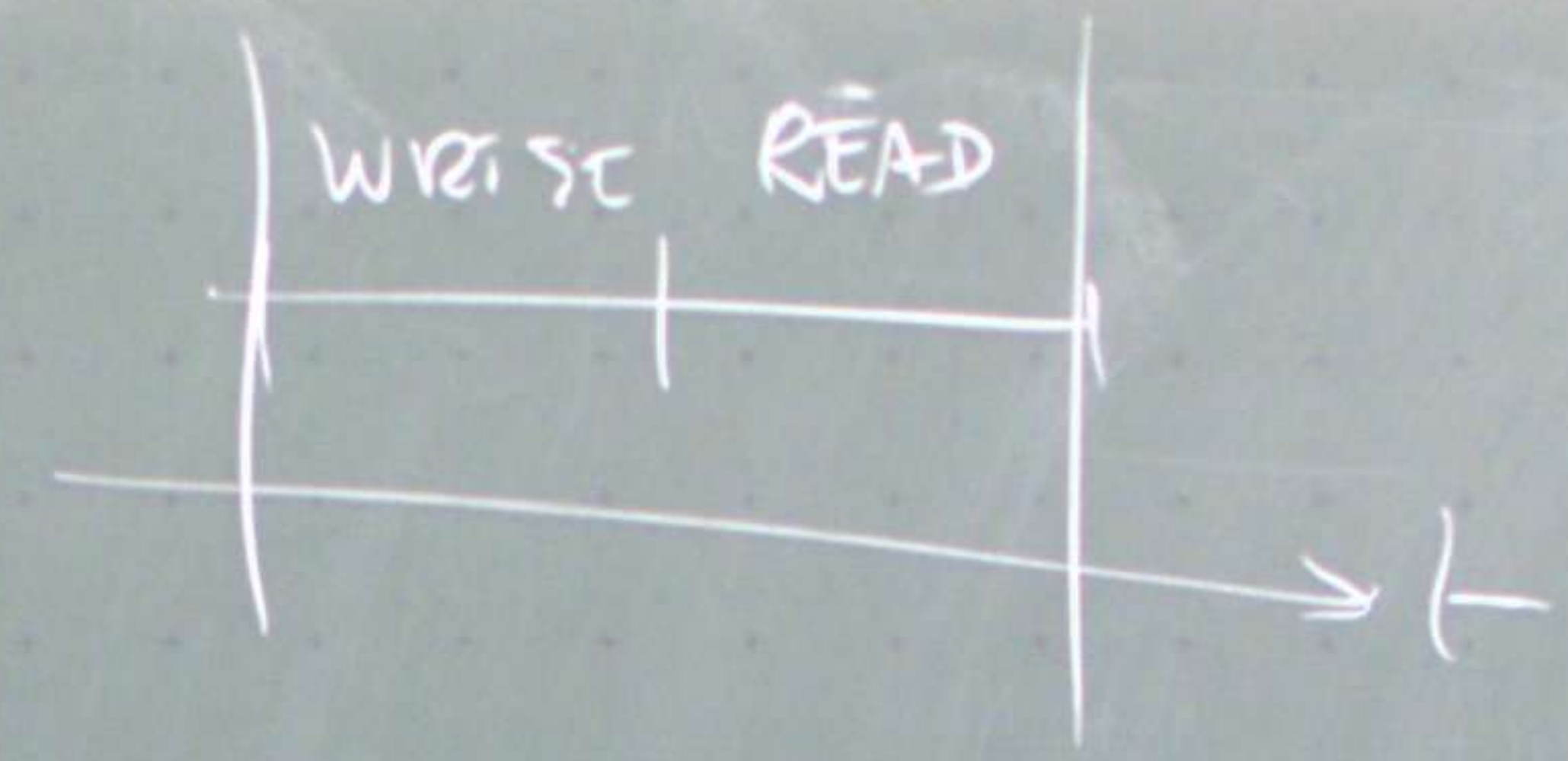
BEQ. ~ PC
JR \$S1 -

INDIRECT
ADDRESSING

LW
ST

INSTN MEM

DATA MEM



SWITCH (VAR)

CASE 0

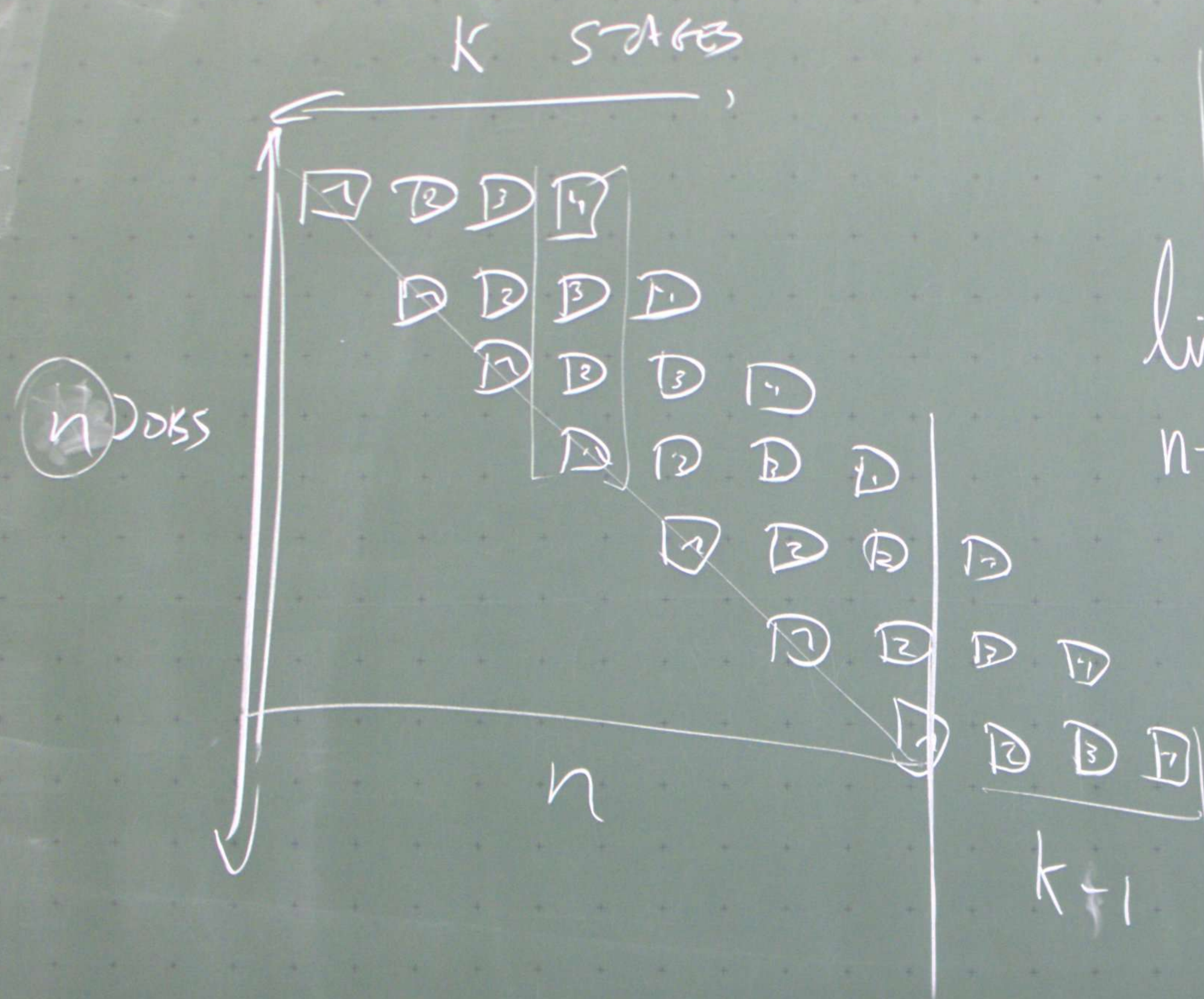
CASE 1

CASE 2

PIPELINING (ILP)

- LATENCY $\equiv$ $\times 1$

- THROUGHPUT $\nearrow$ $\frac{\times K}{S}$



$$\lim_{n \rightarrow \infty} \frac{\cancel{\frac{n}{n}} K \cancel{\phi}}{\left(\underbrace{\frac{n}{n}}_1 + \underbrace{\frac{K-1}{n}}_{\phi} \right) \cancel{\phi}} = \underline{K}$$

MEMORY

(R/W)



FRAGMENTATION



STACK

MEMORY LEAK

FUNCTIONS

PARAMS

WHERE CALLEE CAN FIND THEM

- REGISTERS $\$a0, \dots, \$a3$
- DATA MEM (STACK) $\$SP$
- DATA MEM (GLOBAL) $\$GP$

16 bit 2's complement

ST $\$t0, \text{OFFSET} (\$SP)$

LW $\$t0, \text{OFFSET} (\$GP)$

RETURN VALUE(S)

- REGISTERS $\$V0, \$V1$
- DATA MEM (STACK)
- DATA MEM (GLOBAL)

$$-2^{15} \leq \text{OFFSET} \leq 2^{15}-1$$

SYS CALL

PARAMS

- $\$a0$

TYPE OF

- $\$V0$

UNIX: SYS V

SYSCALL

PARAMS

- flag, ...

