HW
COST

$ COST
REAL ESTATE (SPACE)
ENERGY

$\Delta t = 2 \times DT$
PROCESSING IN → OUT

WORST
CASE

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

→ 123,456,912

→ 123,456,912

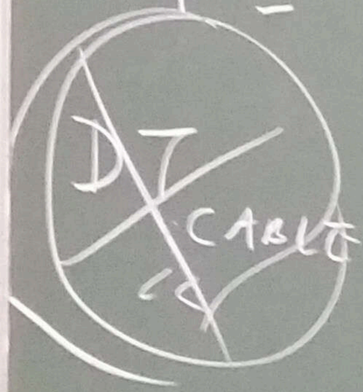
SUM of 1

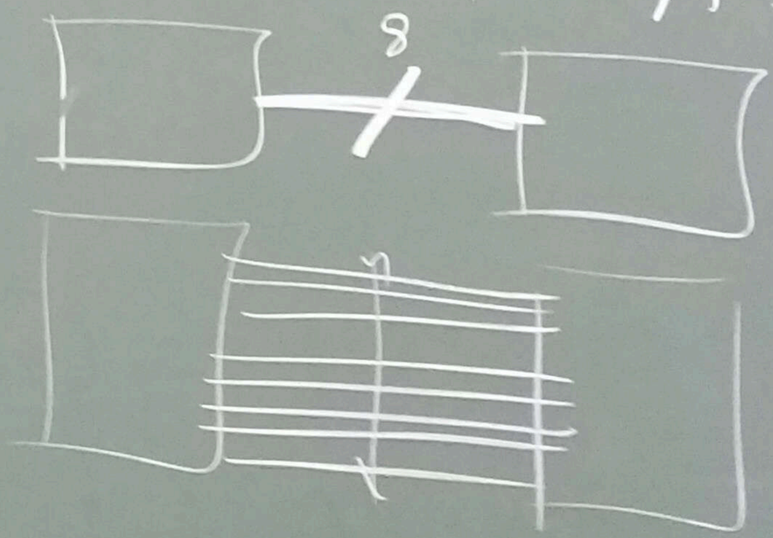
$D = (\overline{A} \cdot B \cdot C) + (A \cdot \overline{B} \cdot \overline{C}) + \ldots$

PRODUCTS

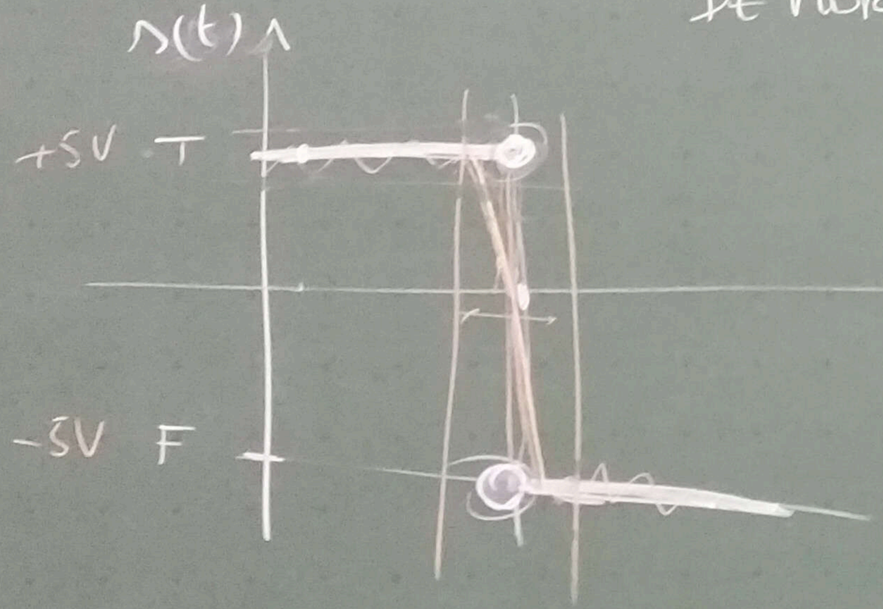
$E = \overline{A} \cdot B \cdot C + A \cdot \overline{B} \cdot C + A \cdot B \cdot \overline{C}$

$F = A \cdot B \cdot C$





"BUS"



DE MORGAN

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

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

$res = sqr(2)$ where

$res = 2$

$f_t : \underline{\mathbb{N}} \longrightarrow \mathbb{N}$

$f_p : \mathbb{N} \longrightarrow \mathbb{N} \cup \{$ Don't know $\}$

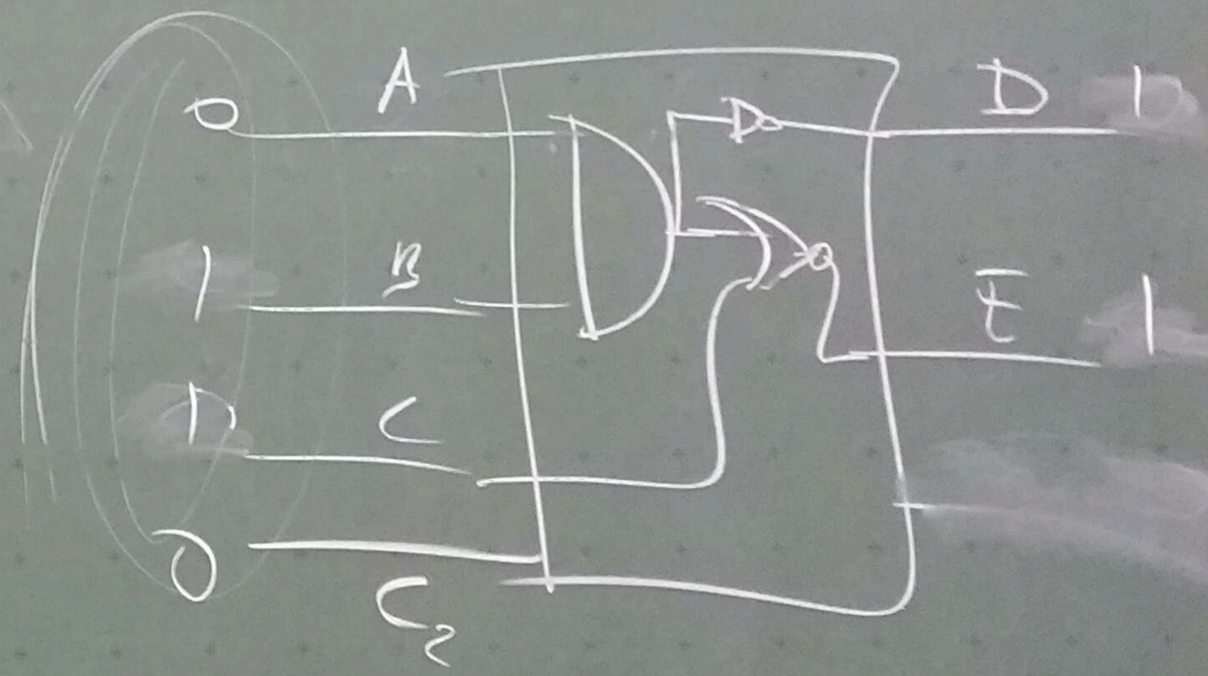
MATLAB

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

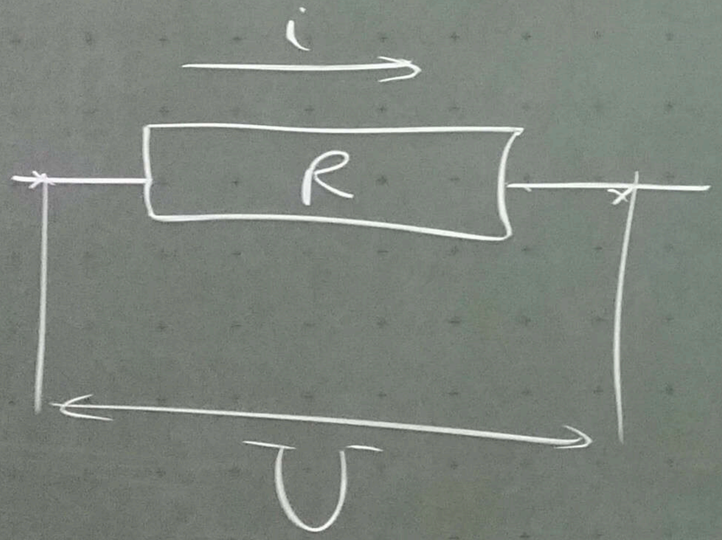
$!=$

$\sim$ $\neg$ $\vee$ $\wedge$ $\vee$

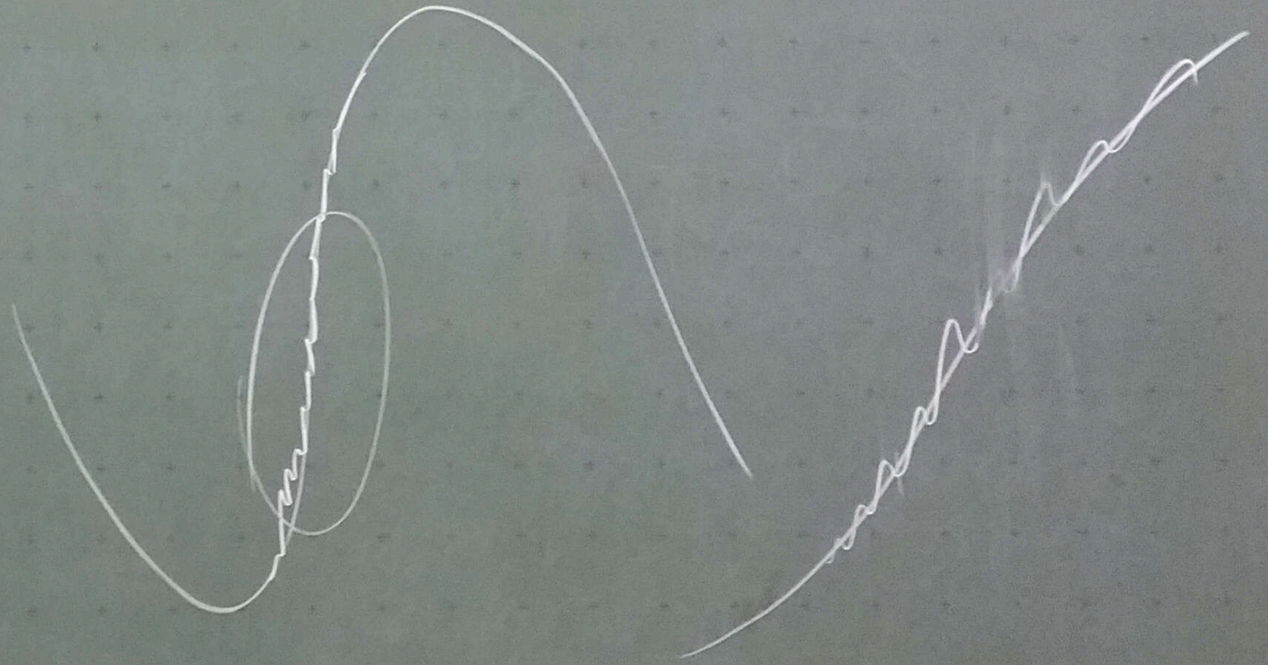
1	2
2	4
3	8
4	16
5	32
N	2^N



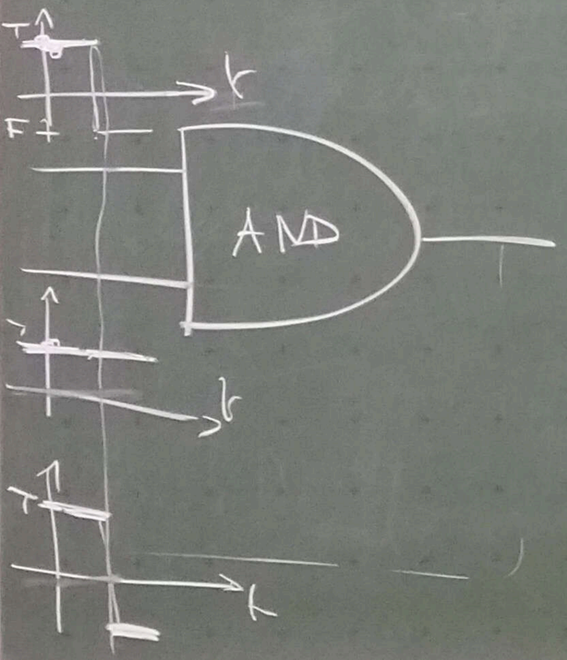
$U = R_{(T)} \cdot i$

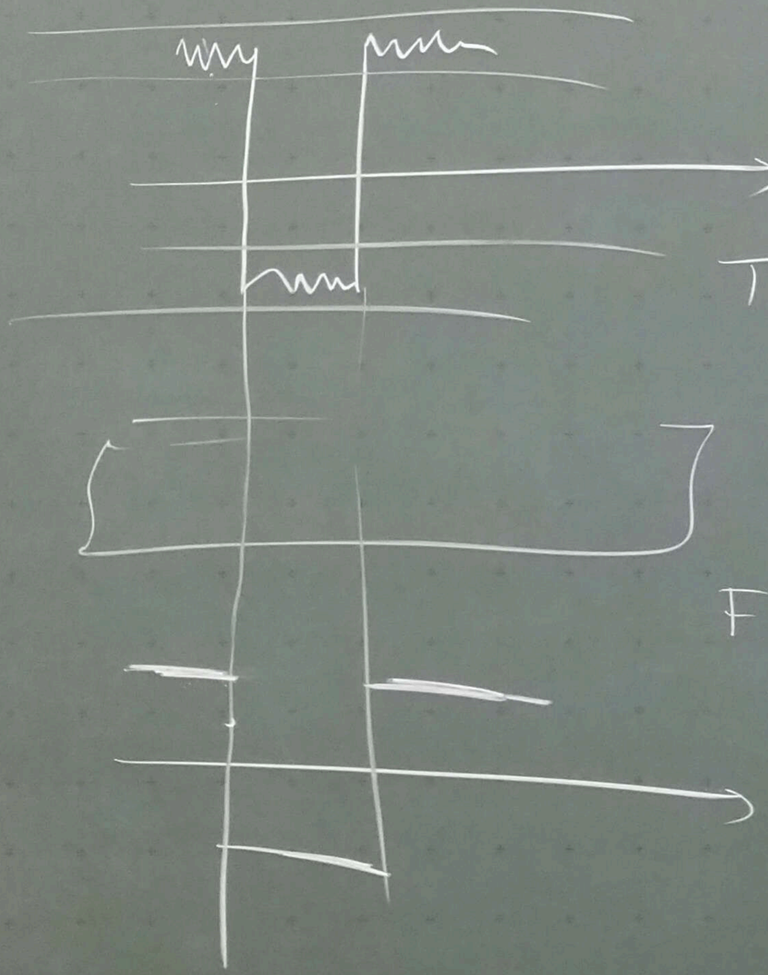


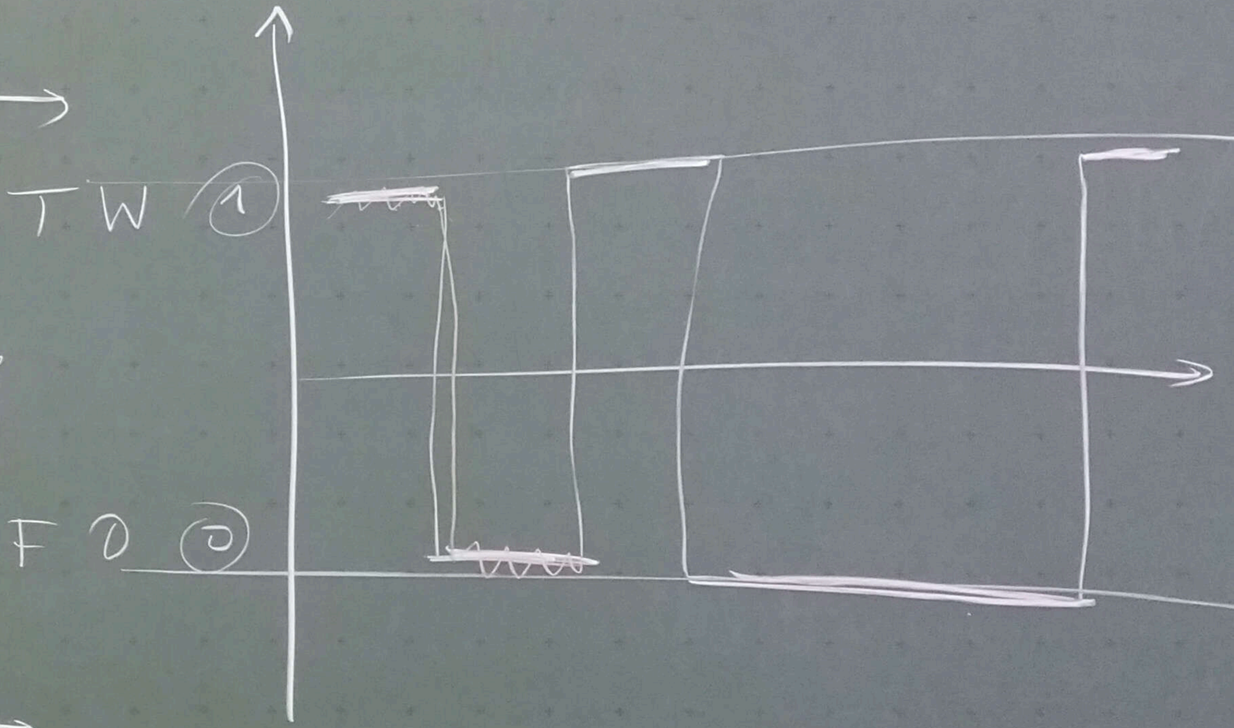
REPRODUCEERBAAR

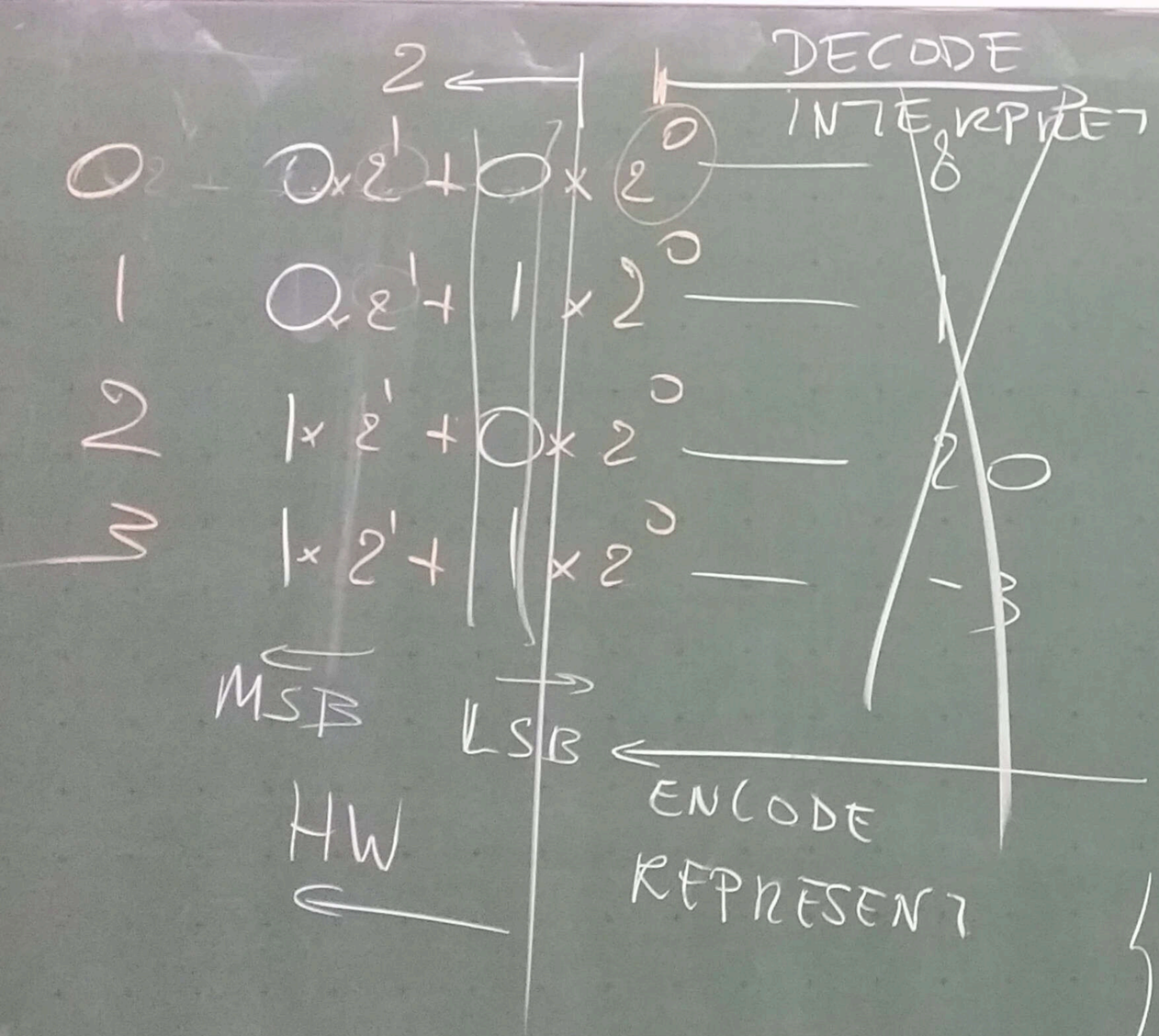


x AND y









$\mathbb{N}$

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

$\mathbb{N}$ $+, \times$

$\mathbb{Z}$

$\mathbb{Q}$

$\mathbb{R}$

MATH

01001_2

N bits $\rightarrow$ 2^N

$X = \underbrace{X_{n-1} \times 2^{n-1} \; \cdots \; X_2 \times 2^2 + X_1 \times 2^1 + X_0 \times 2^0}_{n \text{ bits}}$