TYPE OF SERVICE (OS)

- SVX

LINUX

BSD

MACH

OS X

MEMORY (R/W)

- REGISTER

- DATA

- STACK

- HEAP

- STATIC

- PROGRAM

- TEXT

- MEMORY MAPPED I/O

free



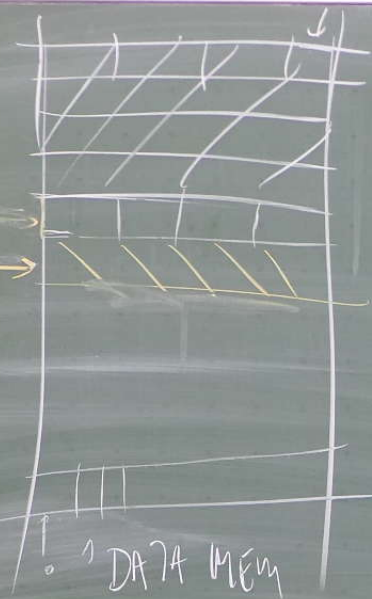
ATION

\$t4 ×

\$SP ←

MTU

CTION"



STACK



PUSH \$t4

ST \$t4, 0(\$SP)

~~ADDIU \$SP, \$SP, -4~~

SUBIU \$SP, \$SP, 4

FIRST FREE SPACE IN STACK
(\$SP)

OFFSET: - 2¹⁵

→ 2¹⁵ - 1

ST \$t4, -4(\$SP)

SUBIU \$SP, \$SP, 4

ST \$t4, 0(\$SP)

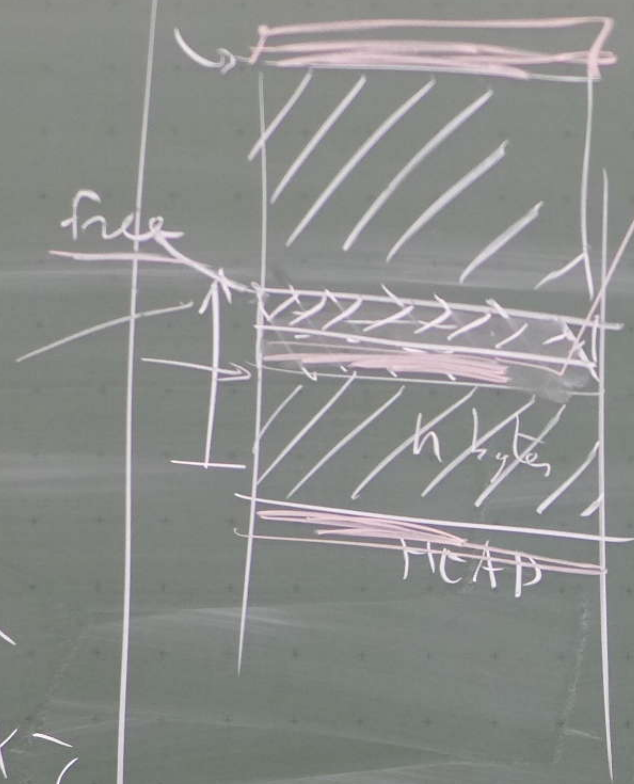
(\$SP)
LAST OCCUPIED
SPACE ON STACK

R/W)

- STACK
- HEAP
- STATIC

TEXT

ED I/O



FRAGMENTATION

TOS

malloc(n)

\$L-7 [X]

\$SP []

MMU

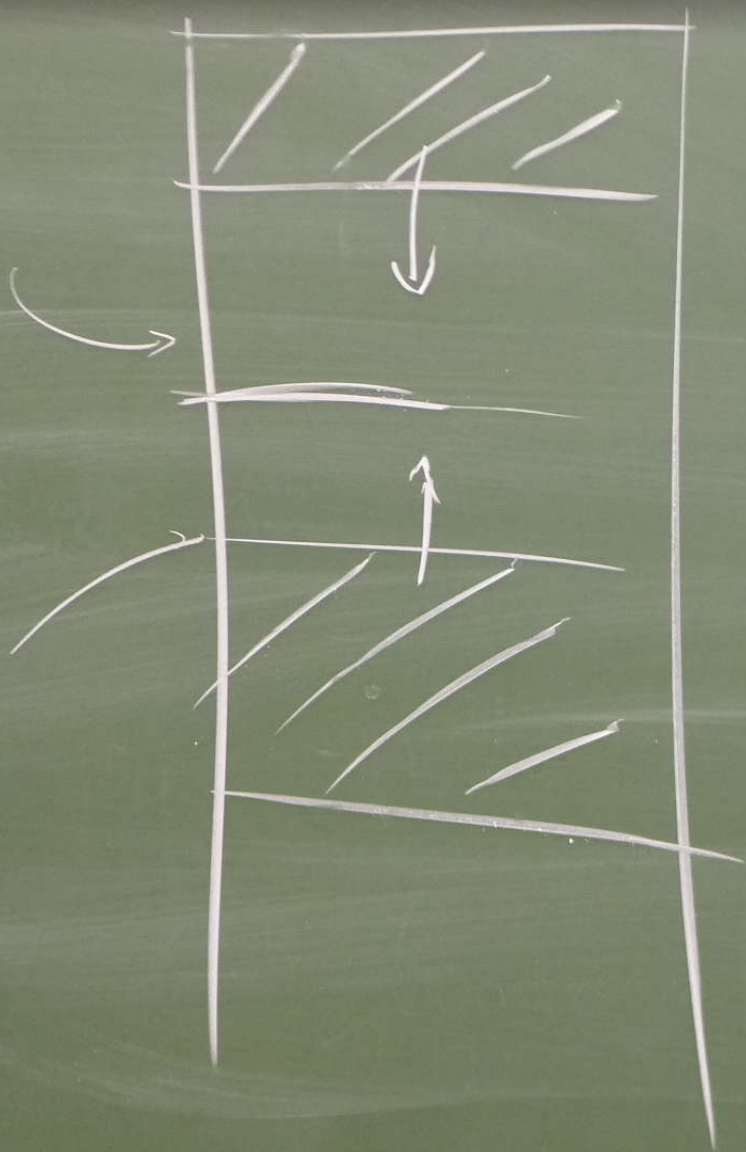
JAVA / PYTHON

~~malloc, free~~

"GARBAGE COLLECTION"



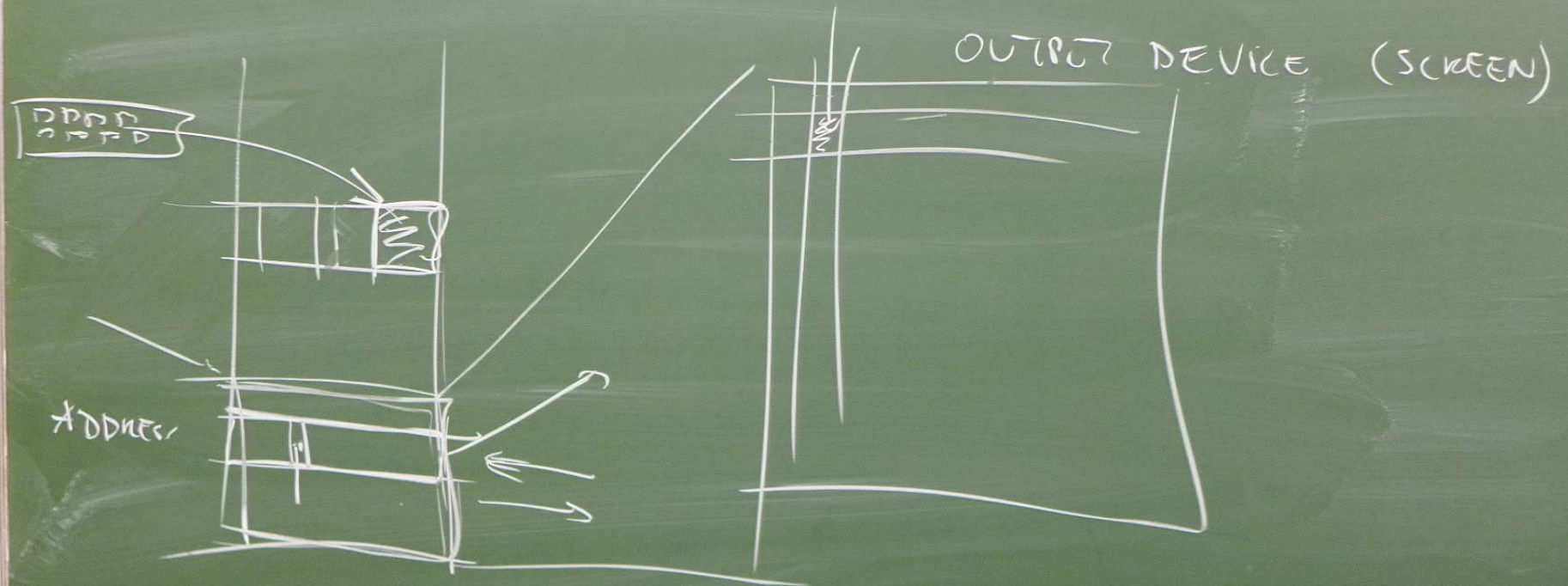
DATA M

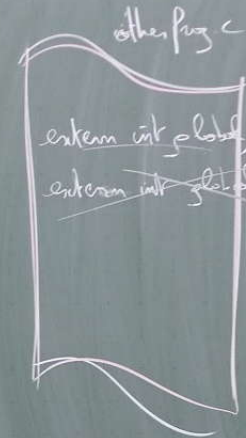
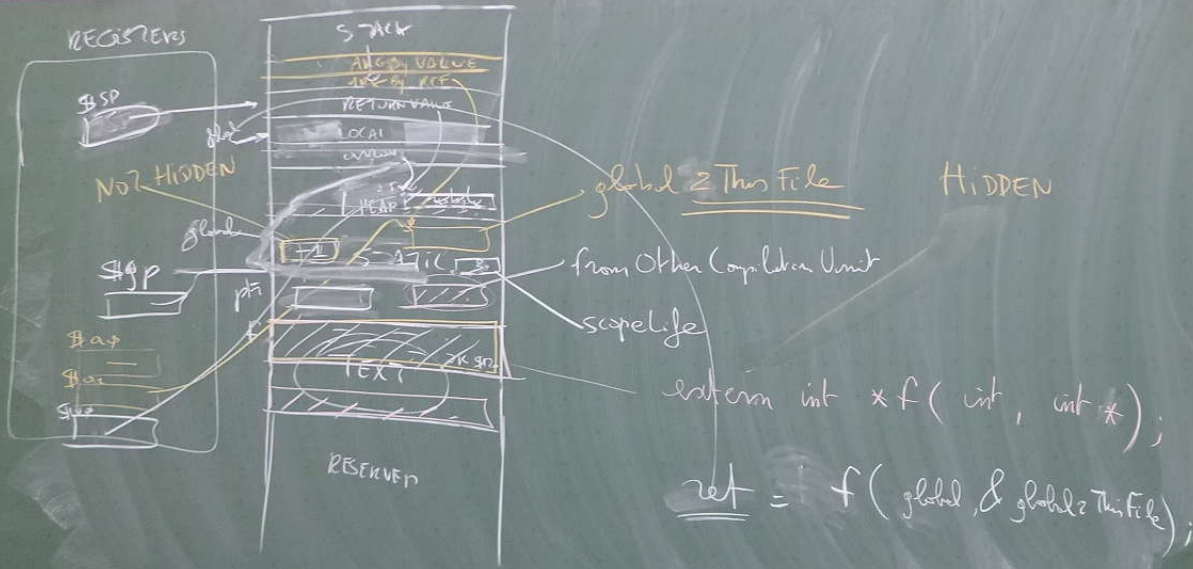


STACK-INTENSIVE
(RECURSIVE)
HEAP-INTENSIVE

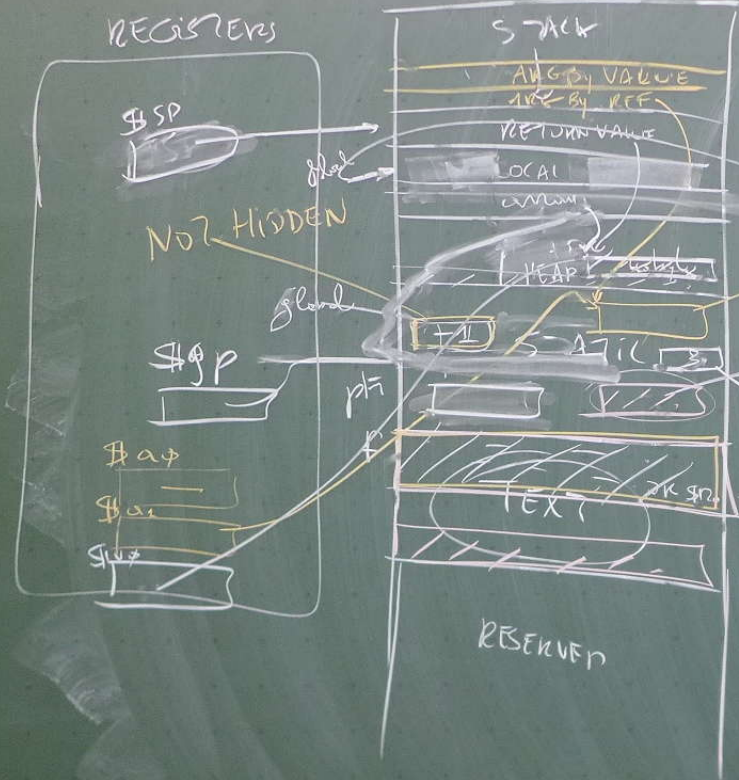
RESERVED:

eg, MEMORY MAPPED I/O





**namespace
pollution**



extern int * f(int, int *);

ret = f(global, & global 2 This File);

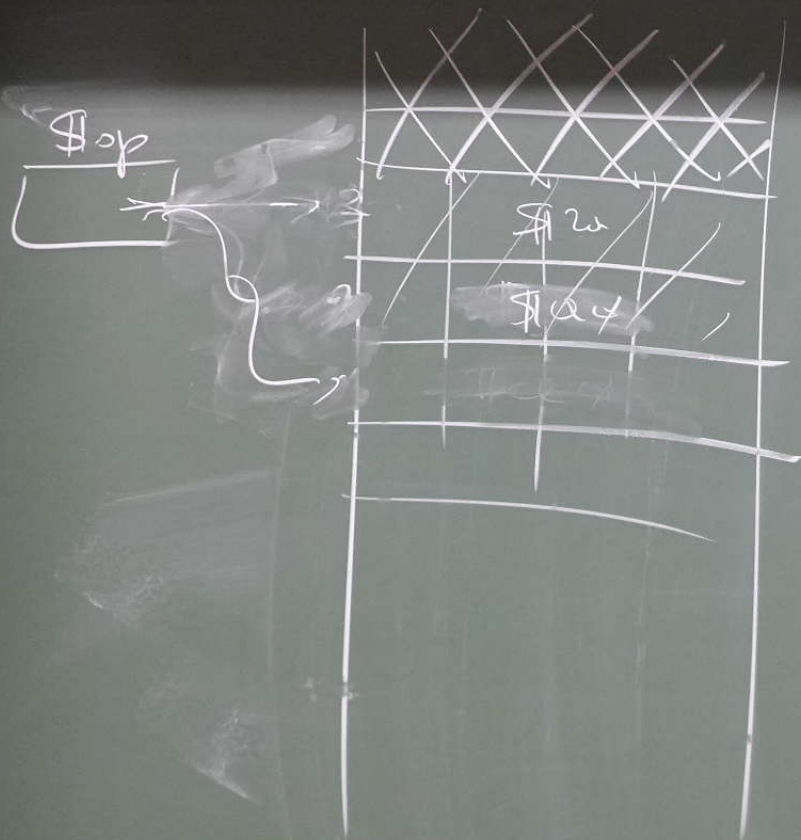
other prog. c

extern int global;

~~extern int global~~ 2 This File;