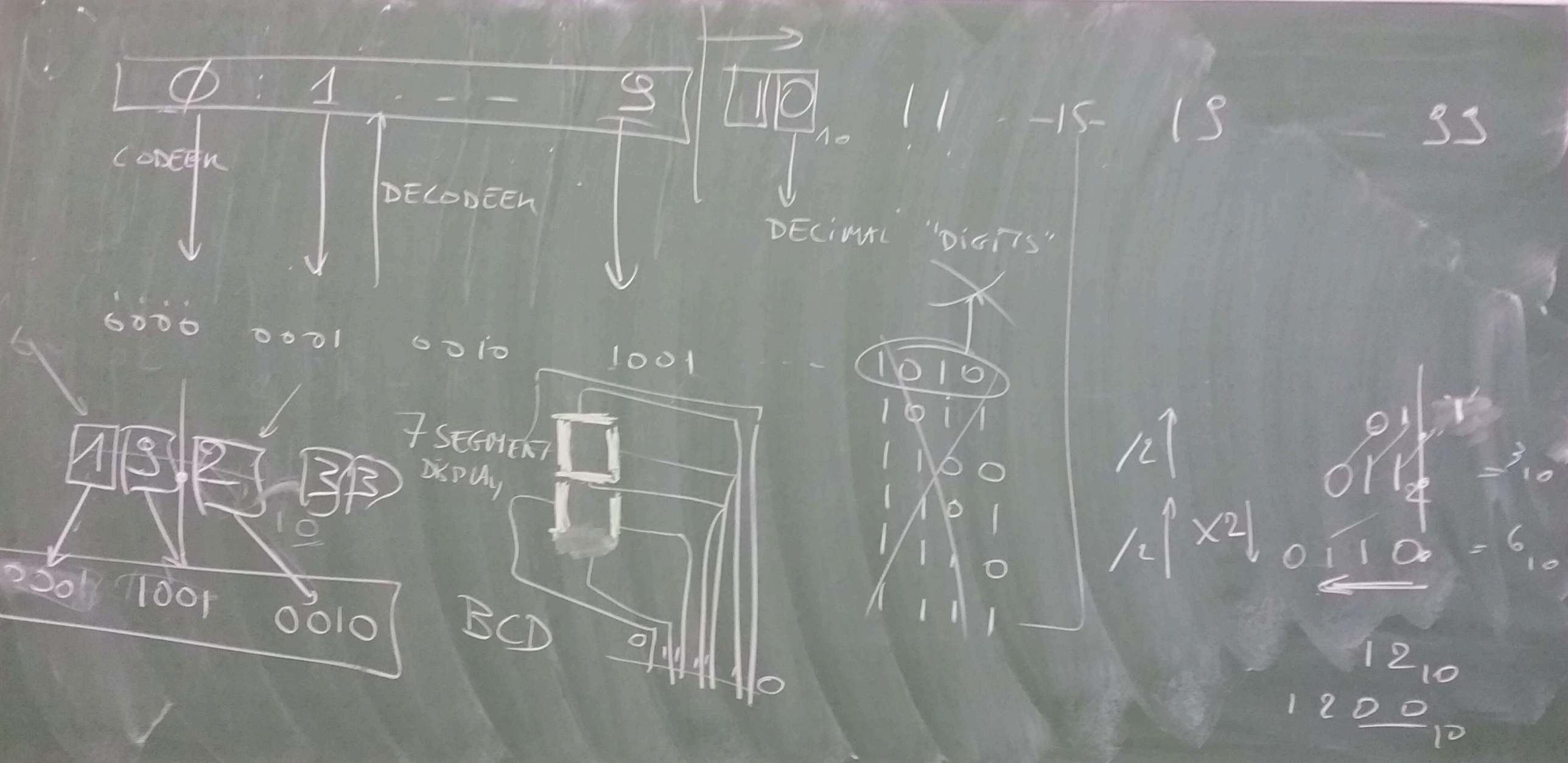
$\rightarrow 0, 1, \ldots, 2^N - 1$

01001

$1021 = 0 \times 10^4 + 1 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$

$\underline{9.75_{10}} \rightarrow 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$

1001.11_2


0 : 1 . . . 9
CODEEN
DECODEEN
10 11 . . . 15 16 . . . 99
DECIMAL "DIGITS"
0000 0001 0010 1001
1010
1 3 2 33
10
0001 1001 0010
7 SEGMENT DISPLAY
BCD
/2
/2 X2
0 1 1 1
0 1 1 0
= 3 10
= 6 10
12 10
1200 10

$2^{32} = 2^{2+10+10+10}$

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

$4 \quad 1024 \quad 1024 \quad 1024$

1_2

1_2

11111111111111111111_2

FFFFF

DECONT

DECONT

$001 \; 0001 \; 1111_2$

$3 \quad 1 \quad F_{16}$

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

123.9_{10}

0.2

$0.1_2 \quad 0.5_{10}$

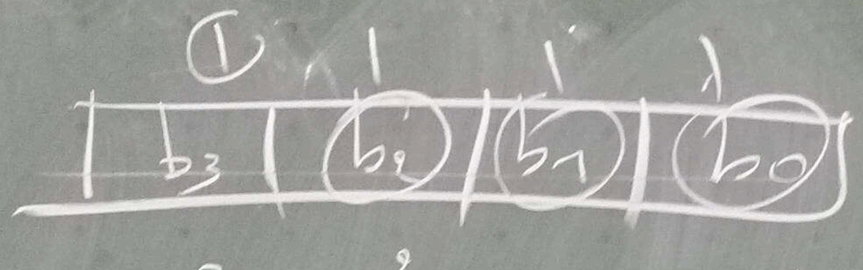
$0.01 \quad 0.25$

$0.001 \quad 0.125$

DECODE

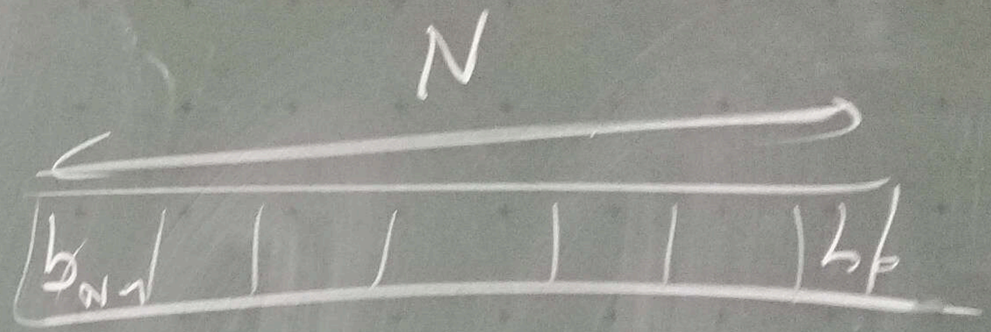
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	1	1	7

$8 = 2^3$ $\quad 2^2 = 4$

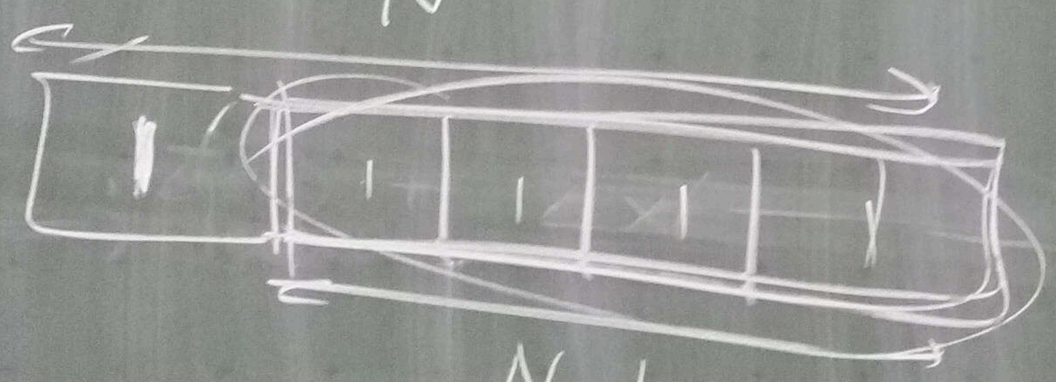


$2^3 \quad 2^2 \quad 2^1 \quad 2^0$

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

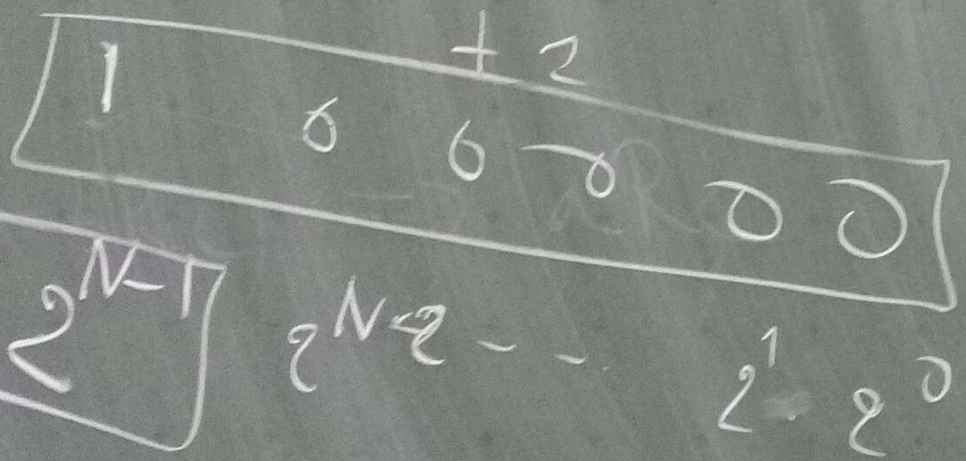


2'S COMPLEMENT



2^{N-1}

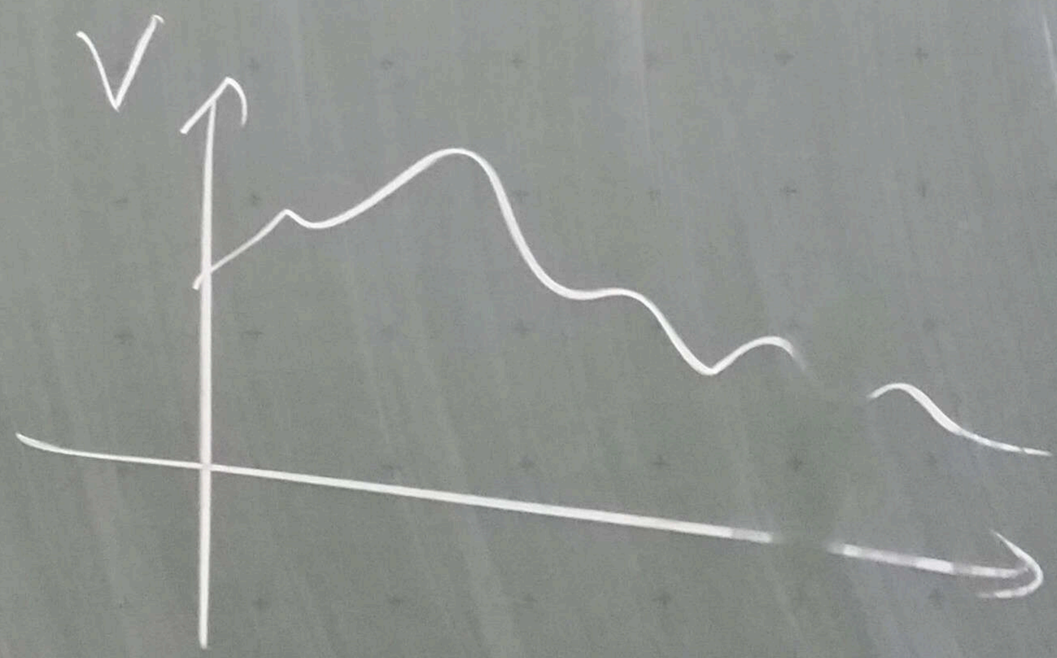
-1

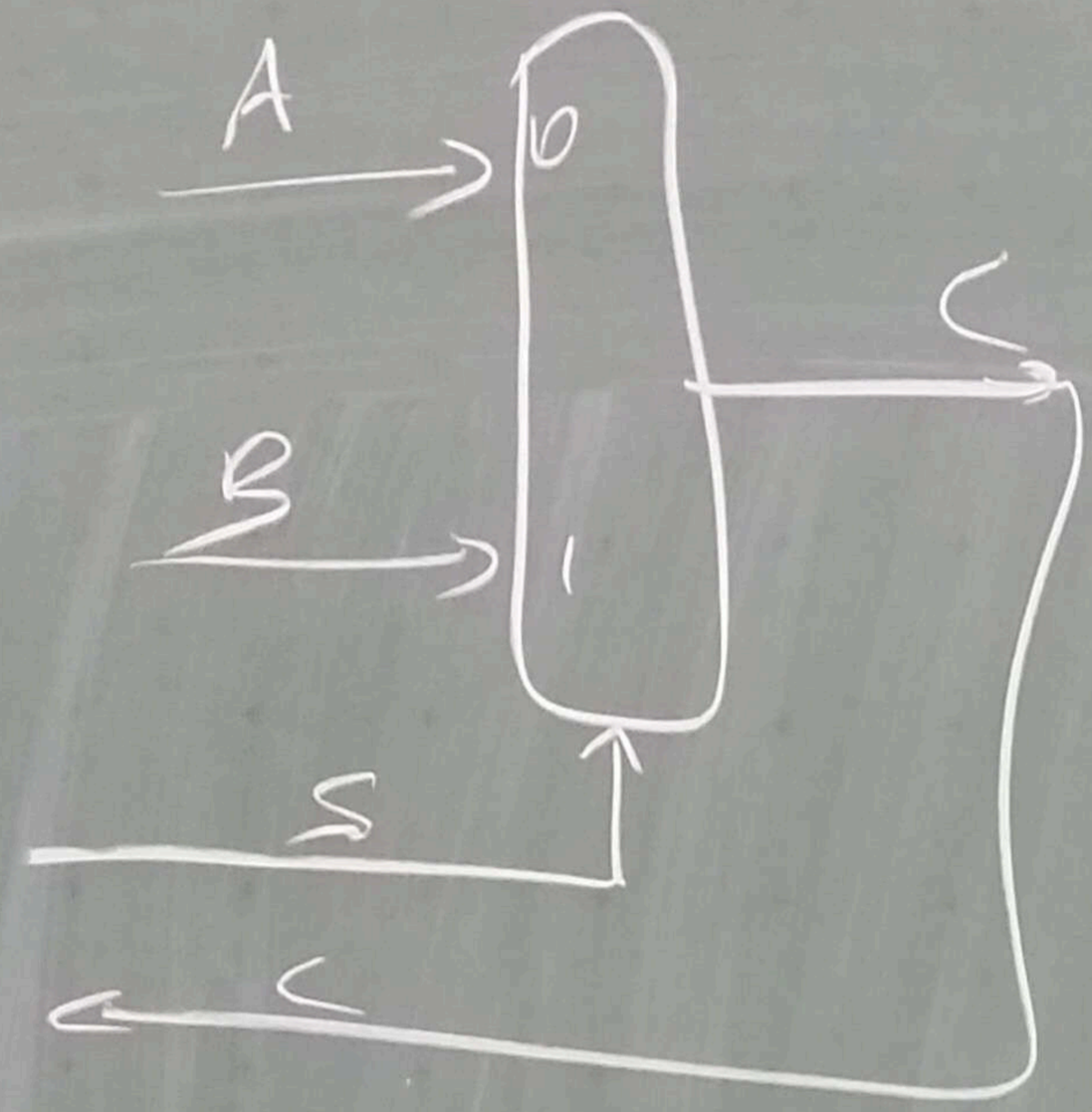


$$N-1 \qquad N-2 \qquad 1 \qquad 1 \qquad 0$$

$$-b_{N-1}z + b_{N-2}z + \cdots + b_1 z + b_0 z$$

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




A
B
S
C
0
1

A
1
0
t
B
1
0
t
S
1
0
t
C_T
1
0
t

$N = 3\ell$

$\mathbb{N}$ ⟶ b_{N-1} ... b_1 b_0

2^{N-1} ... 2^1 2^0

2^N

$0, 1, \ldots, 2^N - 1$

"RANGE"

$\mathbb{Z}$ ⟶ A. 1 - - - 0 1 1

B. 1 1 1 1 1 1 0 0

C.

D. BIASED

-3

SIGNED MAGNITUDE

1'S COMPLEMENT

2'S COMPLEMENT

$0000\,0011$ $\quad x = +3$

$1111\,1100$ $\quad \bar{x}$

$0000\,0001$ $\quad +1$

$1111\,1101$ $\quad -3$

2's COMPLEMENT

x $\quad 1101110_2$

$+\ \bar{x}$ $\quad 0010001_2$

$x + \bar{x} = 1111111_2$ $\quad -1$

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

$-x$

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

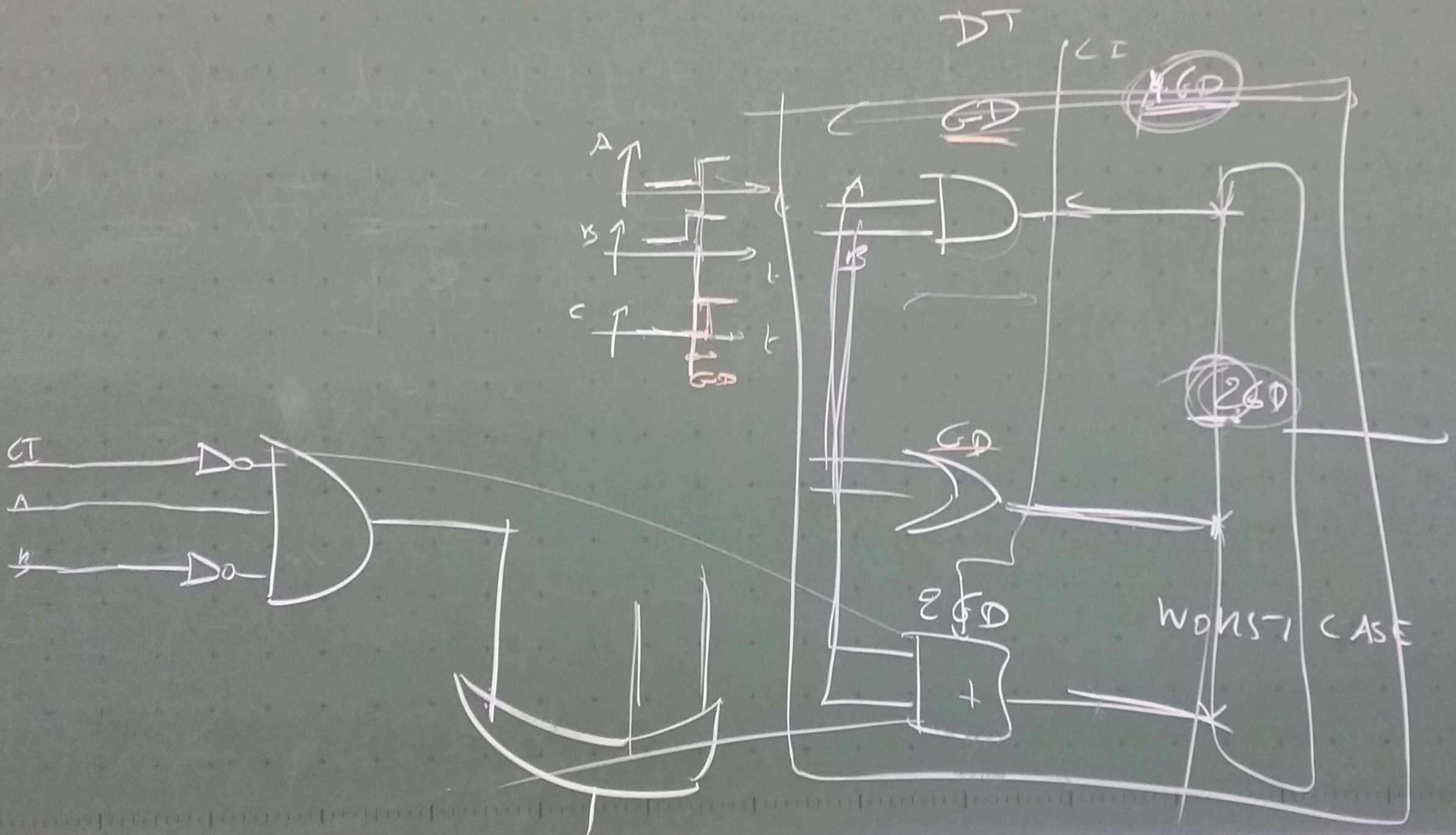
$A < B$

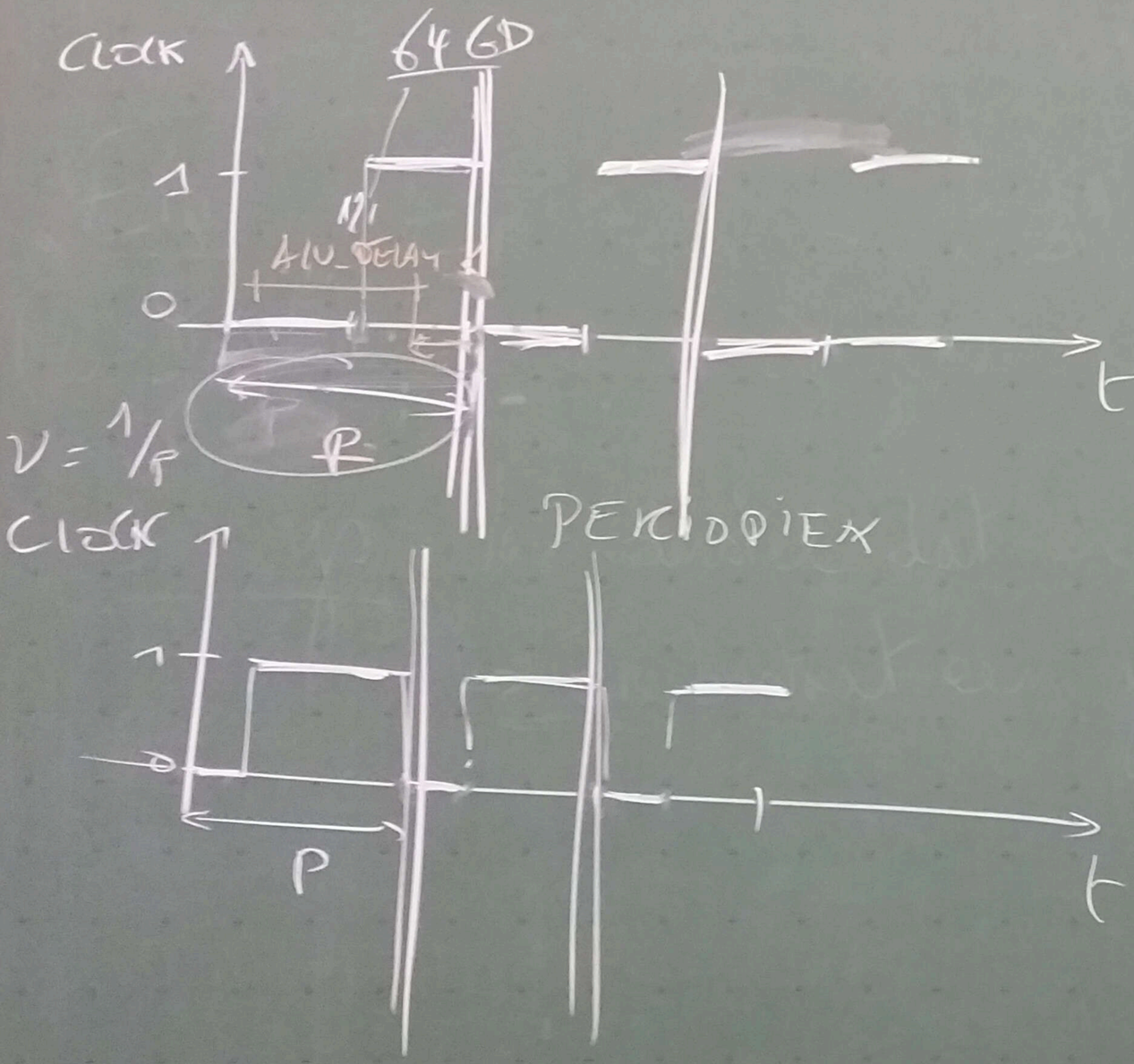
$A - B < B - B$

$A - B = A + (-B)$

$A + B = A + (\bar{B} + 1)$

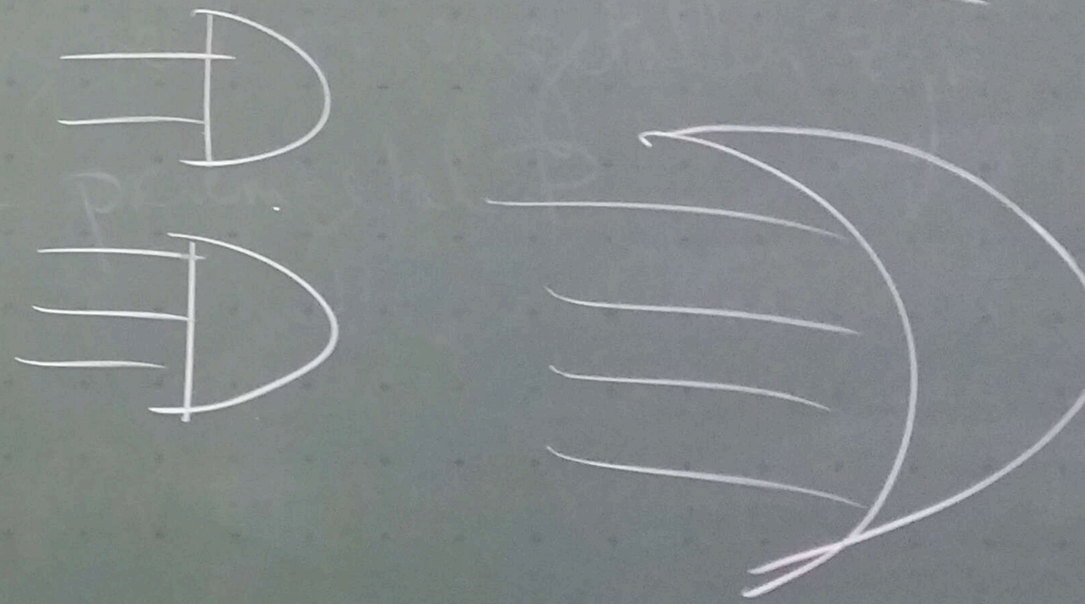
ALU_DELAY





2 GD

7 + 1 = 8



AND OR

P $1 + 4 = 5$

G $3 + 1 = 4$

$4 + 6 = 10$

LS

AND $4 + 4 = 8$

OR $1 + 10 = 11$

14 + (4 x) (18) GATES

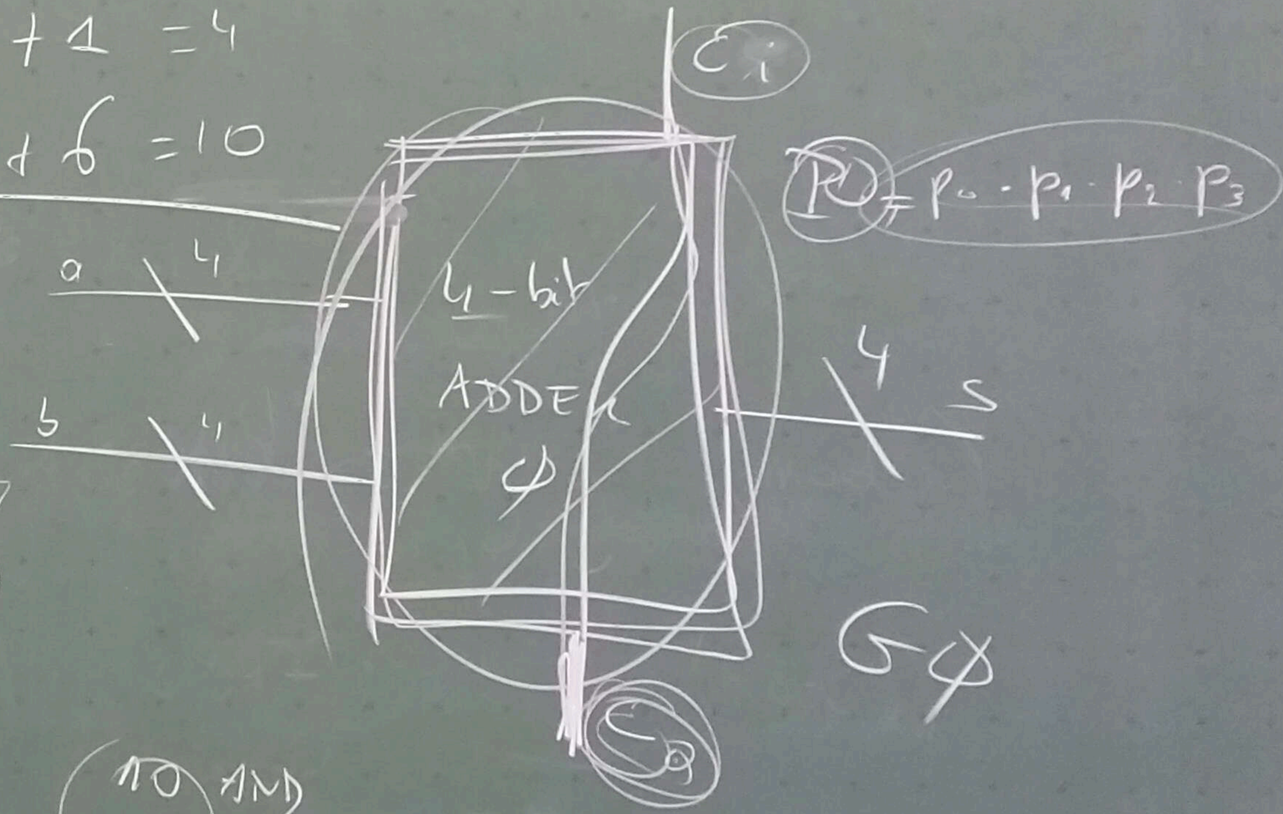
DELAY C_4 = (3 GD)

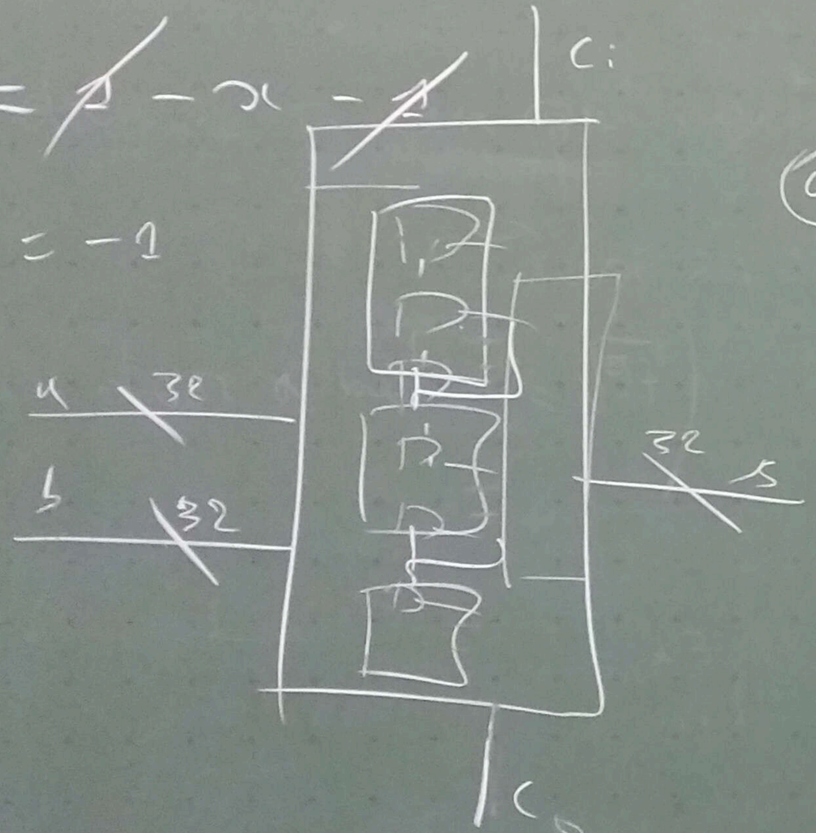
P's C (7+1)

(7+1) G's

(10 AND) (4 OR)

14





$a_{-b} < b_{-b}$

$(a-b) < 0$

16 b.t

- RIPPLE CARRY

D = 32 GD

GATES

- CARRY LOOK-AHEAD

4b, 4x

D = 5 GD

- INFINITE FAN

D = 2 GD

AND OR

P 1+4 = 5

G 3+1 = 4

4+6 = 10

15

AND 4+4 = 8

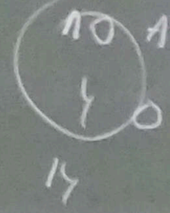
OR 1+10 = 11

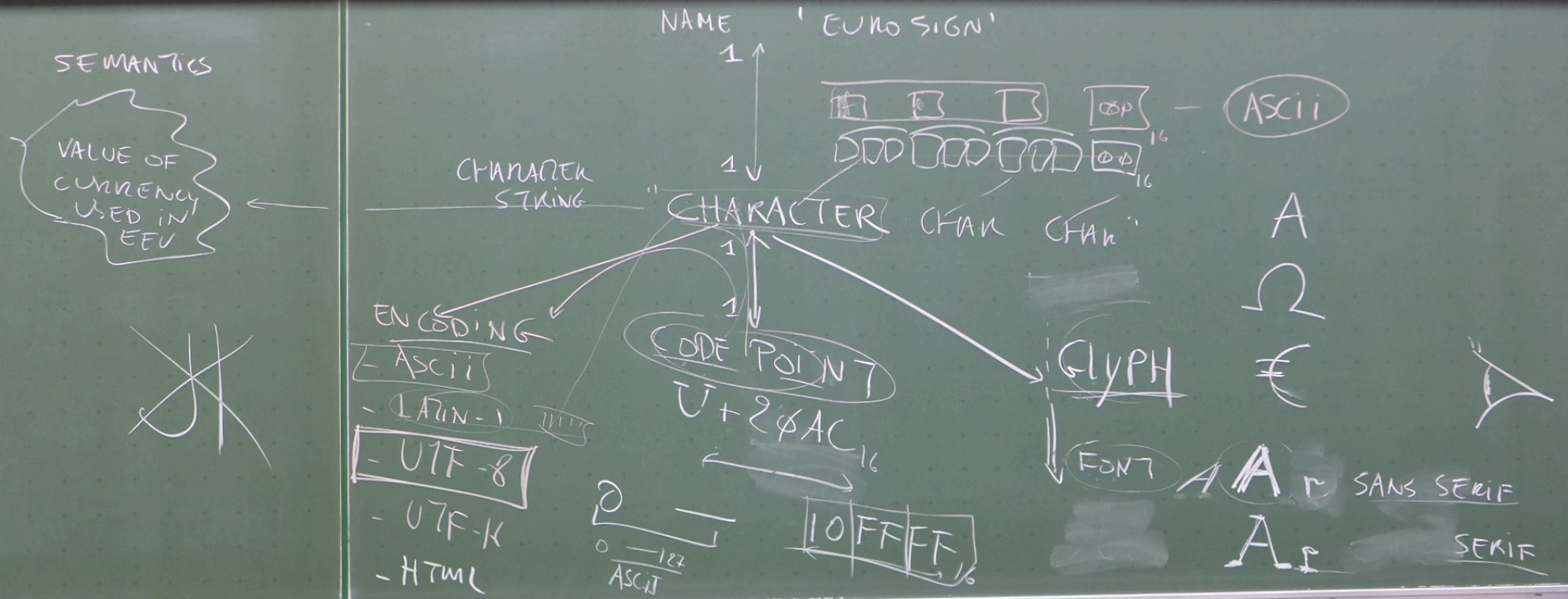
14 + 4x 15 GATES

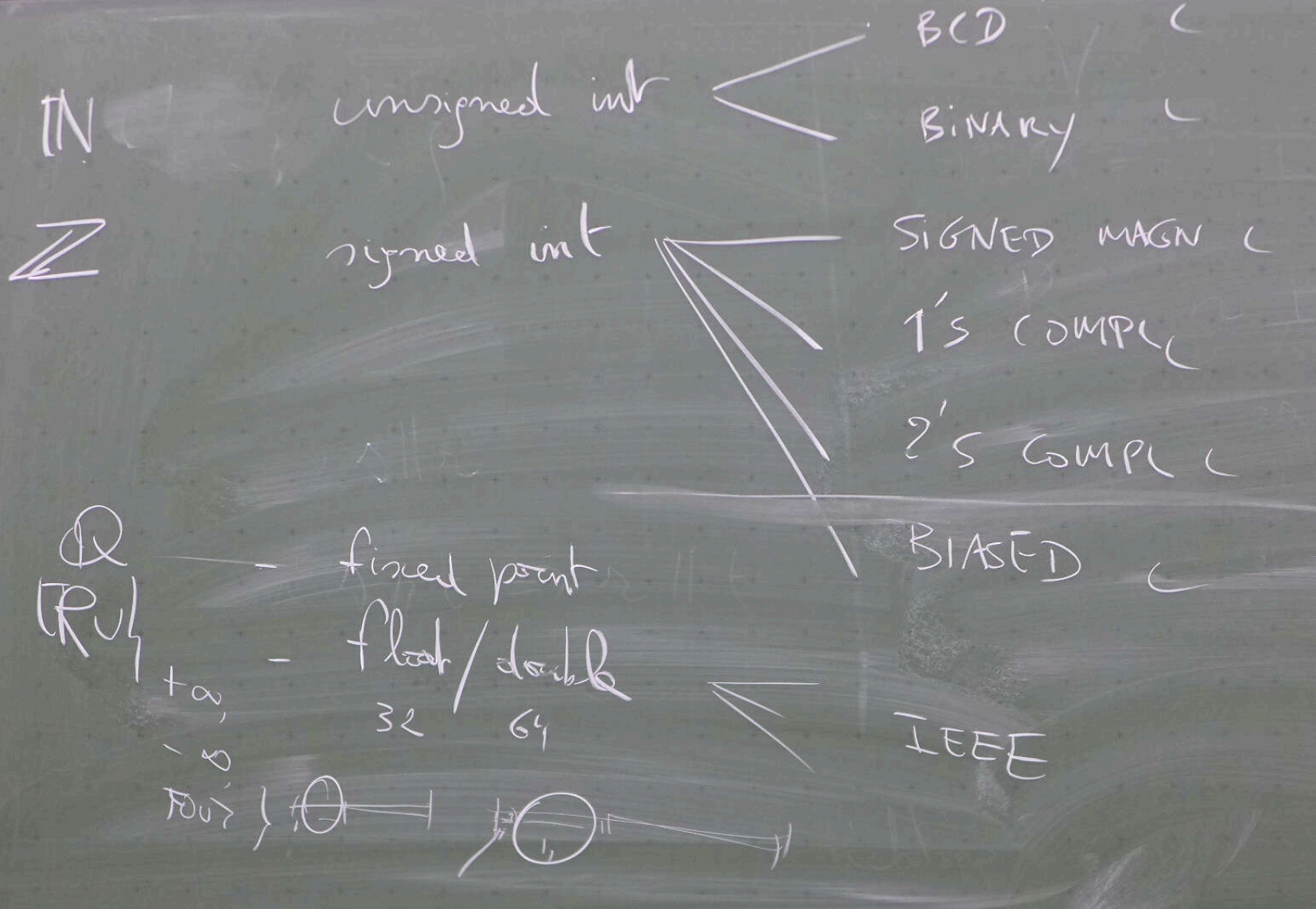
DELAY CY = 3 GD

P's 1+1

G's 1+1




SEMANTICS
VALUE OF CURRENCY USED IN EEU
NAME
'EURO SIGN'
1
1
CHARACTER STRING
"CHARACTER"
CHAR
CHAR
ASCII
ENCODING
- ASCII
- LATIN-1
- UTF-8
- UTF-K
- HTML
CODE POINT
U+20AC16
0
127
ASCII
10FFFF16
GLYPH
FONT
SANS SERIF
SERIF


BCD
unsigned int
BINARY
SIGNED MAGN
signed int
1's COMPL
2's COMPL
BIASED
fixed point
floating
32
64
IEEE