NAMESPACE

POLLUTION



SUBIU $\$SP, \$sp, 8$

SW $\$ra, 8 (\$sp)$

SW $\$a4, 4 (\$sp)$

~~SW $\$a5, 4 (\$sp)$~~

PER FCT

STACK

FACT

N VALUES

TO SAVE

SIZE

(n)

COST
INSTR

SEPARATE

PUSH
POP

COST
INSTR

EFFICIENT

$2(n+1)$

bytes

$\times 4$

SAVE
RES ZONE

$$2 \times (2 \times N)$$

$$2 \times (N + 1)$$

fact(2)

fact(1)

fact(0)

for (index = 0; index < 10; index++)

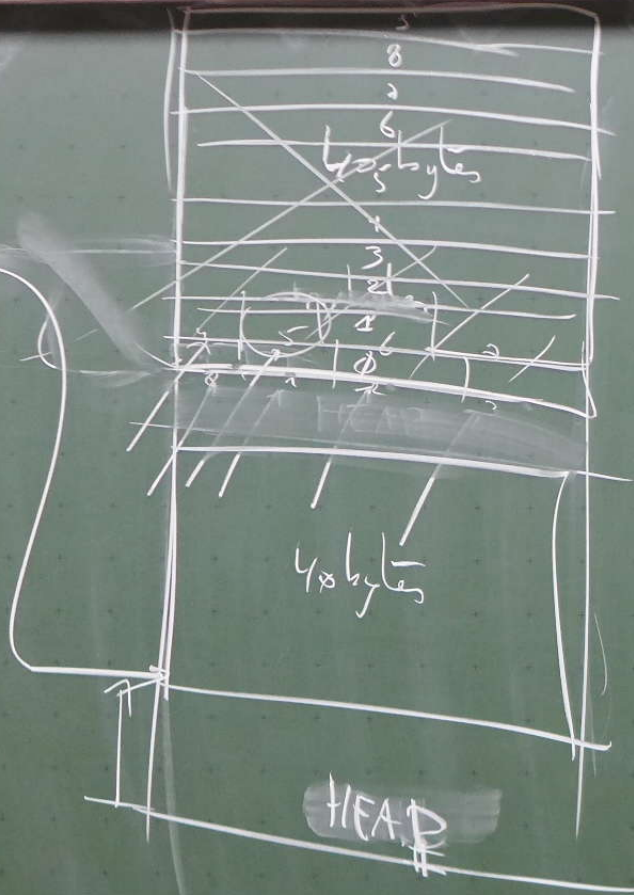
if (~~found~~)
break;

printf("%d", index);

(int)

}

ten Ints Ref

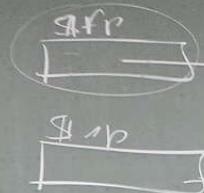


ME many
LEAK

ten Ints Ref [5]

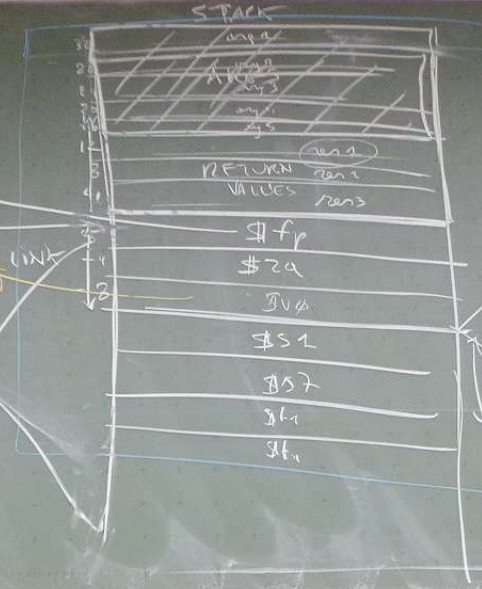
$$* (\text{ten Ints Ref} + 5 * \text{sizeof}(\text{int}))$$