EXP

37 ← E1

39 ← E2

8.34 × 10

8.333 × 10

+	1	1	1
	1	1	1
	1	1	0

−1

−1

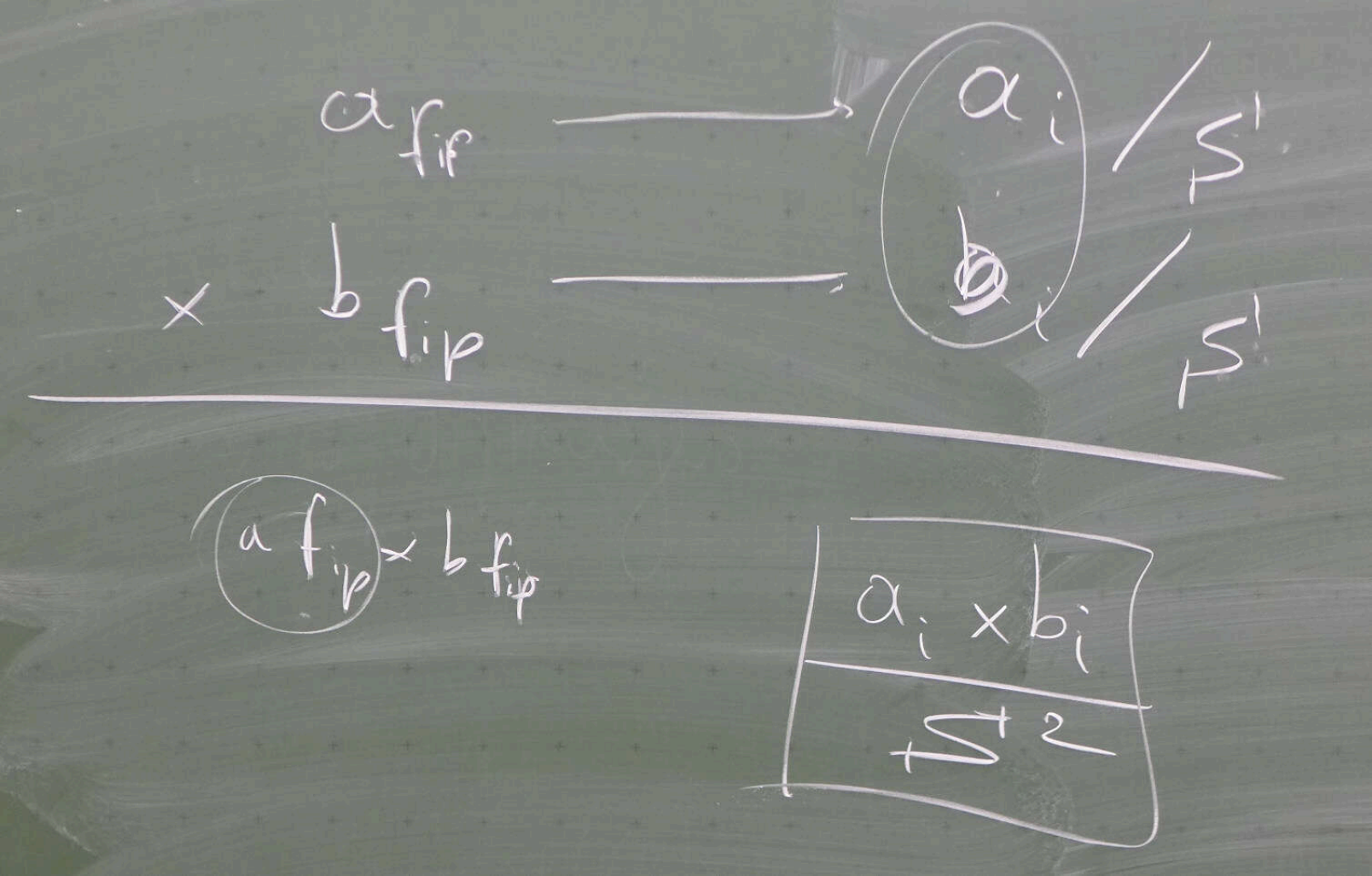
−2

0 1 0

+ 1 0 1

1 1 1


0000
0001
0010
0011
1111
+∞
−∞
NaN


$a_{f_{ip}}$
$\times$ $b_{f_{ip}}$
a_i
b_i
$/S^1$
$/S^1$
$a_{f_{ip}} \times b_{f_{ip}}$
$a_i \times b_i$
$\in S^2$

$$(0.10)_2 \times (0.01)_2 \times (0.10)_2$$

$$(0.10)_2 \rightarrow 0.5$$

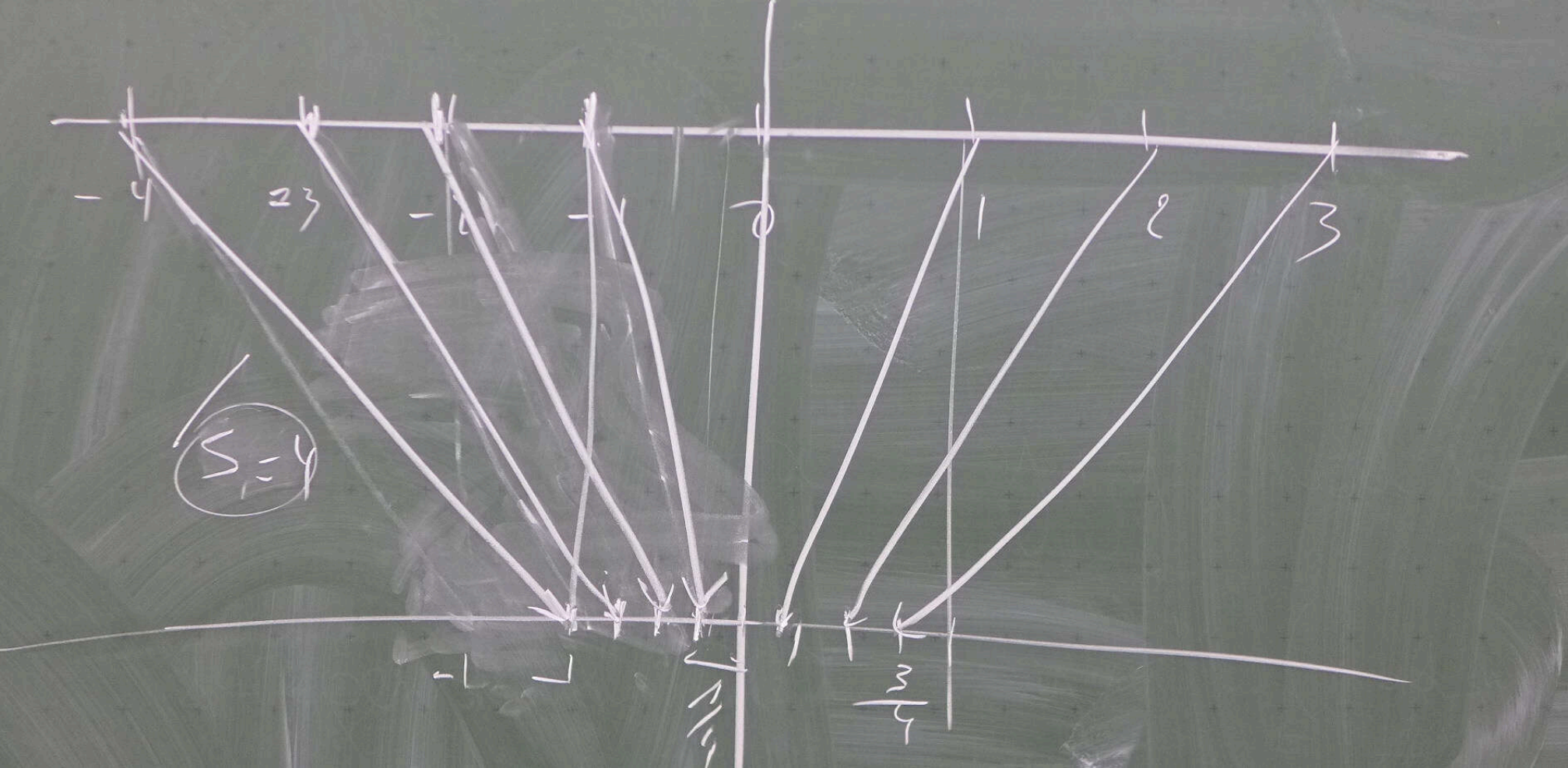
$$(0.01)_2 \rightarrow 0.25$$

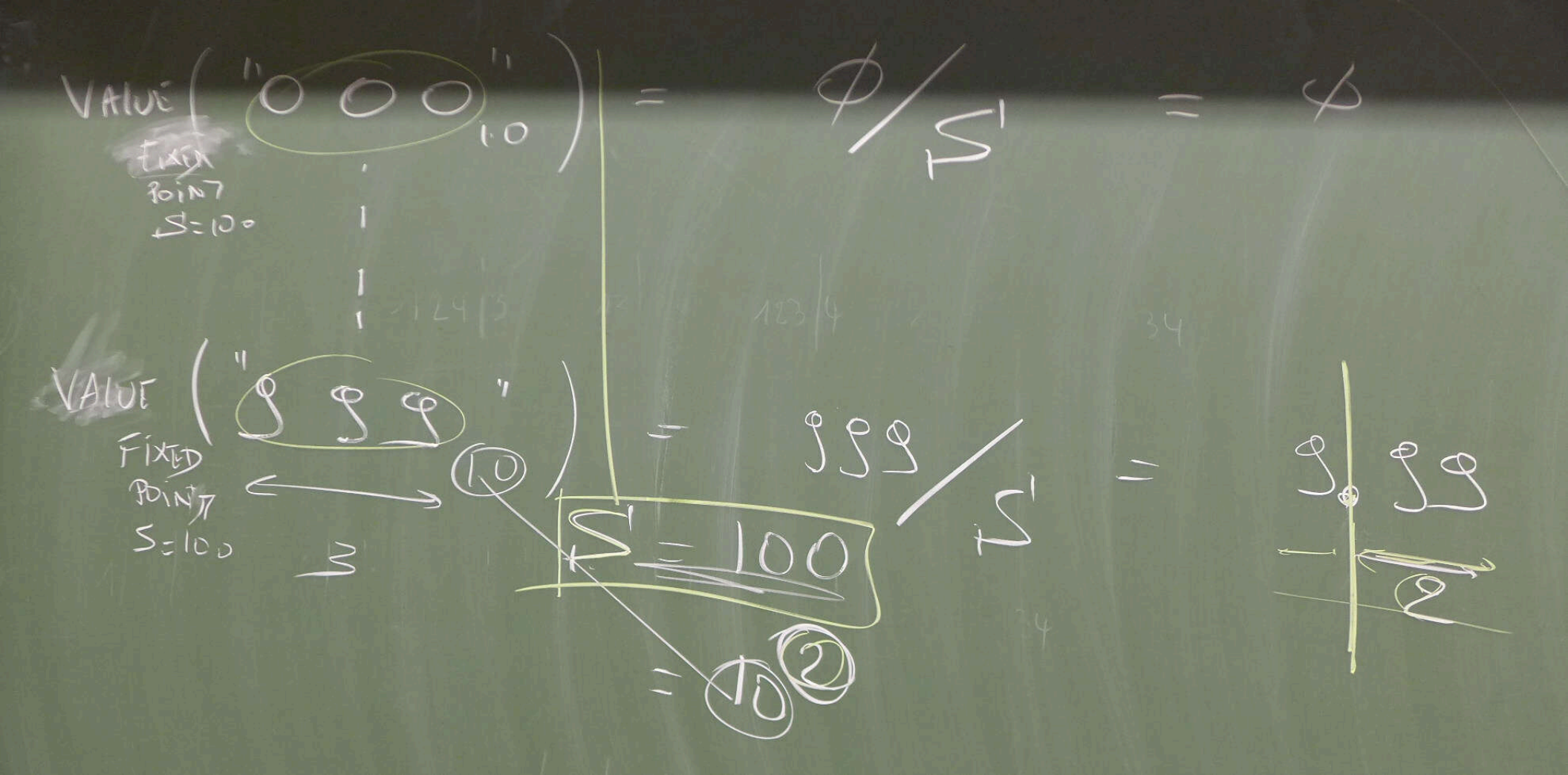
$$(0.10)_2 \rightarrow 0.5$$

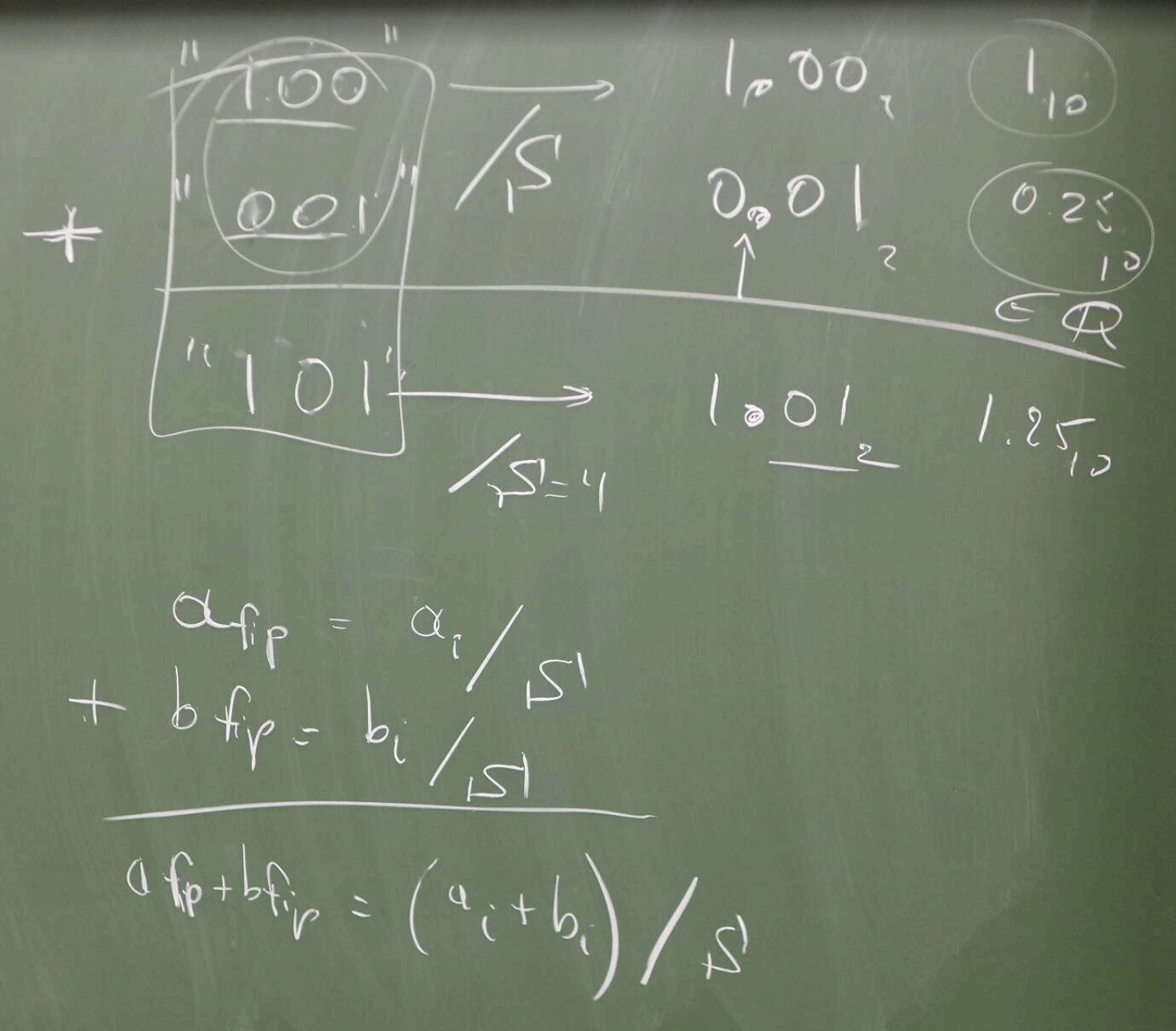
$$0.5 \times 0.25 \times 0.5 = 0.0625$$

$$(0.5)_{10} \neq (0.125)_{10}$$

4


-4
-3
-2
-1
0
1
2
3
S=4
-1
1/2
3/4


VALUE ("000" 10) = 0 / S' = 0
FIXED POINT S=10
VALUE ("999" 10) = 999 / S' =
S' = 100
= 10 2


"100"
"001"
"101"
/s
/s=4
1,00₂
1₁₀
0,01₂
0.25₁₀
∈ ℝ
1,01₂
1.25₁₀
+
$a_{fip} = a_i / s$
$+ \; b_{fip} = b_i / s$
$a_{fip} + b_{fip} = (a_i + b_i) / s$

VALUE("000") = +0

VALUE("000") = -4

VALUE FIXEDP("111") =

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

1.11

2

1.75

10

N = 3

S = 4

SMALLEST = -2^{N-1}

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

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

S

2^{S}

$\frac{-4}{4}$ $\frac{-3}{4}$ $\frac{-2}{4}$ $\frac{-1}{4}$ $\frac{0}{4}$ $\frac{1}{4}$ $\frac{2}{4}$ $\frac{3}{4}$

1/4 1/4 1/4 1/4

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

$b = 10$

$s = 5$

$M = 99$

$m = -99$

12

LARGEST REPR. $= +0.99999 \times 10^{tss} + \left(1-b^{-\Delta}\right) \times b^{m}$