LW $t_1, 32(\$fp)$
 ST $t_2, 12(\$fp)$



STATIC LINK CHAIN

DYNAMIC LINK CHAIN

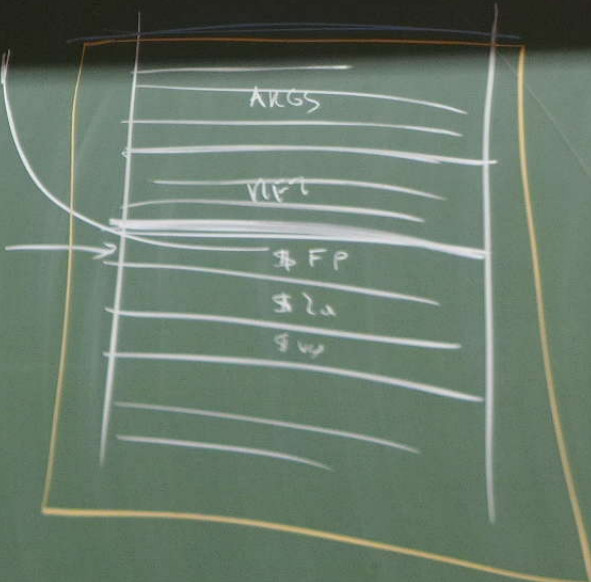


ACTIVATION RECORD
STACK FRAME

FUNCTION CALL
LOCAL "DYNAMIC" VARIABLES

SAVED REG

Problem with
 - # ARGS > 4
 - SIZE
 Solution
 ① PASS v
 OK
 ② PUSH



PROBLEM WITH $\$u4 - \$a3, \$u4, \$u1$

- # ARGS, # RET VALUES
 > 4 > 2

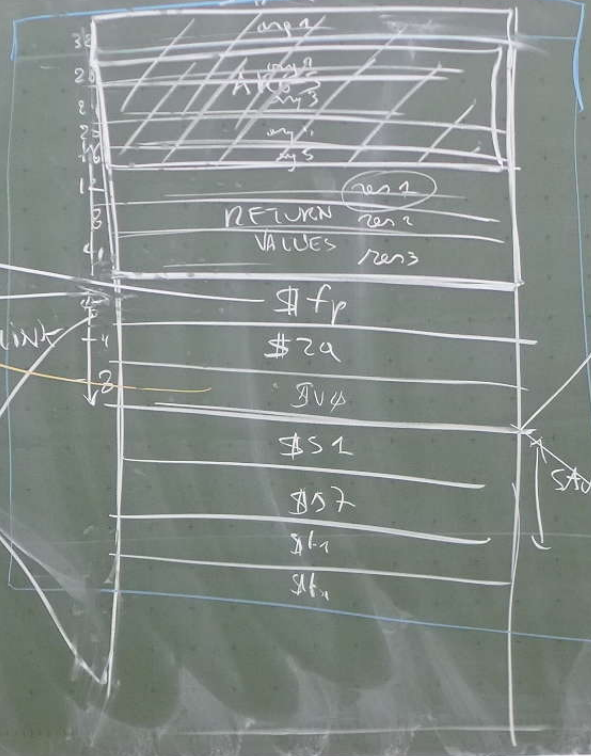
- SIZE OF ARGS, RET VALUES > 4 bytes

SOLUTION

- ① PASS REFERENCE TO STRUCTURE
 OK FOR LEAF, NOT FOR NON-LEAF
- ② PUSH ON STACK

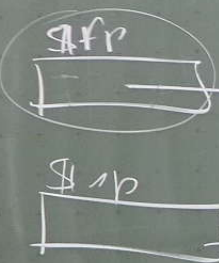
HEAP

STACK



ACTIVATION RECORD STACK FRAME

FUNCTION
CALL
LOCAL "DYNAMIC"
VARIABLES

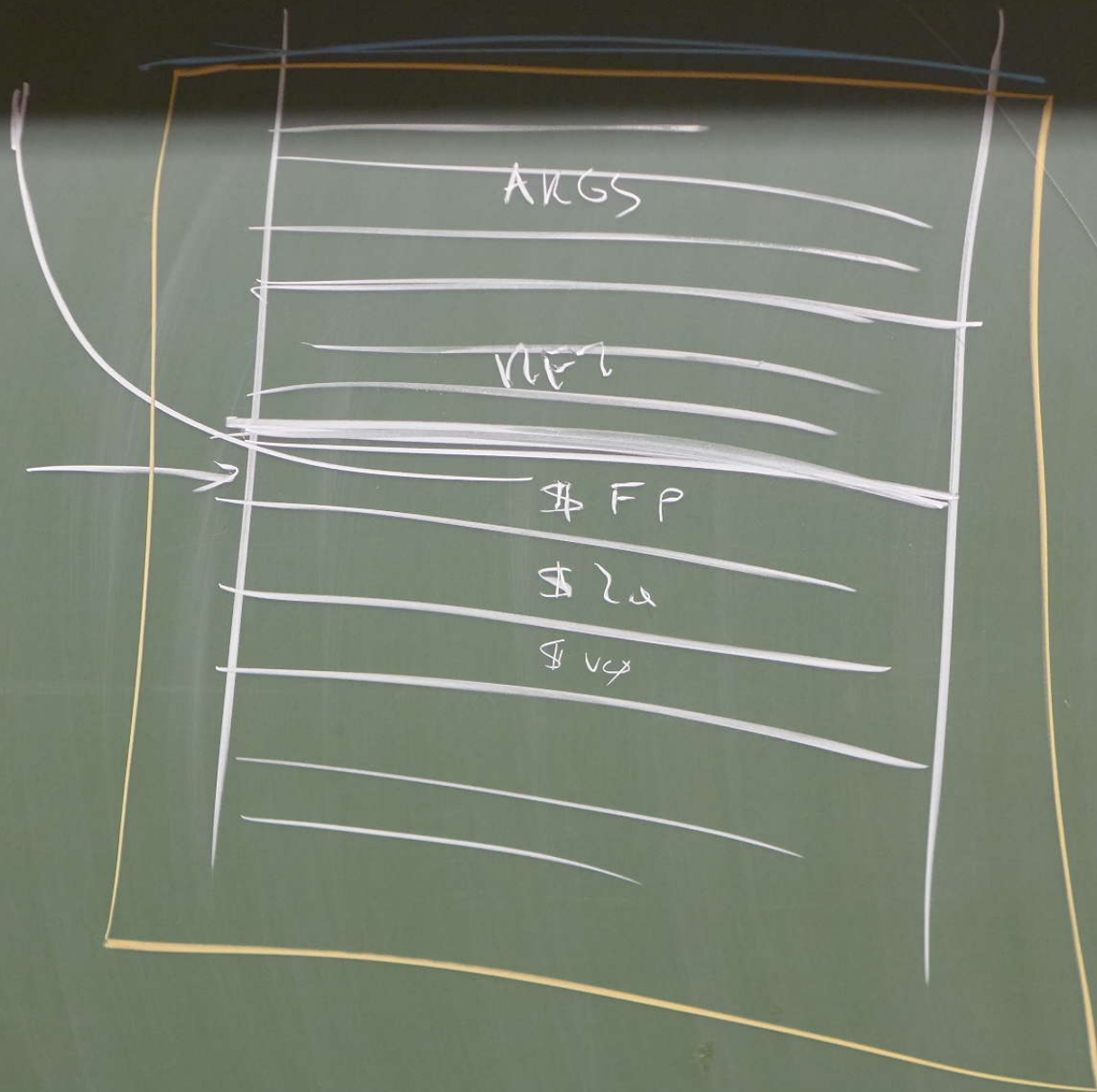


DYNAMIC LINK

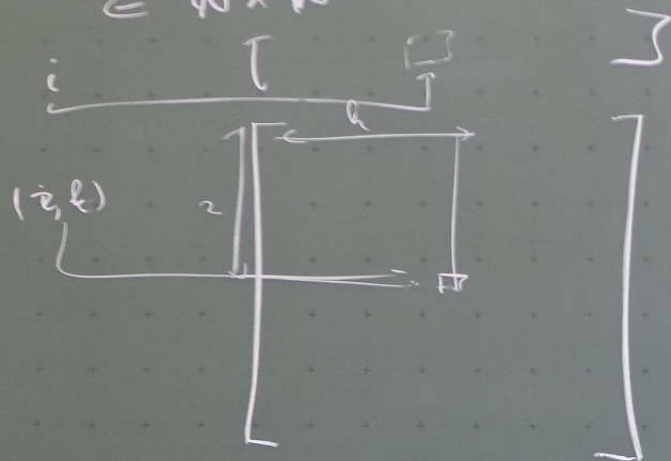
STATIC LINK

SAVED REG

LW \$t1, 32(\$fp)
ST \$t2, 12(\$fp)



vector $\in N$
 $\in N \times N$



T

$i \rightarrow T$

2D

Vector
Matrix

Array

8 DIM - - -

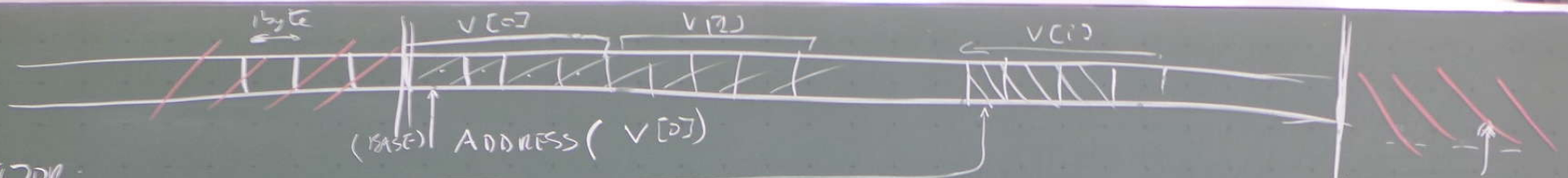
ELEMENT TYPE == $\forall$ elements
 SIZE (ELEMENT TYPE)

ADDRESS (BASE ADDRESS)

SIZE OF THE VECTOR, MATRIX
 DIMENSION
 ARRAY

5x2 MATRIX OF INT

SIZE: $10 \times \text{SIZE}(\text{ELEMENT TYPE})$
 DIMENSION: 2



VECTOR

$$\text{ADDRESS}(v[i]) = \text{ADDRESS}(v[0]) + i \times \text{SIZE}(\text{ELEMENT TYPE})$$

int $v[1]$;

int * v ;

$$v = 10 \times \text{module}(\text{size of (int)})$$

int $v[10]$;

$$\text{SIZE}(v) = \text{LENGTH} \times \text{SIZE}(\text{ELEMENT TYPE})$$

MATRIX

ARRAY

2D

ADDRESS($a[r, k]$)

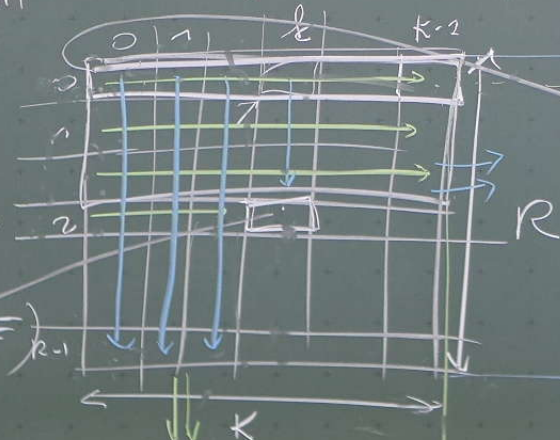
ROWS = R

COLUMNS = K

SIZE(ELEMENT TYPE)

$$0 \leq i < \text{LENGTH}$$

JAVA
C



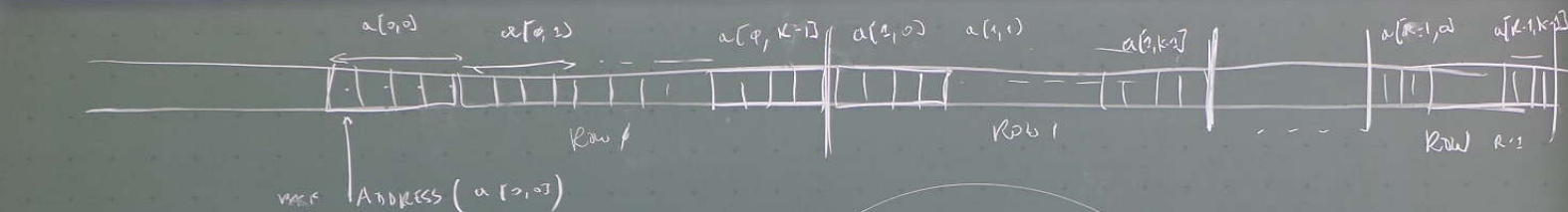
(BASE) ADDRESS($a[0, 0]$)

$$\text{SIZE}(a) = R \times K \times \text{SIZE}(\text{ELEMENT TYPE})$$

INT
(ELEMENT TYPE)

MATRIX 2D
ARRAY
 ADDRESS ($a[r, k]$)
 # ROWS = R
 # COLUMNS = K
 SIZE (ELEMENT TYPE)