$0.99999 = 1 - 1 \times b^{-5}$

$+\ 0.00001$

$+\ 1\ 0.00000$

SMALLEST REPR. $= 0.10000 \times 10^{-99} =$

$1 \times b^{-1} \times b^{m} = (b-1) \sum_{i=1}^{\Delta} b^{-i}$

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

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

LARGEST GAP $= 0.00001 \times 10^{99} = 1 - b^{-\Delta}$

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

SMALLEST GAP $= 0.00001 \times 10^{-99}$

$b^{-1} \times b^{m}$

GAP AROUND 0

$2\left(1-b^{-1}\right)b^{m}$

$+ 11.0110 \times 2^{2}$

1101.10×2^{0}

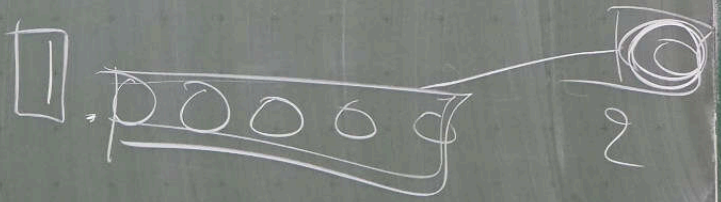
0.000×2^{1}

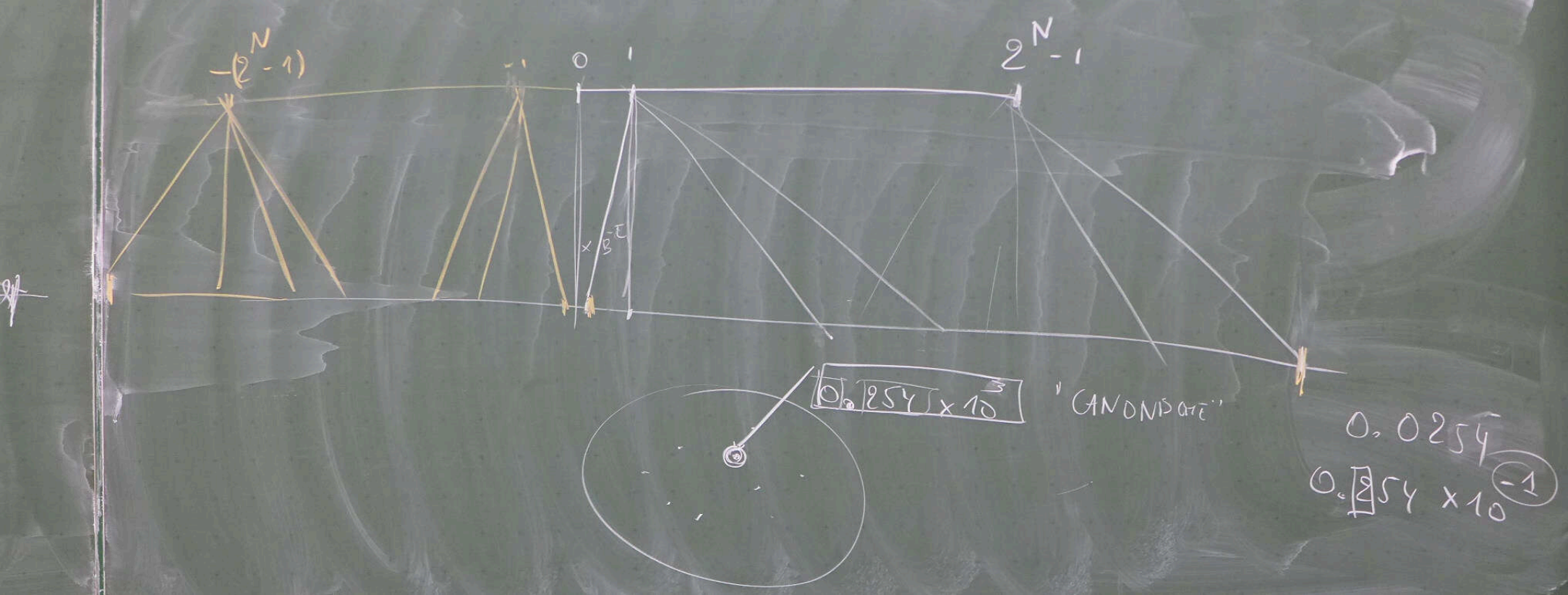
CANONISCH

EXP

A 0.110110×2^{4}

B 0.10110×2^{3}



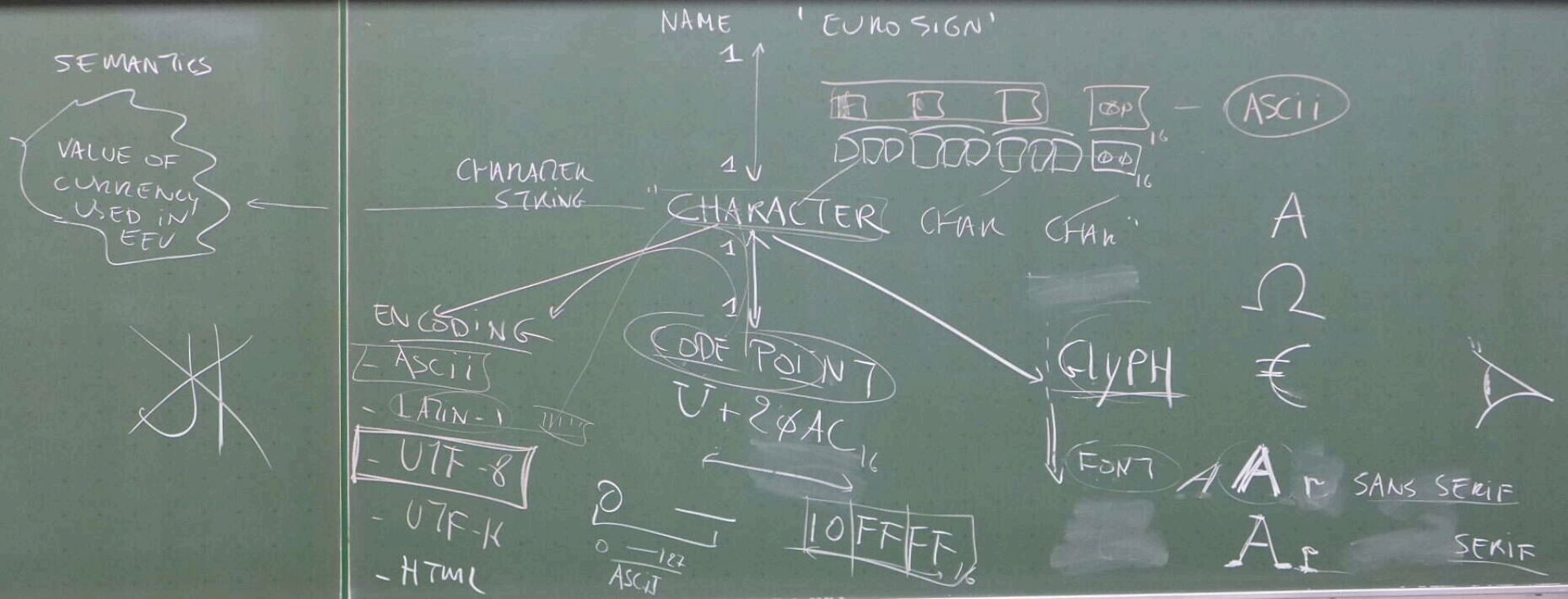

$-(2^N - 1)$
-1
0
1
$2^N - 1$
x
B-E
0,254 x 10^-3
"CANONISATE"
0,0254
0,254 x 10^-1

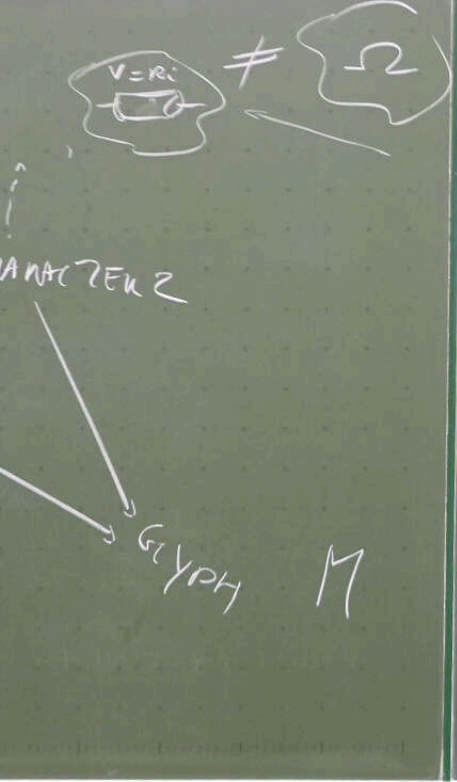
ASCII 7-bit

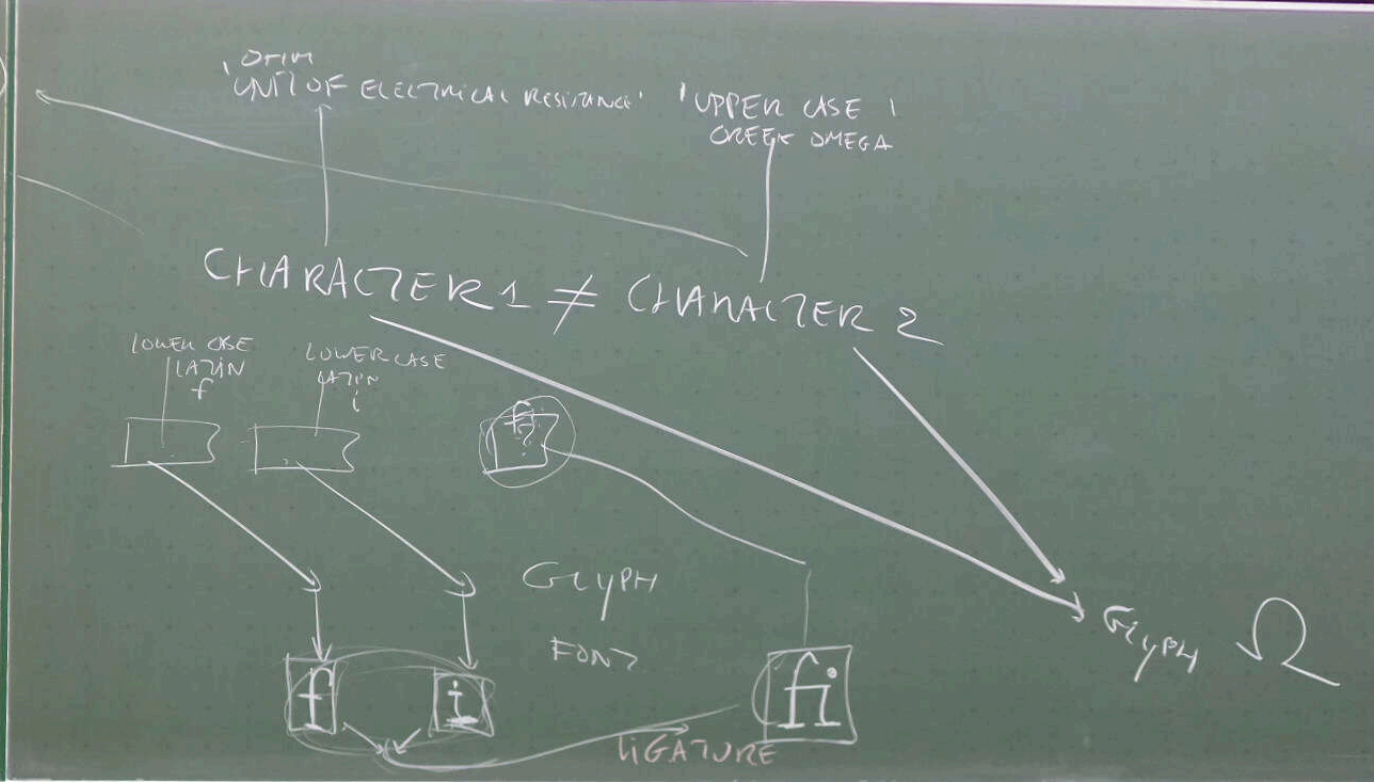
EXTENDED ASCII 8 bit

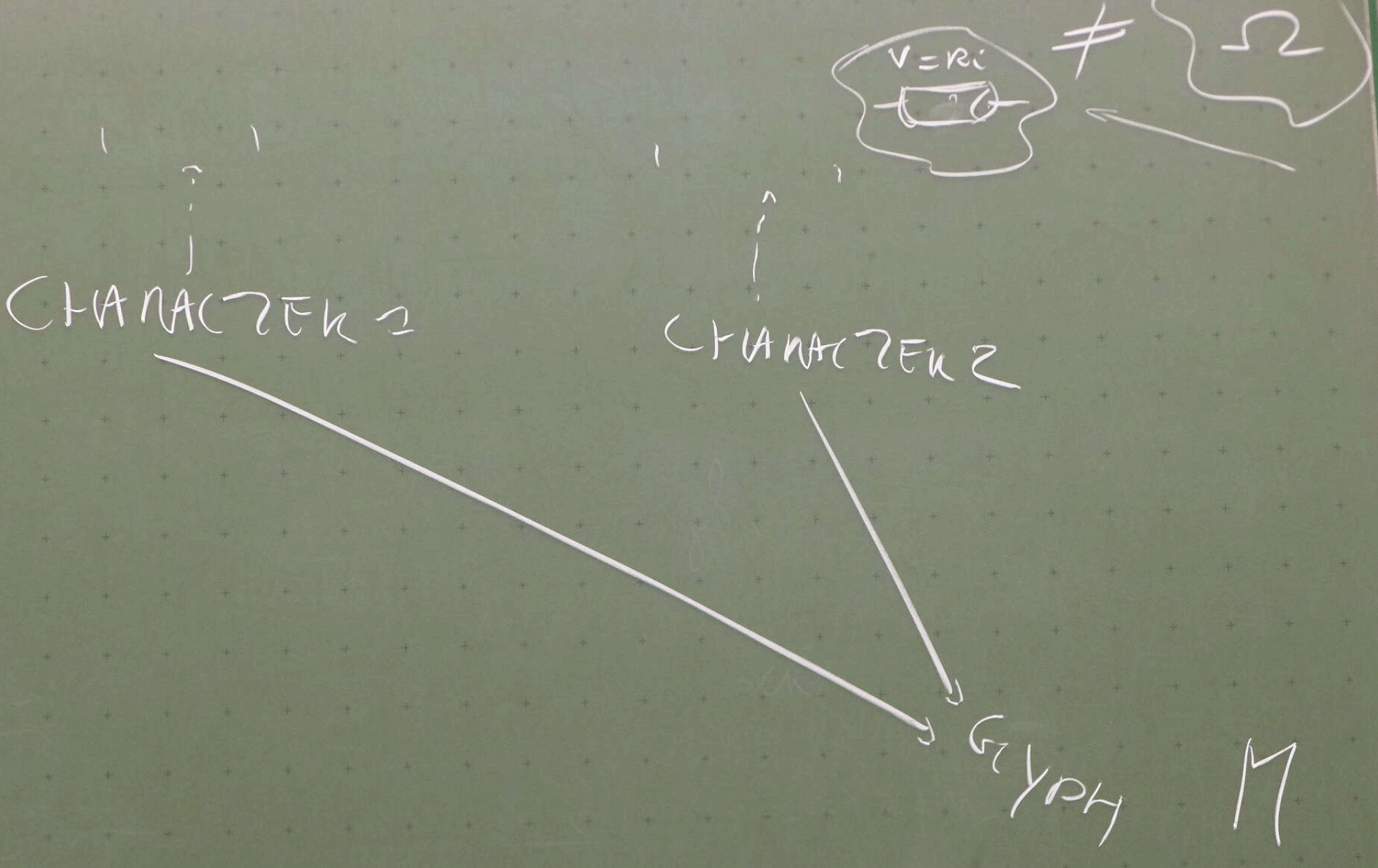
EBCDIC 8 bit

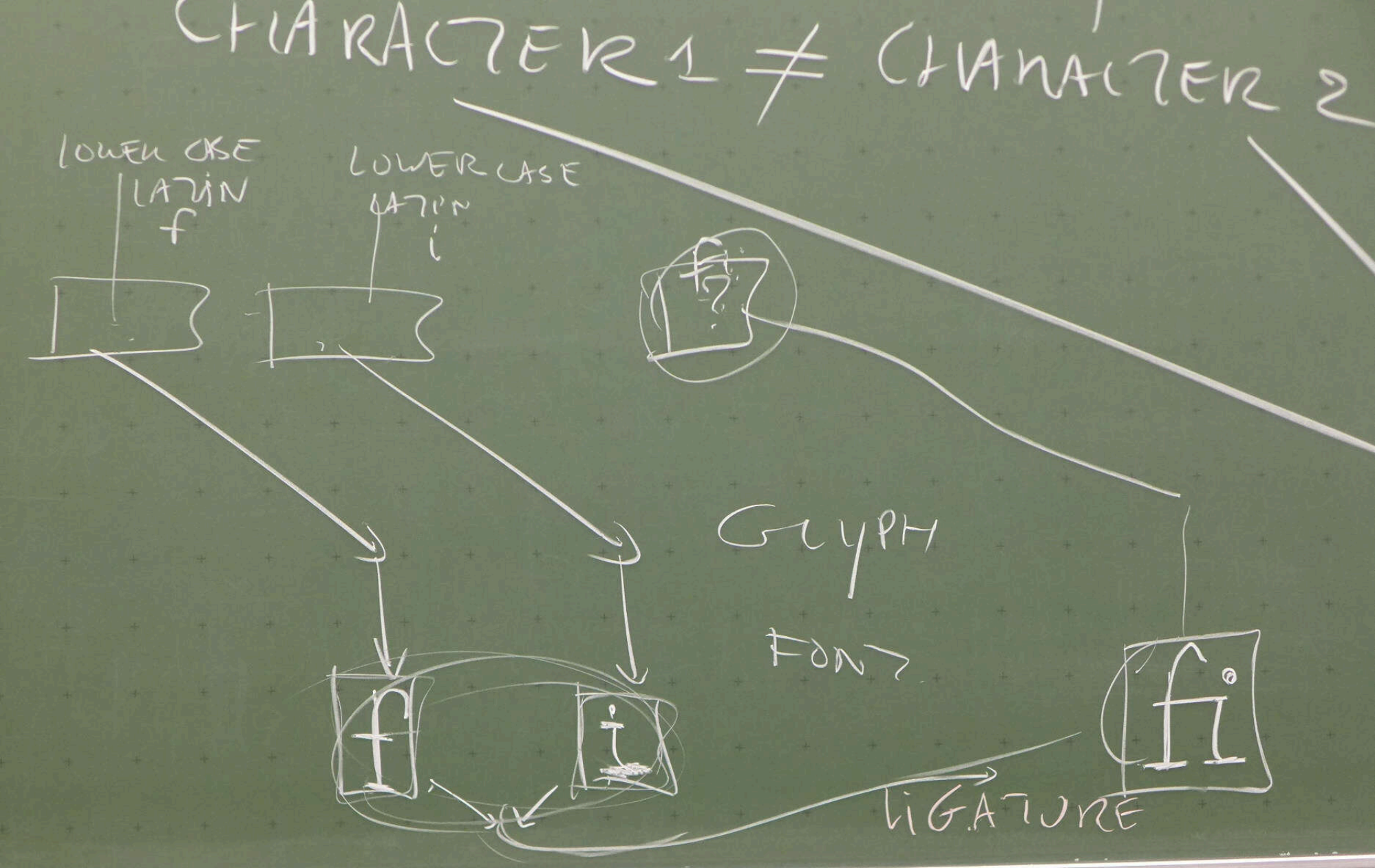
UNICODE ...

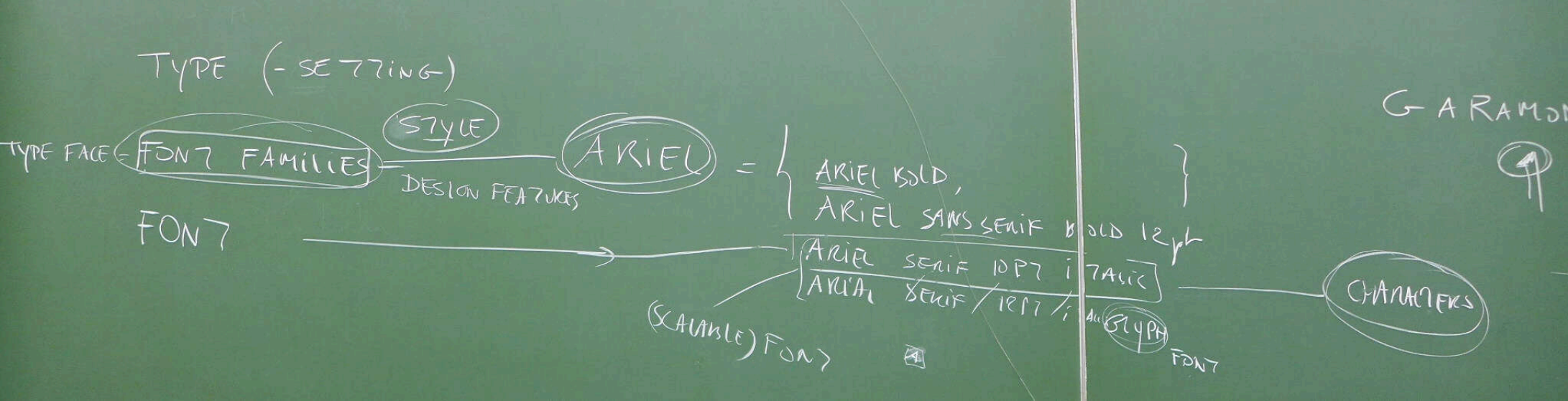

SEMANTICS
VALUE OF CURRENCY USED IN EEU
NAME
'EURO SIGN'
1
1
CHARACTER STRING
"CHARACTER"
CHAR
CHAR
ASCII
ENCODING
- ASCII
- LATIN-1
- UTF-8
- UTF-16
- HTML
CODE POINT
U+20AC16
0
127
ASCII
10FFFF16
GLYPH
FONT
SANS SERIF
SERIF


V=Ri
≠
Ω
i
GLYPH
M


'OHM'
'UNIT OF ELECTRICAL RESISTANCE'
'UPPER CASE I'
CREEK OMEGA
CHARACTER 1 ≠ CHARACTER 2
LOWER CASE LATIN f
LOWER CASE LATIN i
GLYPH
FONT
GLYPH
Ω
LIGATURE


V=Ri
≠
CHARACTER 1
CHARACTER 2
GROUP M

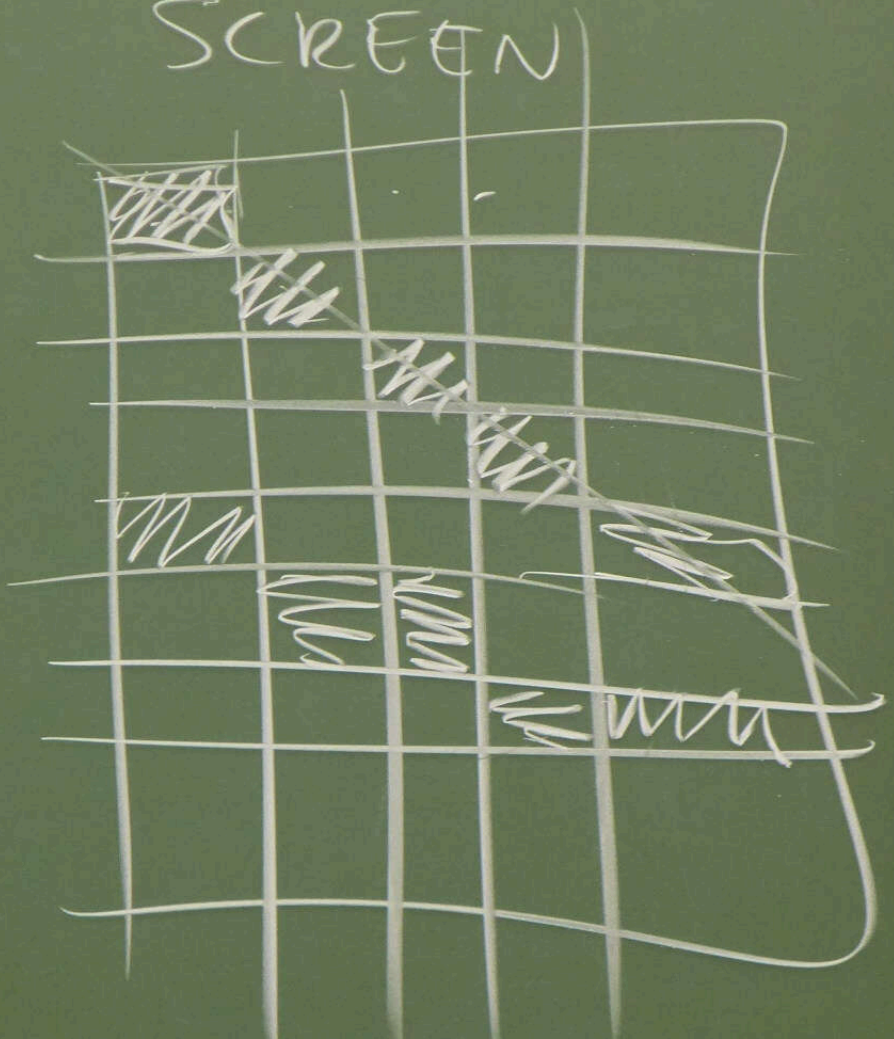

CHARACTER 1 ≠ CHARACTER 2
LOWERCASE LATIN f
LOWERCASE LATIN i
?
GLYPH
FONT
fi
LIGATURE


TYPE (-SETTING)
TYPE FACE
FONT FAMILIES
STYLE
DESIGN FEATURES
ARIEL
= { ARIEL BOLD, ARIEL SANS SERIF BOLD 12pt }
FONT
ARIEL SERIF 10PT ITALIC
ARIAL SERIF / 12PT / i
(SCALABLE) FONT
GLYPH
FONT
CHARACTERS
G A RAMON