(BASE) ADDRESS ($a[0,0]$)
 $SIZE(a) = R \times K \times SIZE(ELEMENT TYPE)$

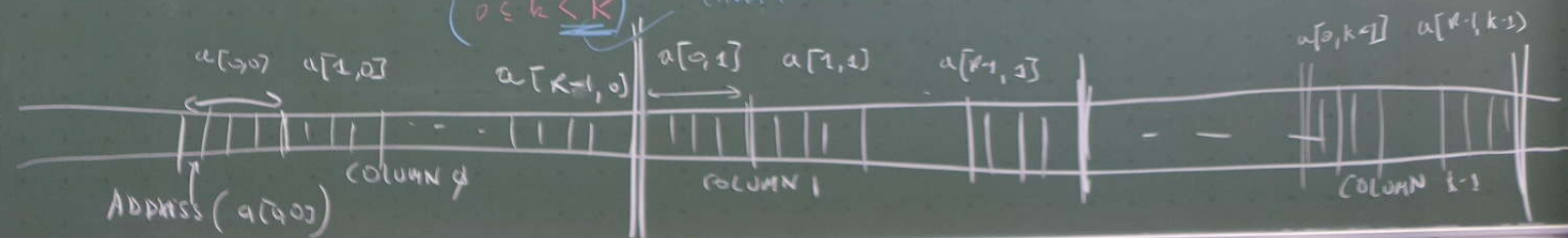


ROW-MAJOR (C)
 ADDRESS ($a[r, k]$) = ADDRESS ($a[0,0]$) + $(r + 2 \times k) \times SIZE(ELEMENT TYPE)$
 $(0 \leq r < R)$
 $(0 \leq k < K)$ CAVEAT: modulo

COLUMN-MAJOR (FORTRAN)

ADDRESS ($a[r, k]$) = ADDRESS ($a[0,0]$) + $(2 + k \times R) \times SIZE(ELEMENT TYPE)$
 $(0 \leq r < R)$
 $(0 \leq k < K)$ CAVEAT: modulo

$\times SIZE(ELEMENT TYPE)$

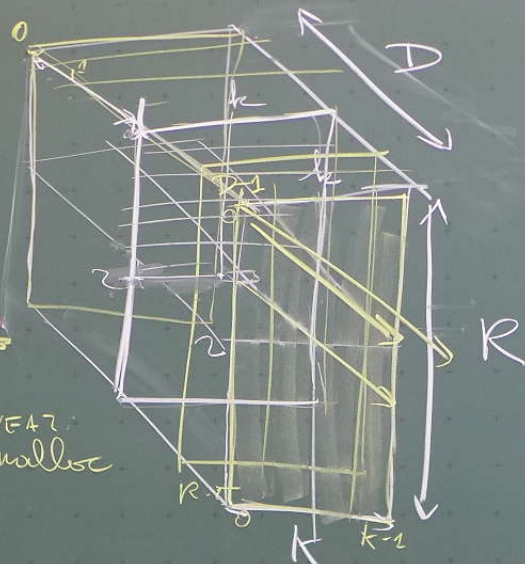


3D ARRAY

$$a[r, k, d]$$

$0 \leq r < R$
 $0 \leq k < K$
 $0 \leq d < D$

$0 \rightarrow R-1$
 $0 \rightarrow K-1$
 $0 \rightarrow D-1$

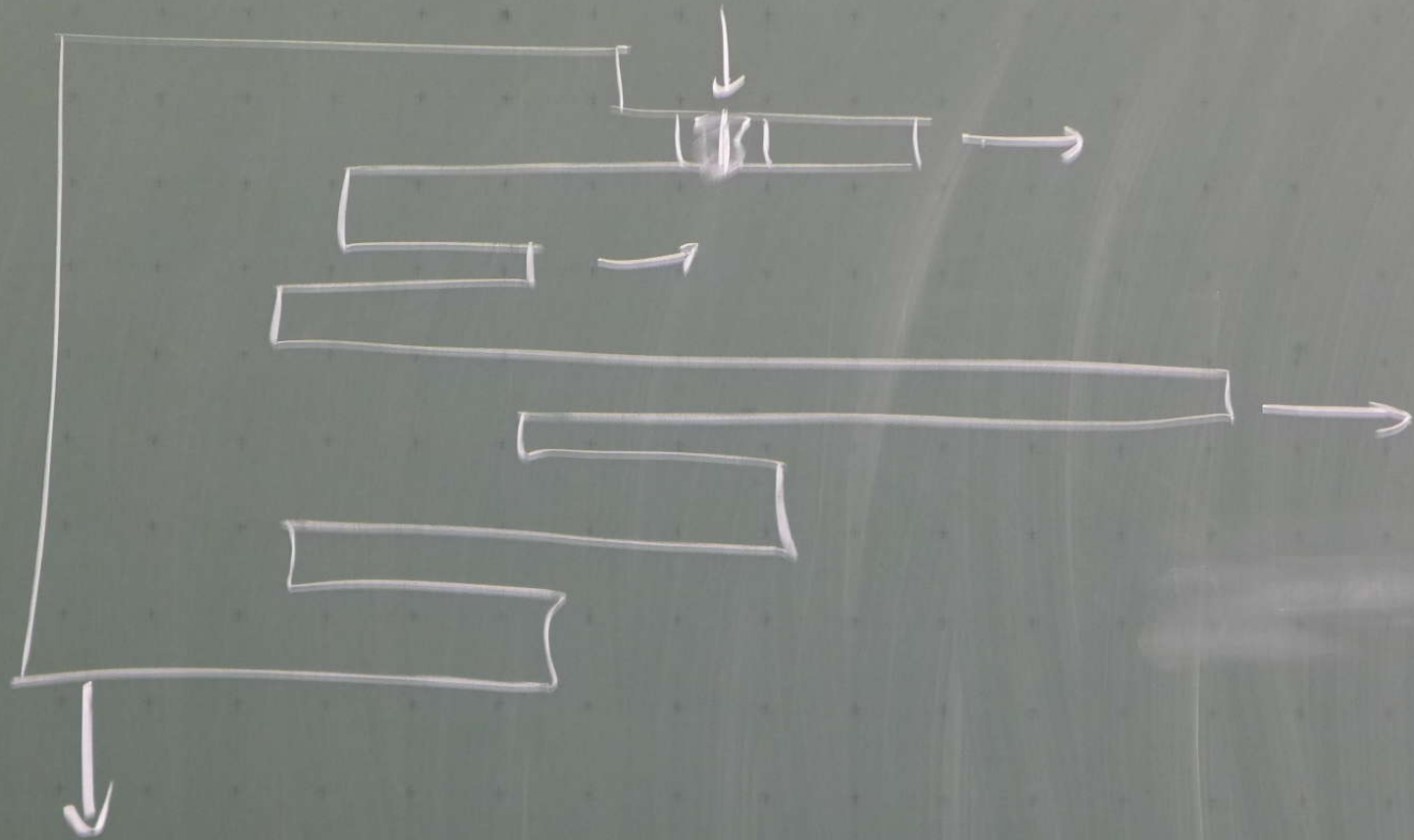


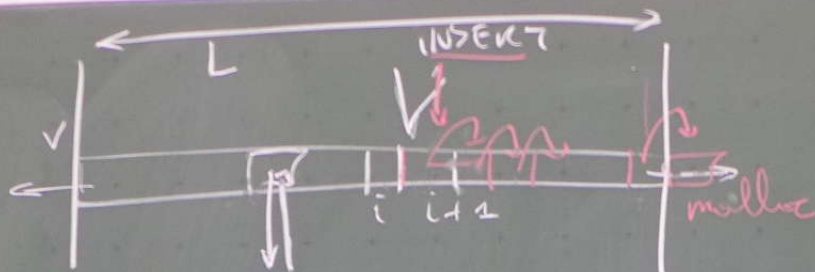
CAVEAT:
matrix

BASE ADDRESS ($a[0,0,0]$)

$$SIZE(a) = R \times K \times D \times SIZE(\text{ELEMENT TYPE})$$

$$ADDRESS(a[r, k, d]) = ADDRESS(a[0,0,0]) + (k + r \times K + d \times R \times K) \times SIZE(\text{ELEMENT TYPE})$$





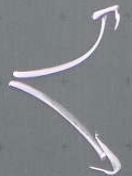
① realloc under L

② ←

③ INSERT $j \in [i+1, i+2, \dots, L-1]$

(LENGTH)
DYNAMIC SIZE
VECTOR

ALTERNATIVES:



LINKED LIST

$$SIZE(V) = L \times (SIZE(ELEMENT TYPE) + \boxed{SIZE(PTN)})$$

$$\begin{aligned} \#OP \rightarrow ELEM i &= \\ \#MEM REFS &= 2 \times i \end{aligned}$$

$$\begin{aligned} SIZE(V) &= L \times SIZE(ELEMENT TYPE) \\ &= 1 \\ &= 1 \end{aligned}$$

$$+ \boxed{SIZE(PTN)} = 4$$

.text < LA \$S4, DL1
 LA \$S5, DL2

LW DD \$S1, ~~(\$S4)~~

LW \$S2, ~~(\$S5)~~

ADD \$S3, \$S1, \$S2

ST \$S3, ~~(\$S6)~~

LA \$S6, DL3

Δf

.data

DL1 .word

DL2 .word

DL3 .word

ADD DL3, DL1, DL2

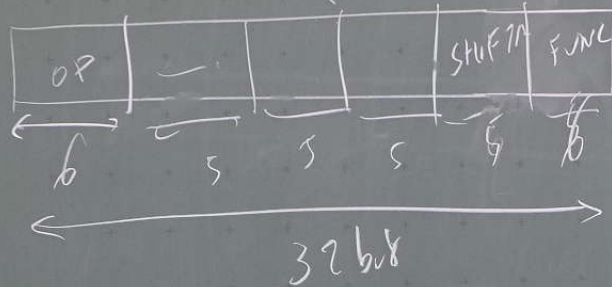
Δt

①

DURATION

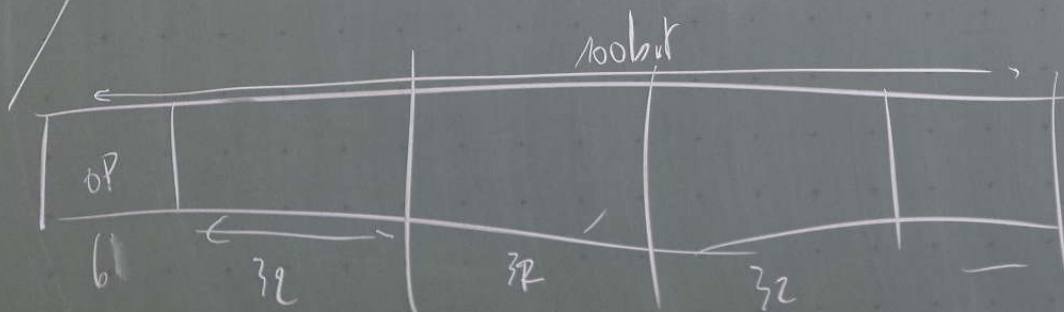
ADD \$S3, \$S2, \$S2

OF INSTRUCTIONS



②

SIZE



if (~~(a=f)~~ || ~~(c=d)~~ || ~~(a=b)~~)



jc - 03

Address (vlt) + $\textcircled{i} \times \text{SIZE (EL type)}$

INSTR

y(3)

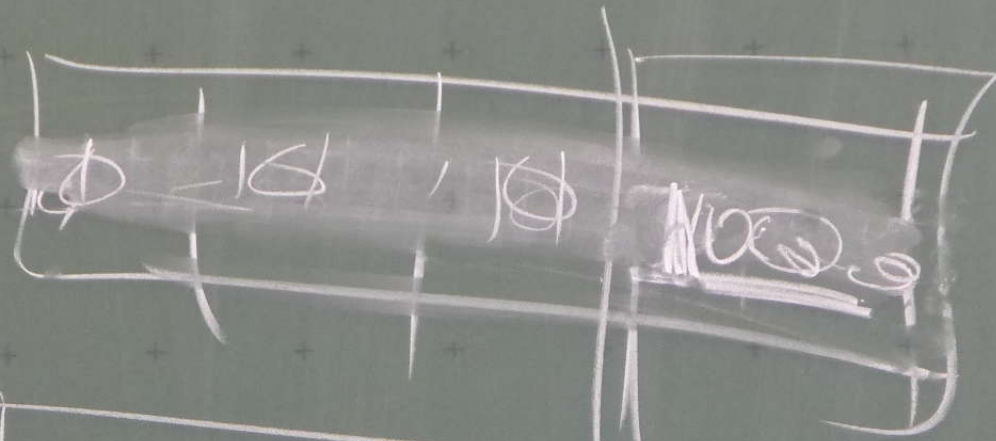
x(3)



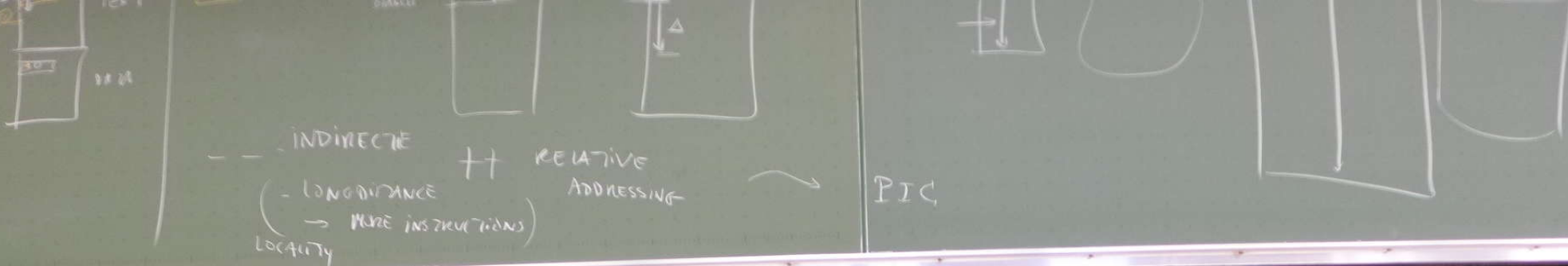
ADDRESS(y[0])

ADDRESS(x[0])

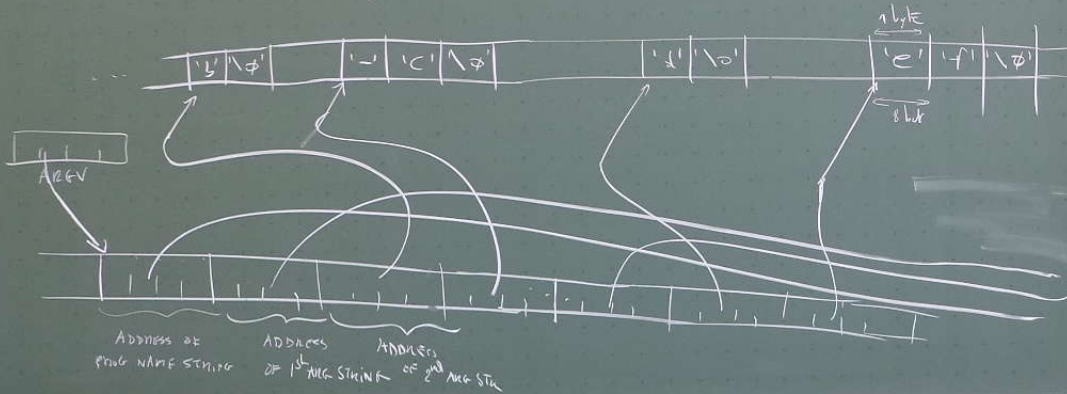
\$12



For 1000



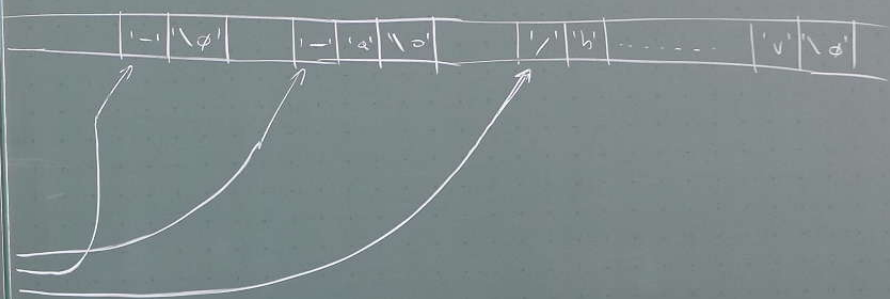
```
int main ( int argc, char ** argv )
7
```