PRINT

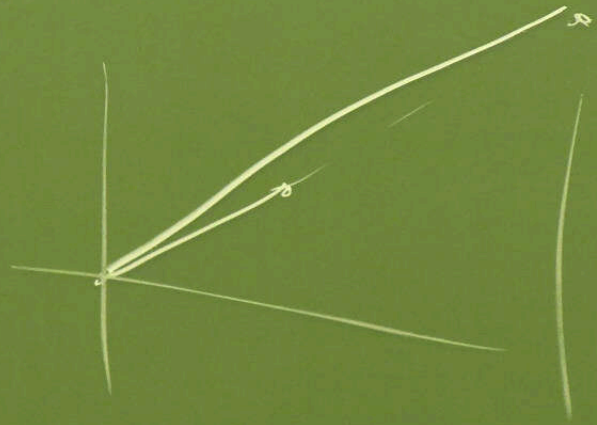
SCREEN

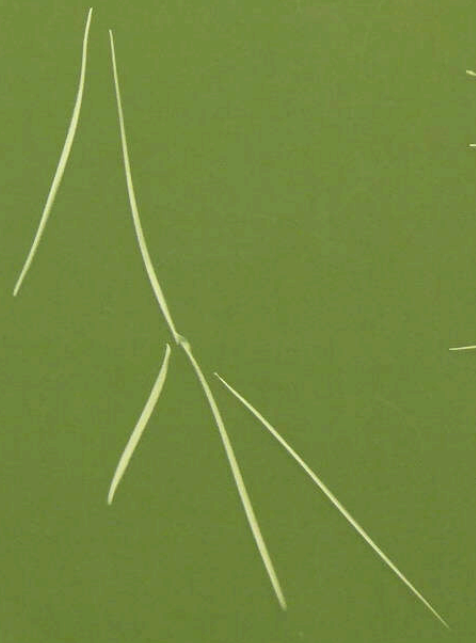
1600



GARAMOND

1600





CHARACTERS

HIGH-LEVEL LANGUAGE CODE

C, PYTHON

A = (B + C * D);

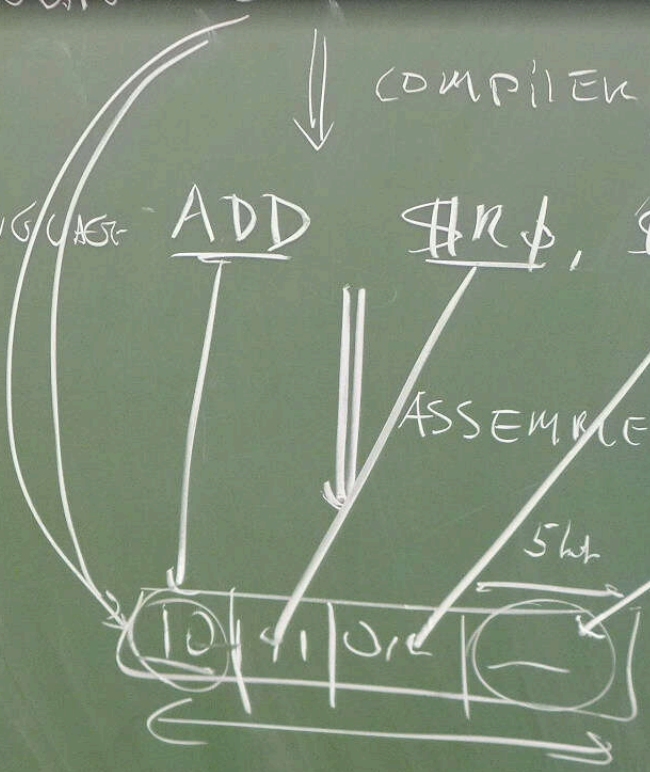
COMPILER

GCC

ASSEMBLY LANGUAGE CODE

ADD $R0, $R1, $R31

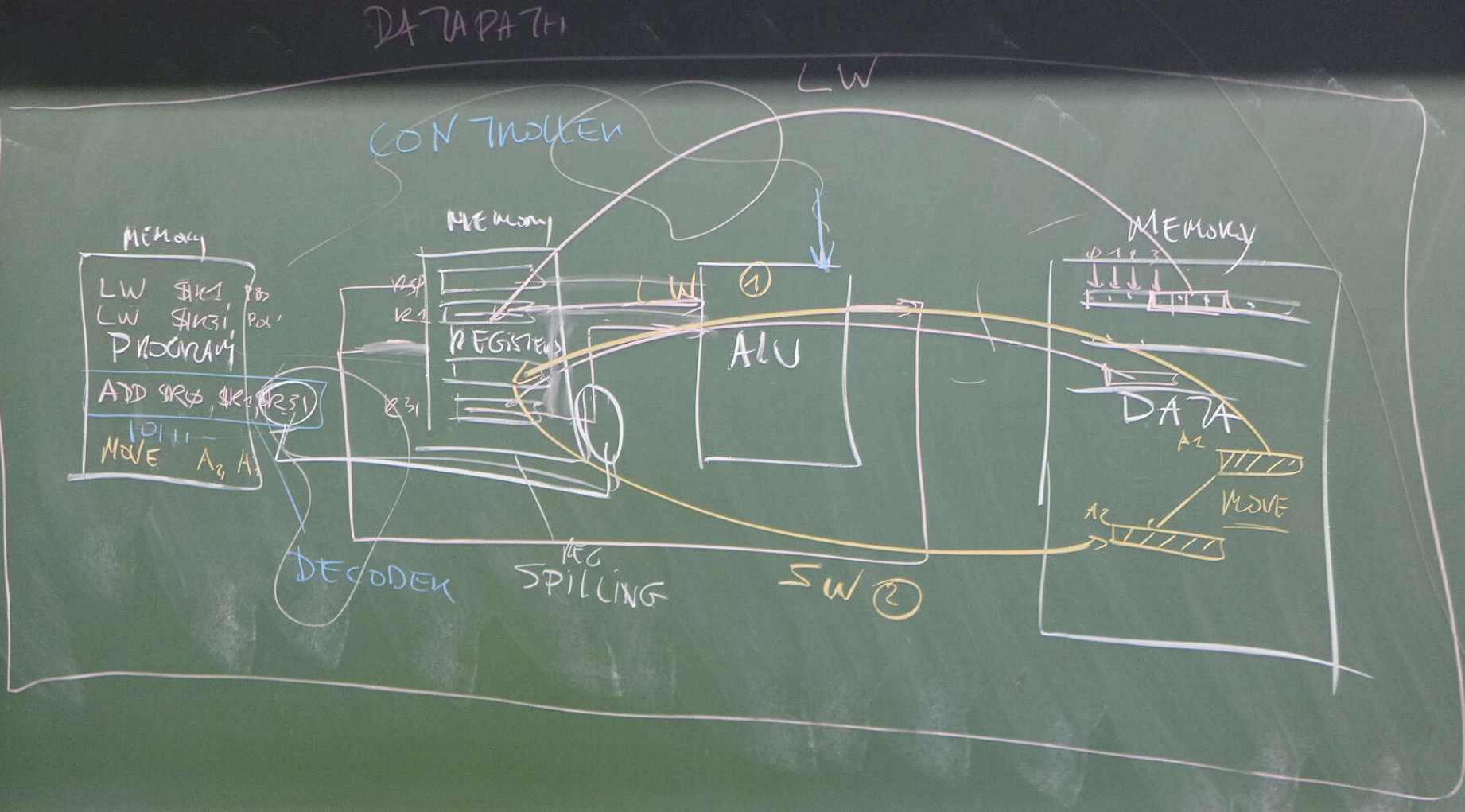
ASSEMBLER (COMPILER)

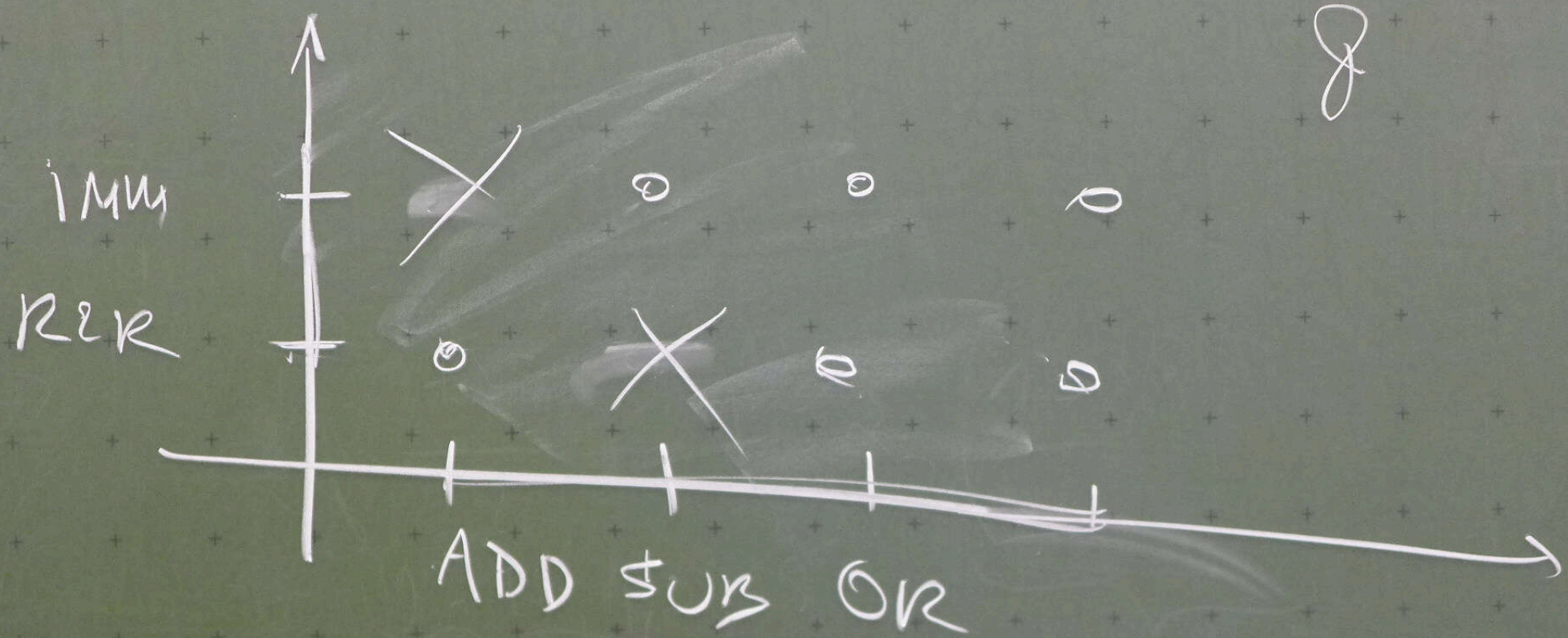


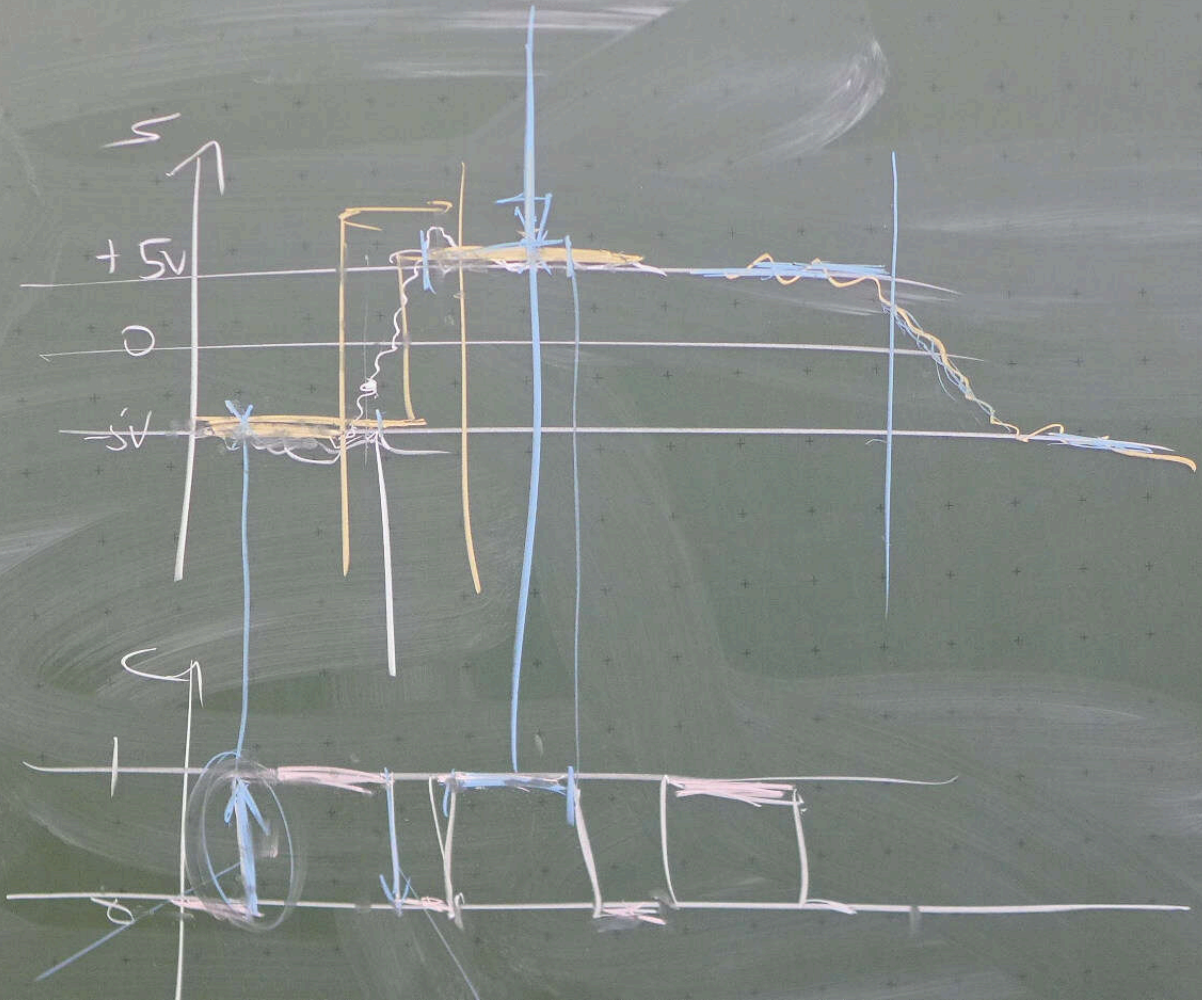
3 3 3bit

MACHINE CODE

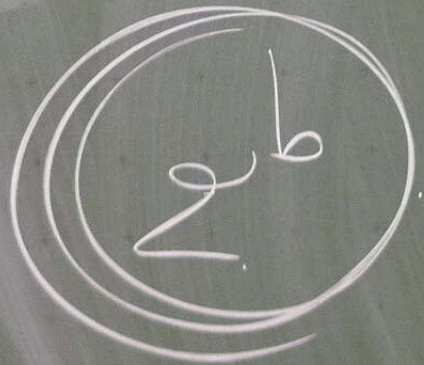
HARDWARE / DATAPATH

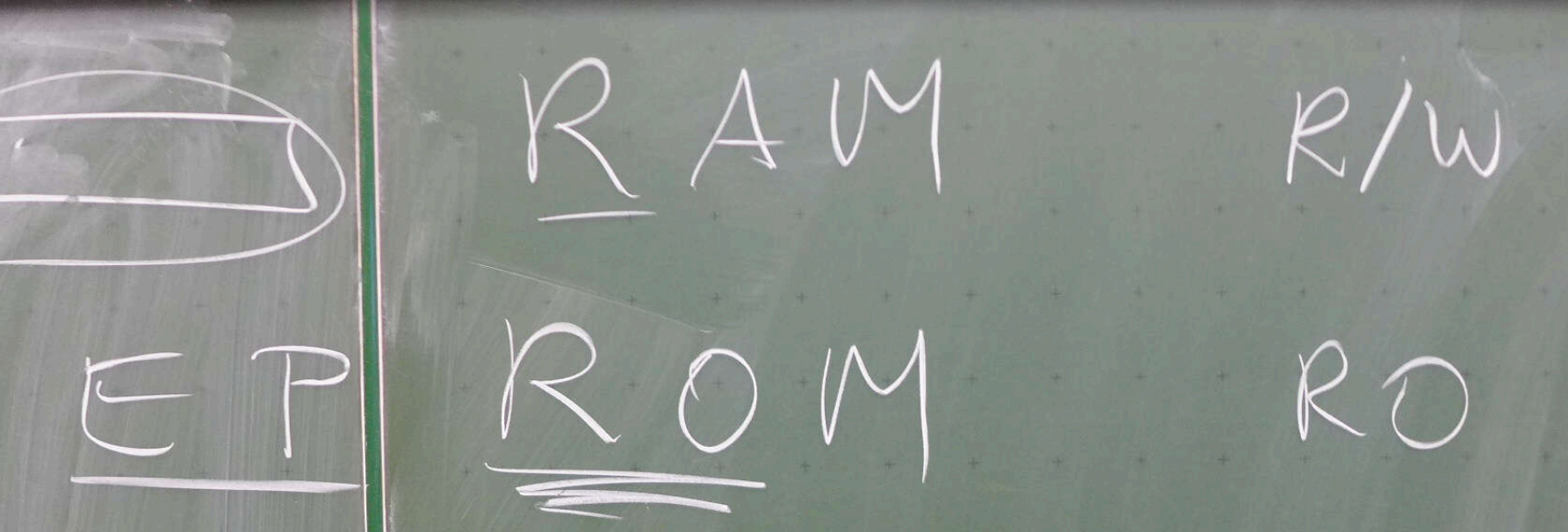

DATAPATH
LW
CONTROLLER
MEMORY
LW $R1, 100
LW $R31, 104
PROGRAM
ADD $R0, $R1, $R31
10 11 ...
MOVE A2, A1
DECODER
MEMORY
REGISTERS
R0
R1
R31
LW
1
ALU
REG SPILLING
SW 2
MEMORY
0 1 2 3
DATA
A1
A2
MOVE

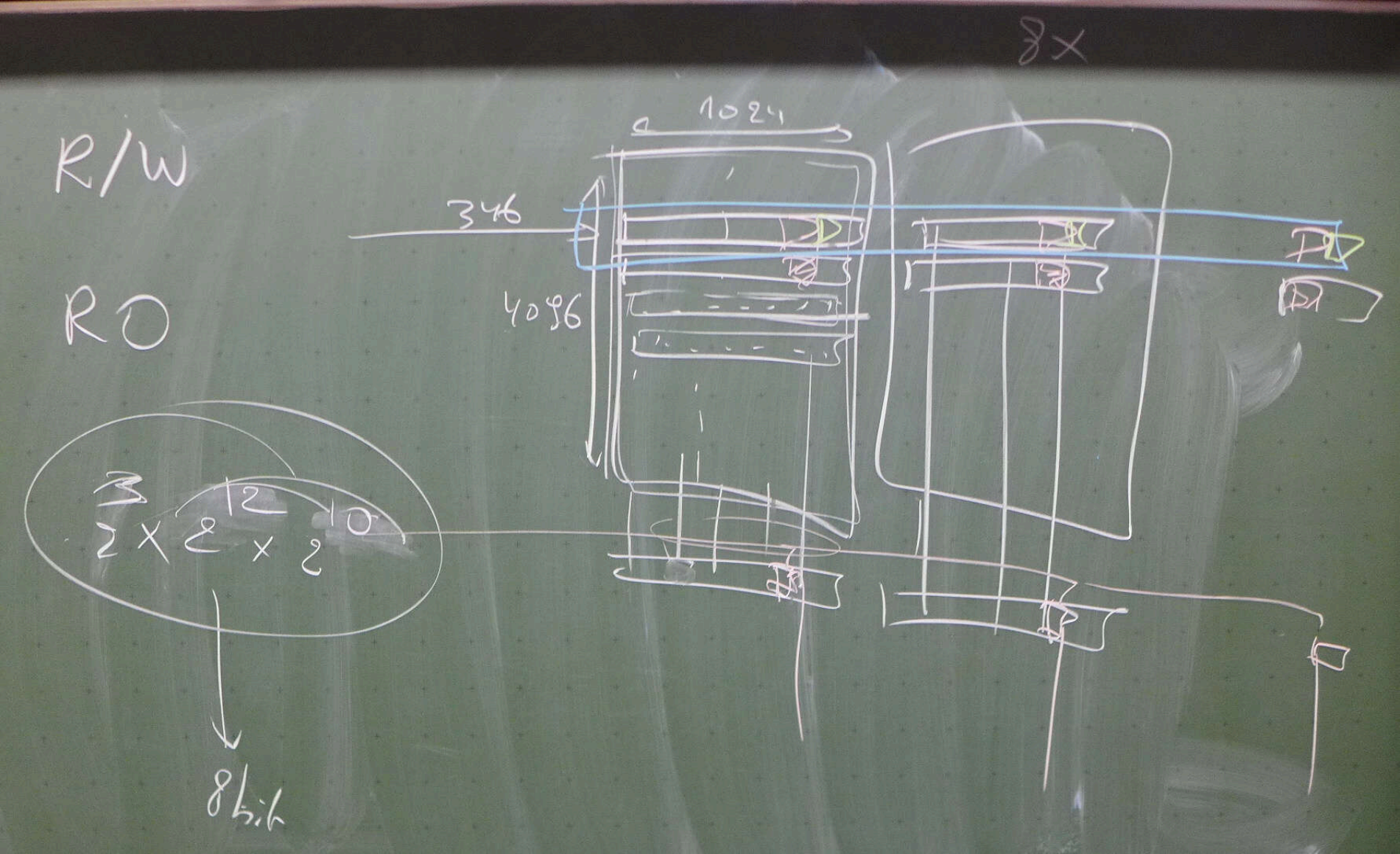

IMM
REG
ADD
SUB
OR

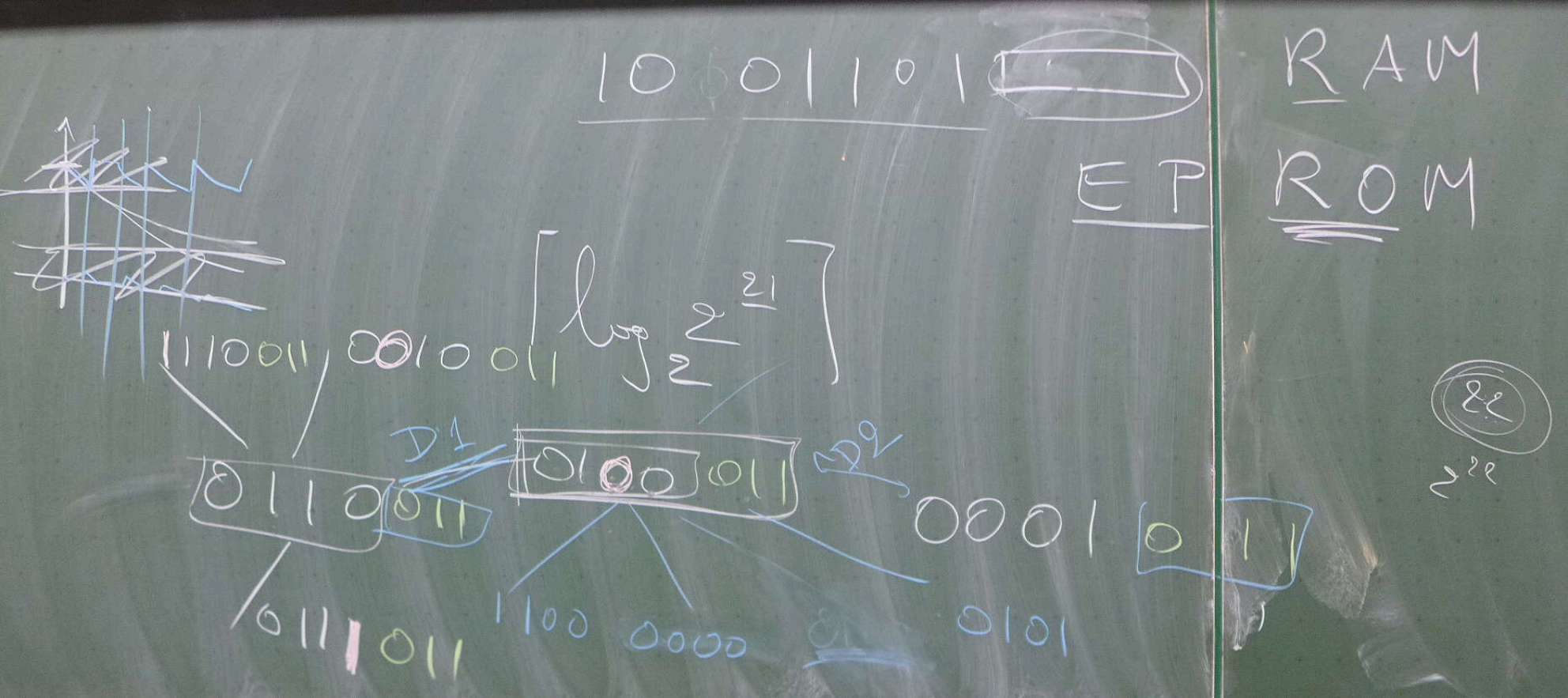

S
+5V
0
-5V
1

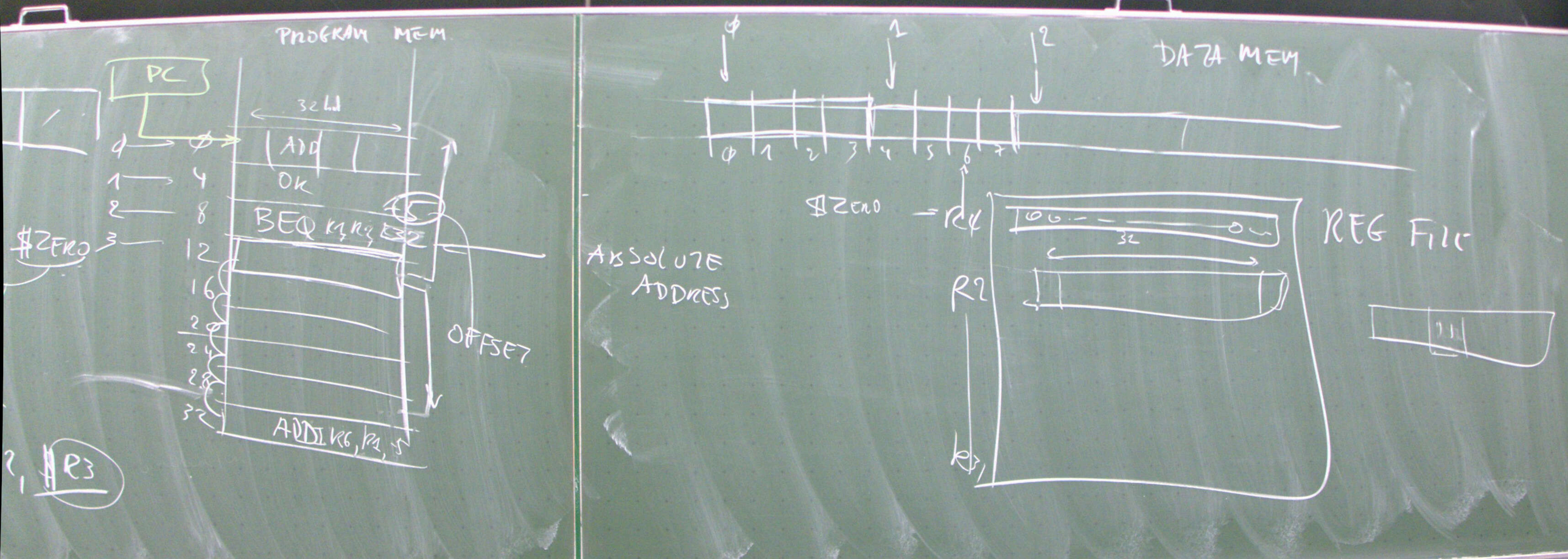
N $\longrightarrow$ $b = \lceil \log_2 N \rceil$




RAM R/W
EP ROM RO


8x
1024
R/W
346
4096
RO
3
12
10
2 x 2 x 2
8bit


10 0 1 1 0 1
RAM
EP
ROM
$\lceil \log_2 2^{21} \rceil$
1110011 0010011
D1
0110 011
0100 011
0001 01
0111 011
1100 0000
0101
2^{22}

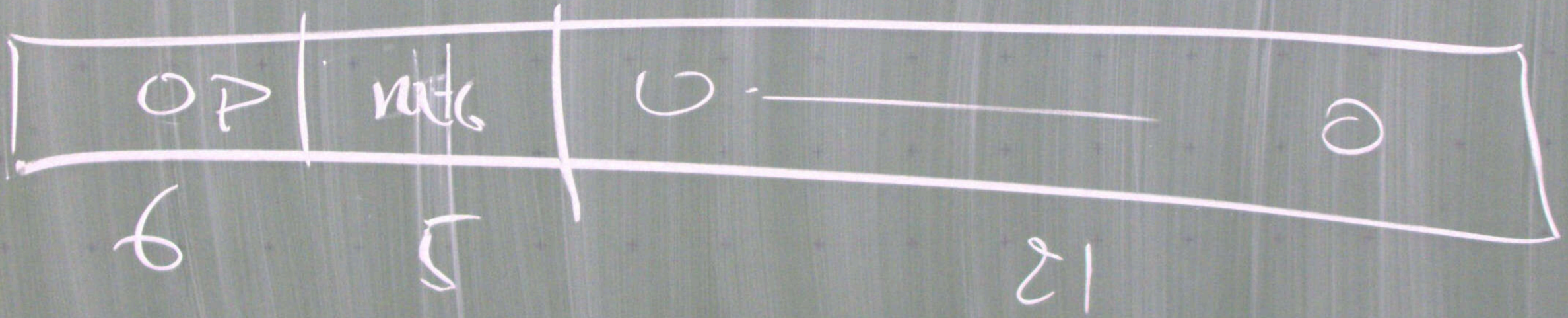

PROGRAM MEM
PC
32 bit
ADD
OK
BEQ R1, R2, L2
OFFSET
ADDI R6, R1, 5
$ZERO
R3
0
1
2
3
4
8
12
16
20
24
28
32
DATA MEM
$ZERO
R4
R2
R3
32
REG File
11
ABSOLUTE ADDRESS

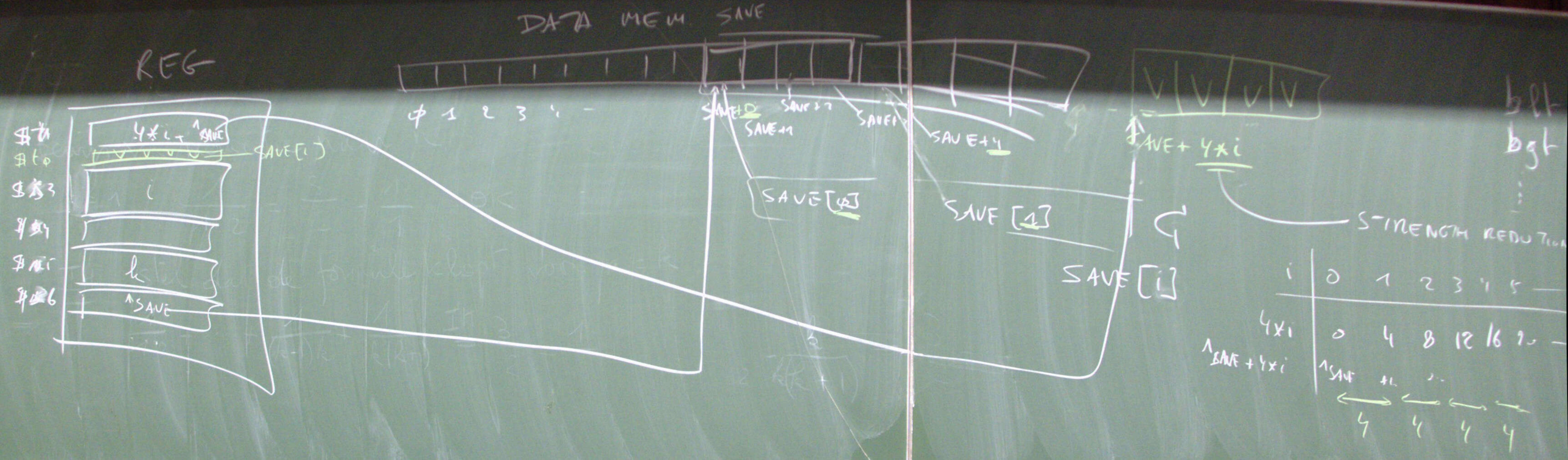
A. AND $R2, $R2, $ZERO

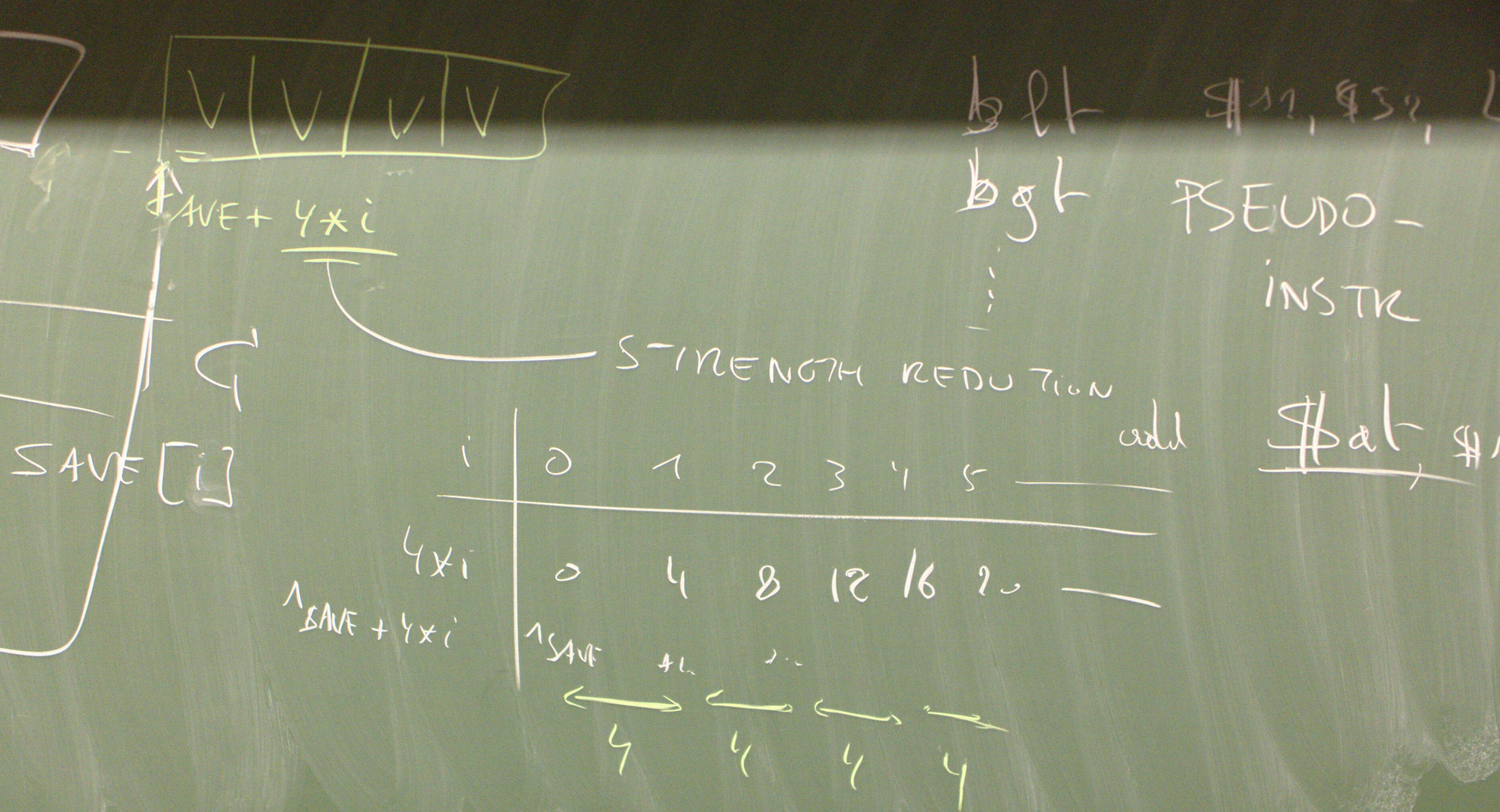
B. NOT $R3, $R2

AND $R2, $R2, $R3

SET $R2, 0




DATA MEM SAVE
REG
4*i + ^SAVE
SAVE[i]
i
k
^SAVE
SAVE+4
SAVE+4*i
SAVE[i]
STRENGTH REDUCTION
i
0 1 2 3 4 5
4*i
0 4 8 12 16
^SAVE + 4*i
4 4 4 4


V | V | V | V
SAVE + 4*i
SAVE[i]
STRENGTH REDUCTION
bgt
PSEUDO-INSTR
addi
$at,
i | 0 1 2 3 4 5
4*i | 0 4 8 12 16 20
SAVE + 4*i
SAVE
4 4 4 4

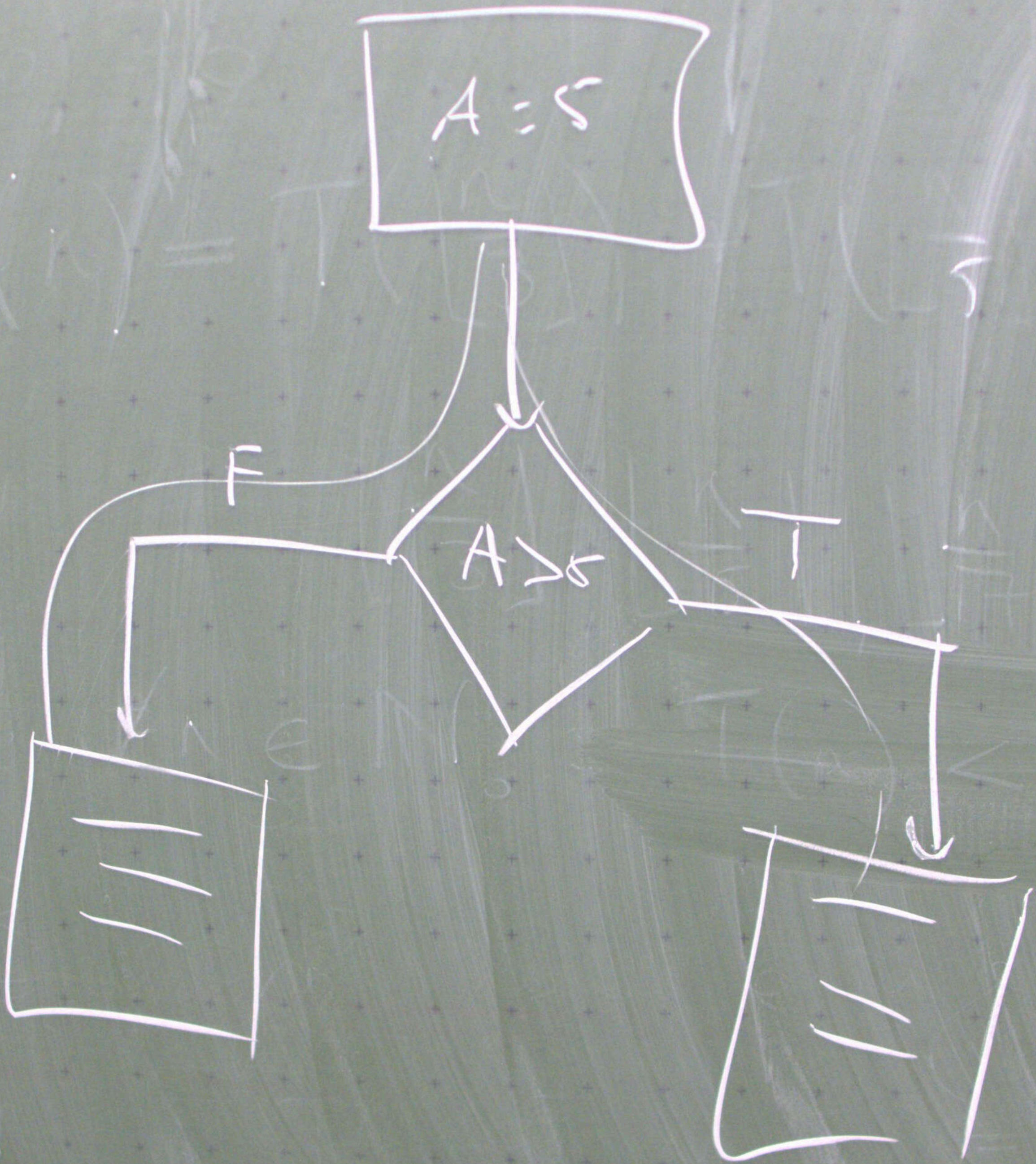
bll Φ_1, Φ_2, Γ

bgl PSEUDO-

INSTR

REDUCTION

add Φ_0, Φ_1, Φ_2


A:=5
F
A>5
T

i = 0
j = -1

if (i / 7 == j+1) BB1 ;