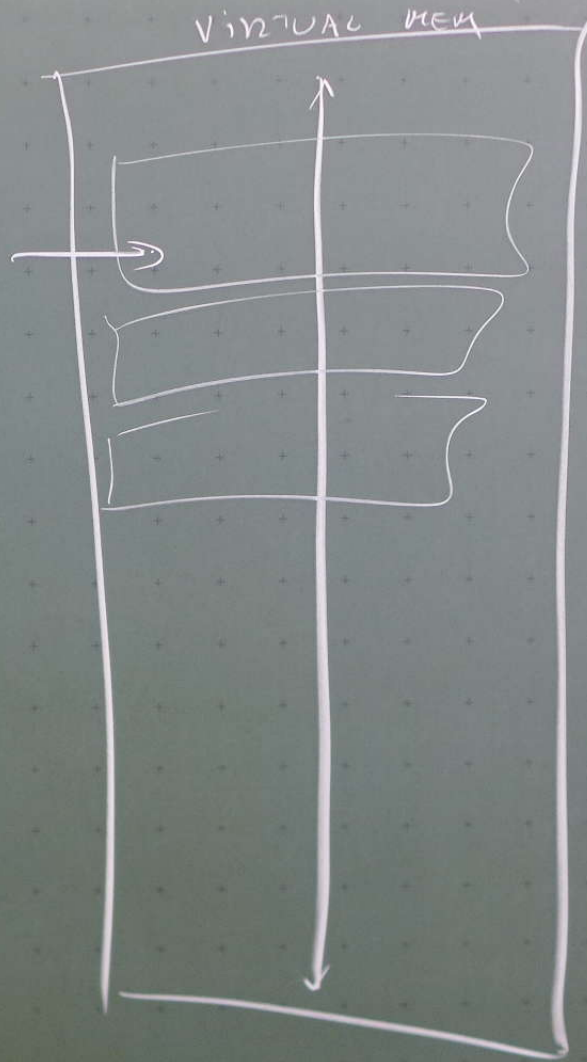
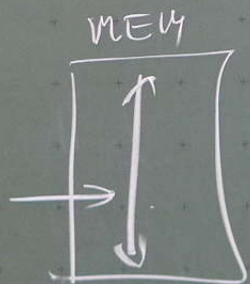


```
T *p ; p++
one f(T) p = p + 1
```

C-STYLE STRING

ZERO-TERMINATED ASCII STRING



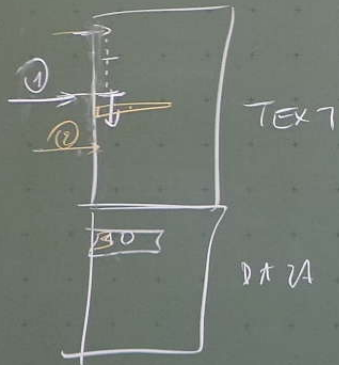


PIC

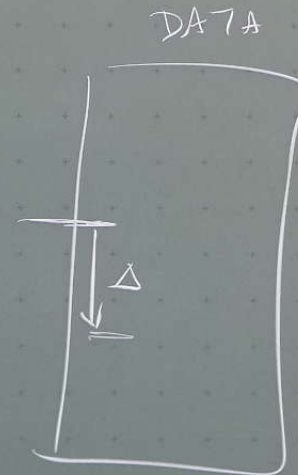
Address of
prog name string

Addresses
of 1st ARG string

Addresses
of 2nd ARG string



BRANCH



INDIRECT

++ RELATIVE ADDRESSING

(- LONG DISTANCE
→ MORE INSTRUCTIONS)
LOCALITY

PIC

LOCALITY → MORE INSTRUCTIONS

main()

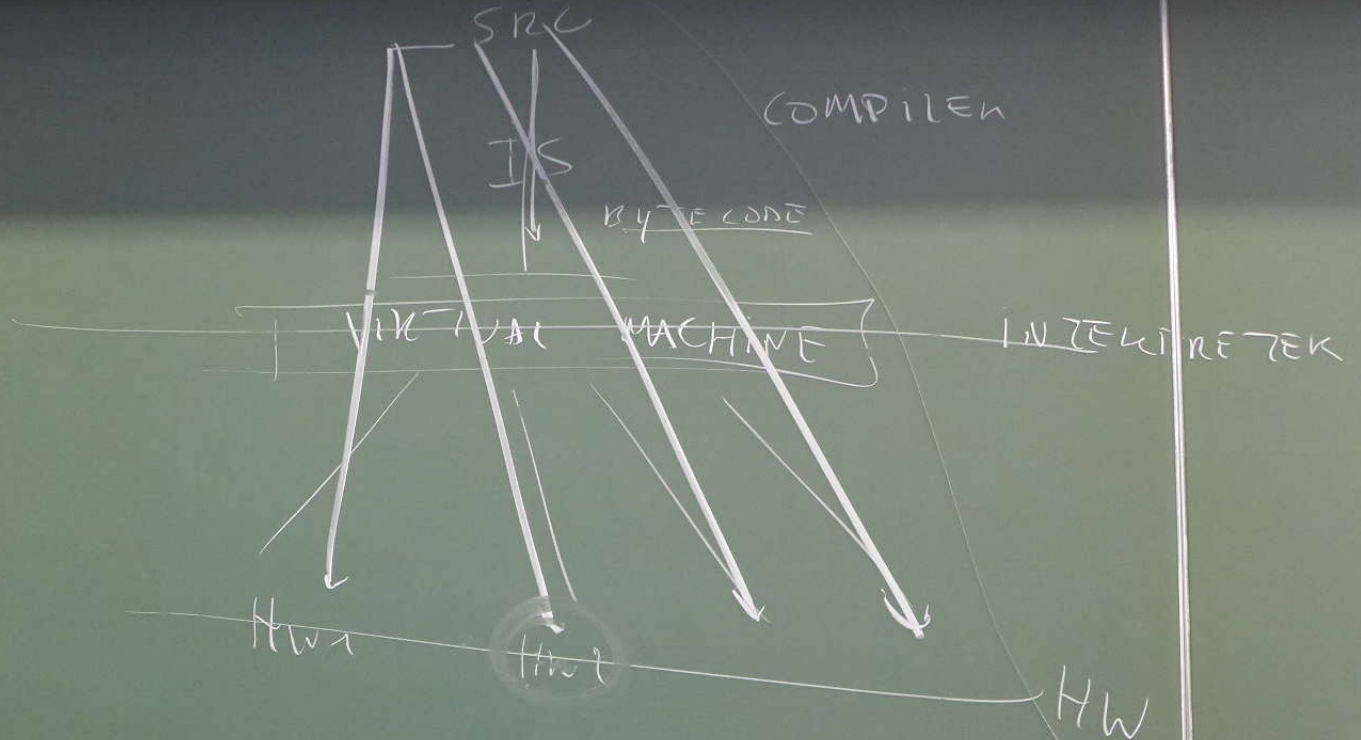
if cond

f()

else

g()

g(), g();



-5

$$- : \mathbb{Q} \rightarrow \mathbb{Q}$$

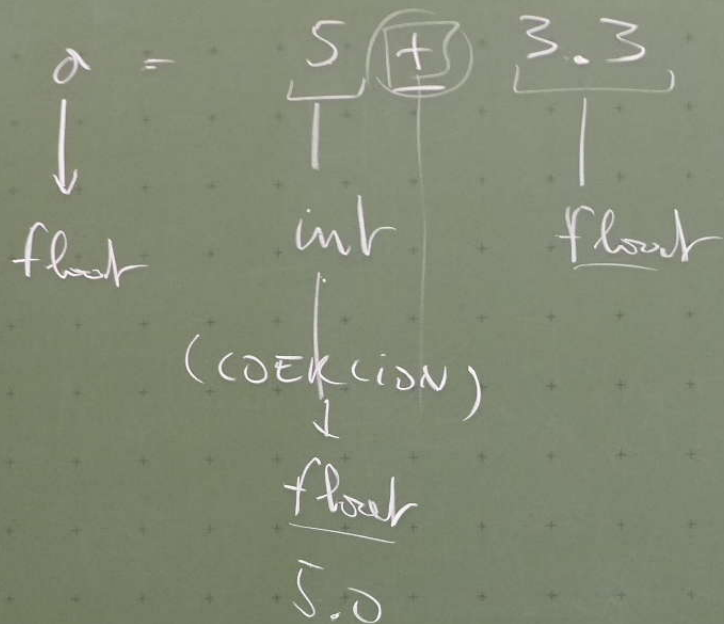
Unary Min (i)

3 - 5

$$- : \mathbb{Q} \times \mathbb{Q} \rightarrow \mathbb{Q}$$

Binary Min (i , j)





$+$: int $\times$ int $\rightarrow$ int

~~$+$~~ : float $\times$ float $\rightarrow$ float

RPN

INFIX

$$(2) + (3 \times 4)$$

PREFIX

$$+ 2 * 3 4$$

→ POSTFIX