else BB2 ;

i = a + Lxc

BOEHM & JACOPINI

① SEQ

② IF

③ LOOP

PSEUDO-INSTRUCTIE

BLT $s1, $s2, L

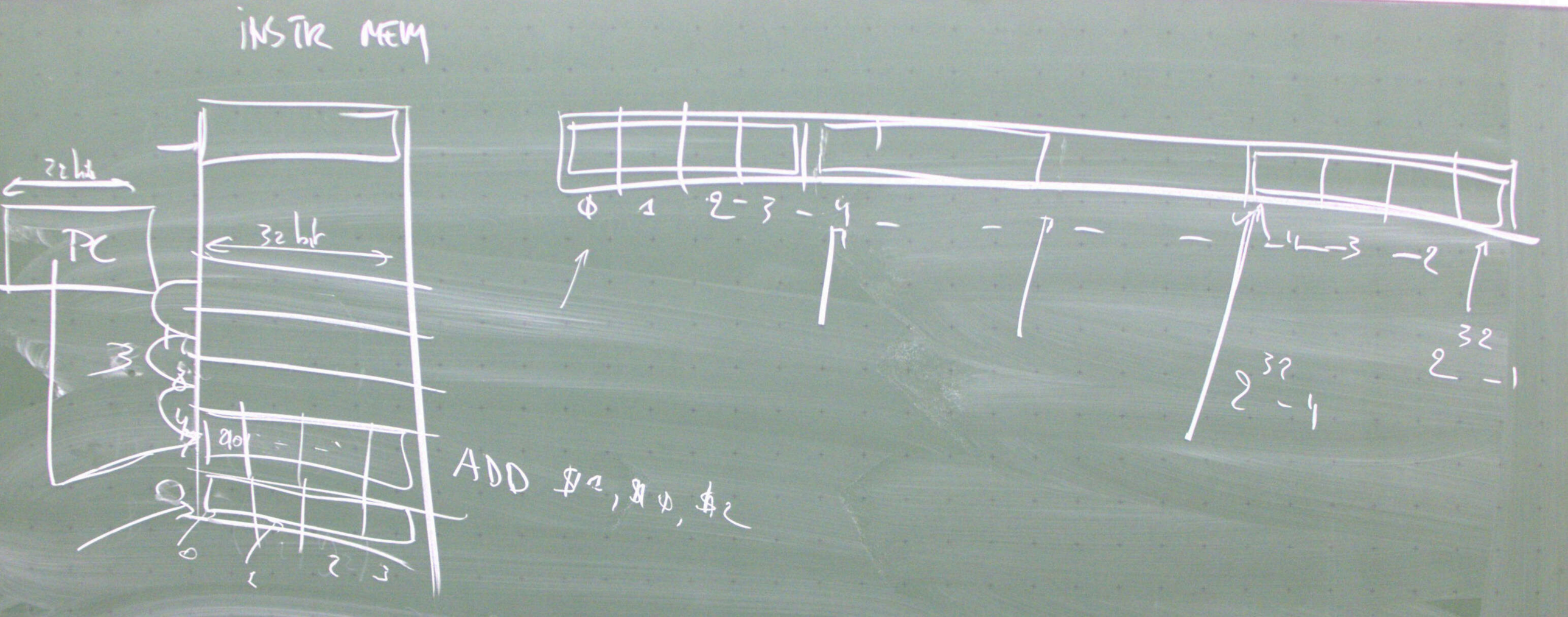
→ SLT $s1, $s2, $t4

→ BNE $t4, $zero, L

1 SA

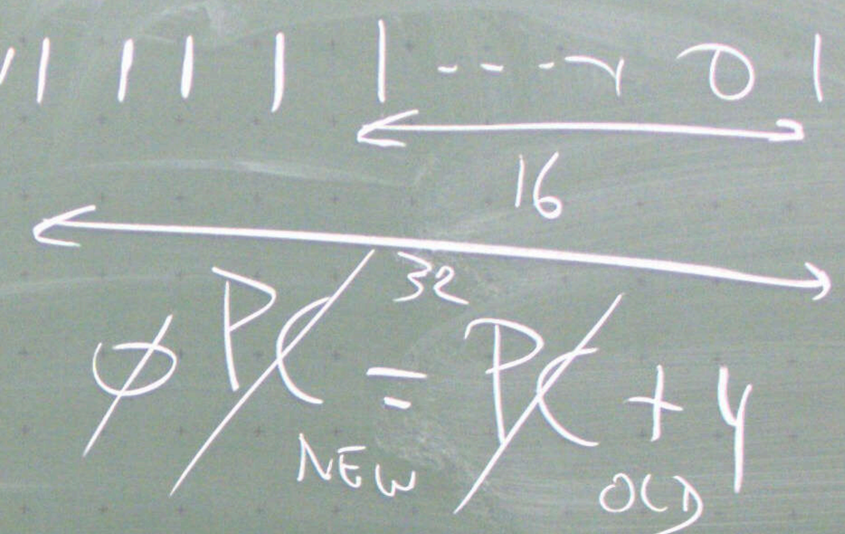
1 ϕ

S

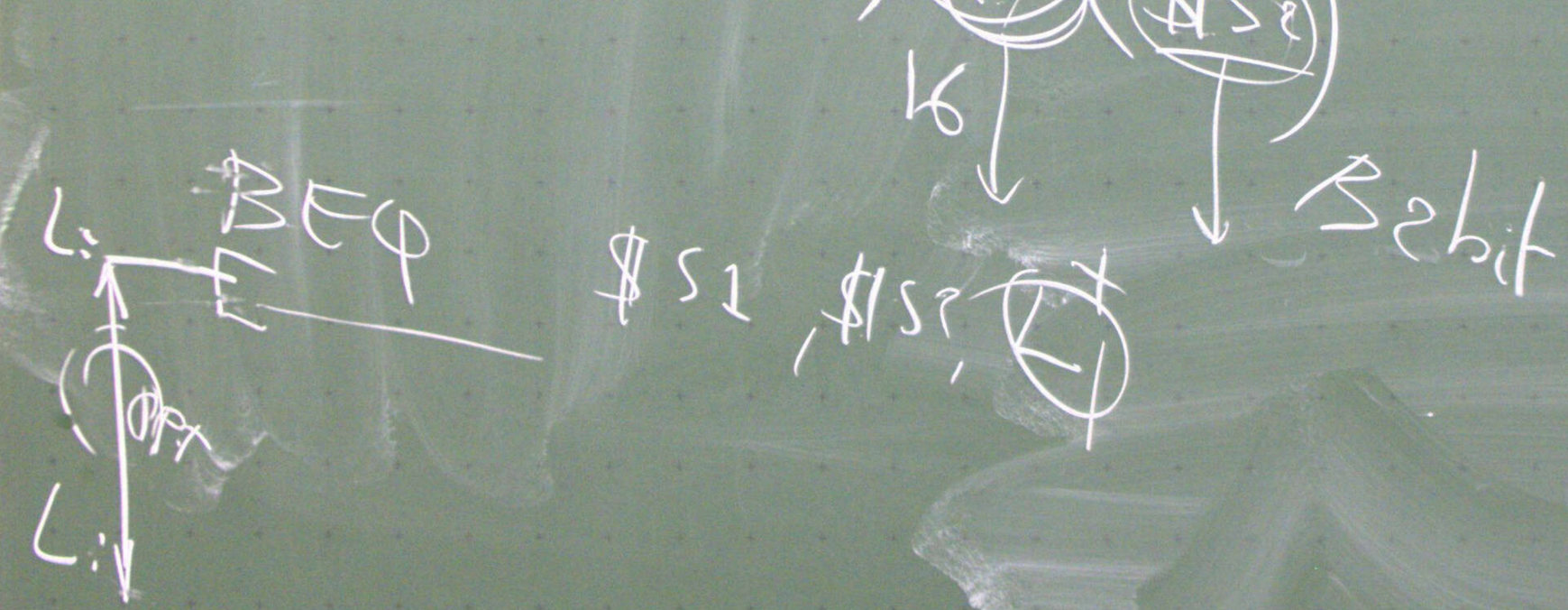

INSTR MEM
32 bit
PC
32 bit
ADD $d, $s, $t
0
1
2
3
4
32
2 - 1
32
2 - 4

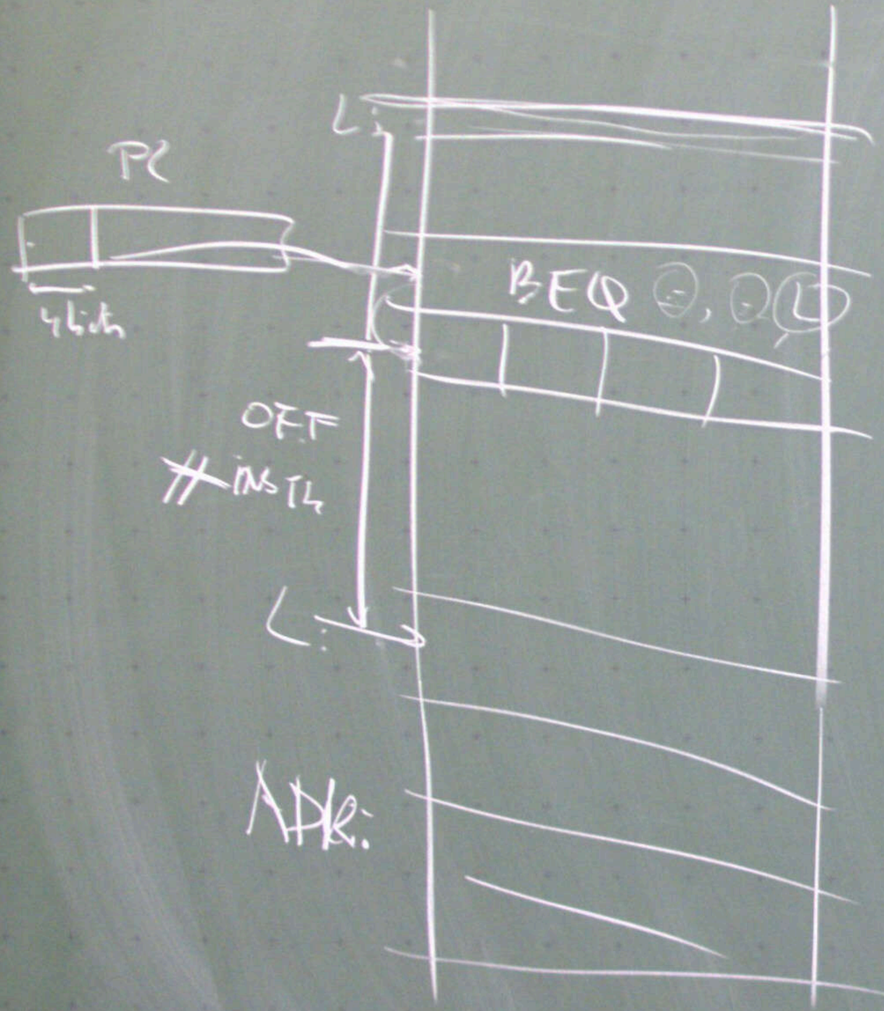
ADD ; es, (12), (13)

LW $s1, 5($s2)



ST $s1, (12)($s2)

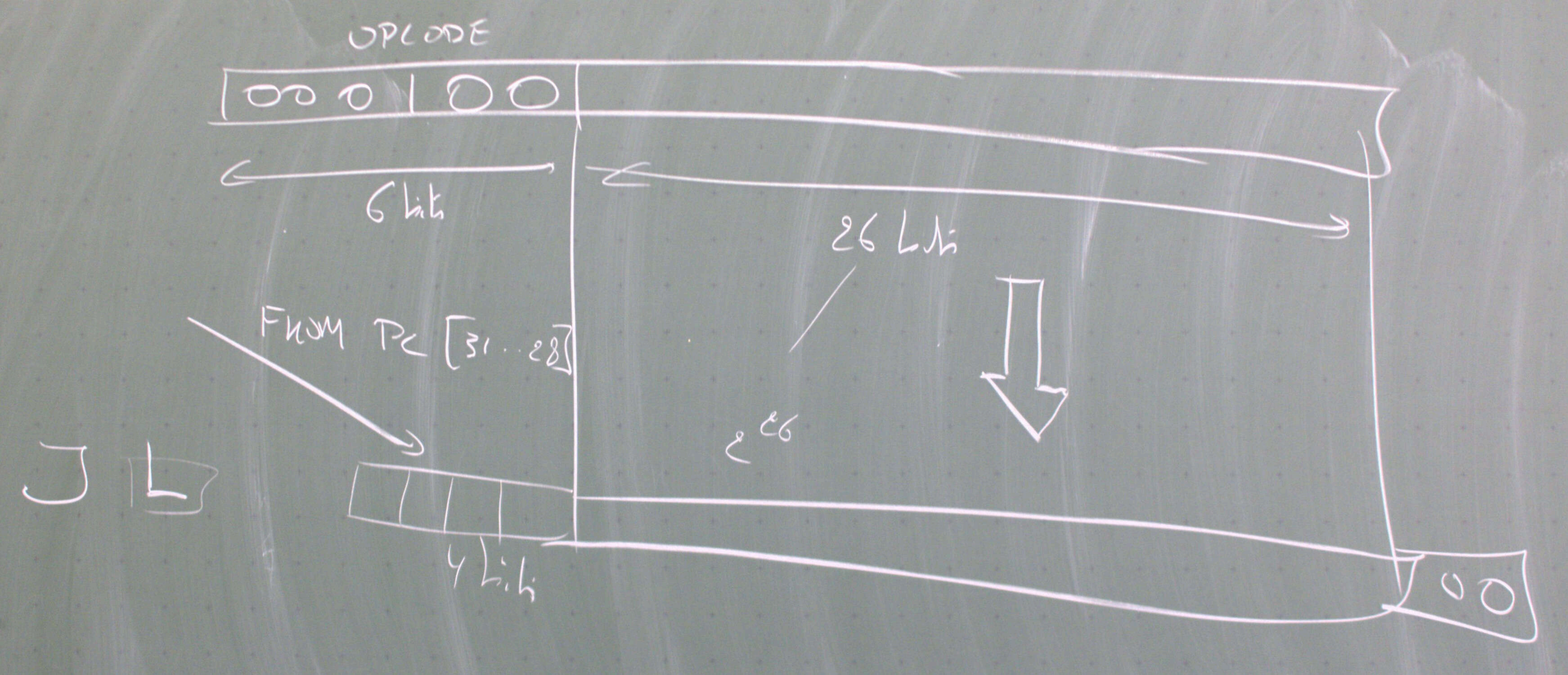




LW $S1, OFF ($S1)

ST $S1, OFF ($S1)

J ADR


OPCODE
6 bits
26 bits
FROM PC [31..28]
4 bits
00

4 28

2^{28} BYTES

2^{26} INSTR

$0 \rightarrow (2^{26}-1)$

$2^{28}-1$

2^{26} INSTR

MEM # BLOCKS

2^4

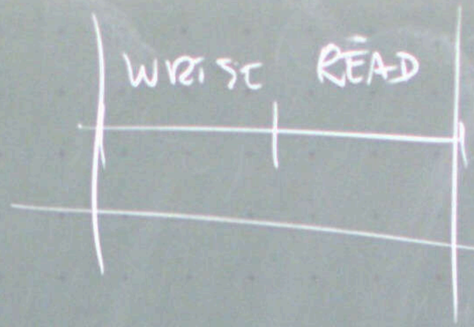
BEQ. ~ PC

JR $S1 —

INDIRECT ADDRESSING

LW

ST



INSTN MEM

DATA MEM

SWITCH (VAR)

CASE 0

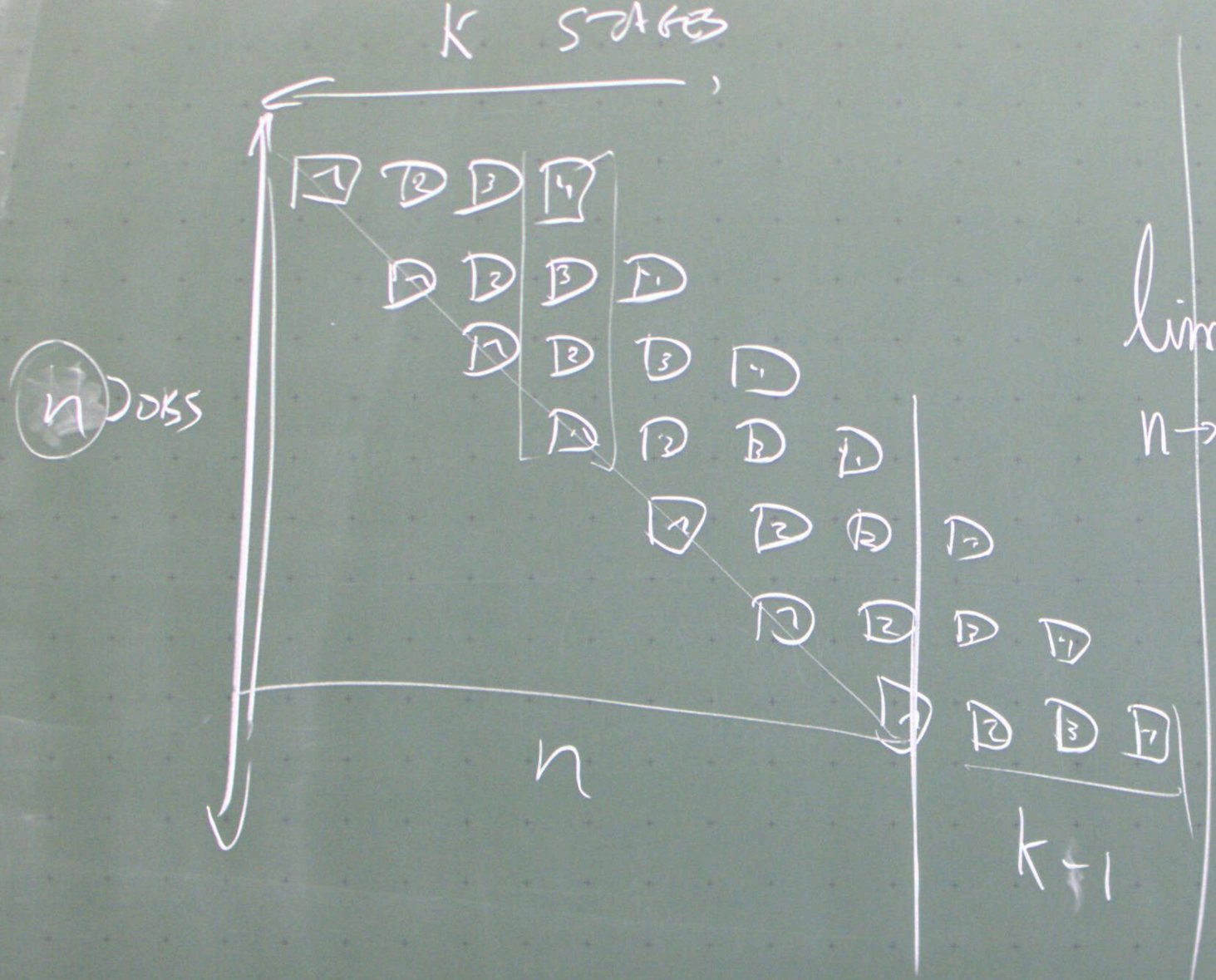
CASE 1

CASE 2

PIPELINING (ILP)

- LATENCY ≡ × 1
- THROUGHPUT ↗ × K

S



$$\lim_{n \to +\infty} \frac{n K}{\left(\frac{n}{n} + \left(\frac{K-1}{n}\right)\right) K} = K$$

$\frac{n}{n} \to 1$; $\frac{K-1}{n} \to 0$

MEMORY LEAK

FUNCTIONS

PARAMS WHERE CALLEE CAN FIND THEM

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

16 bit 2's COMPLEMENT

ST $t0, OFFSET ($sp)
LW $t0, OFFSET ($gr)

RETURN VALUE(S)

- REGISTERS $v0, $v1
- DATA MEM (STACK)
- DATA MEM (GLOBAL)

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

2^{16}

UNIX: SYS V

BSD

MACH

LINUX

OS X

SYSCALL

PARAMS

- %eax ....

TYPE OF SERVICE (OS)

- %V%

MEMORY (R/W)

- REGISTER
- DATA
  - STACK
  - HEAP
  - STATIC
- PROGRAM — TEXT
- MEMORY MAPPED I/O

$t7 x

TION

$SP

MU

STACK

DATA MEM

PUSH $t4

ST $t0, 0 ($SP)

ADDIU $SP, $SP, -4

SUBIU $SP, $SP, 4

FIRST FREE SPACE ON STACK ($SP)

OFFSET: $-2^{15} \rightarrow 2^{15}-1$

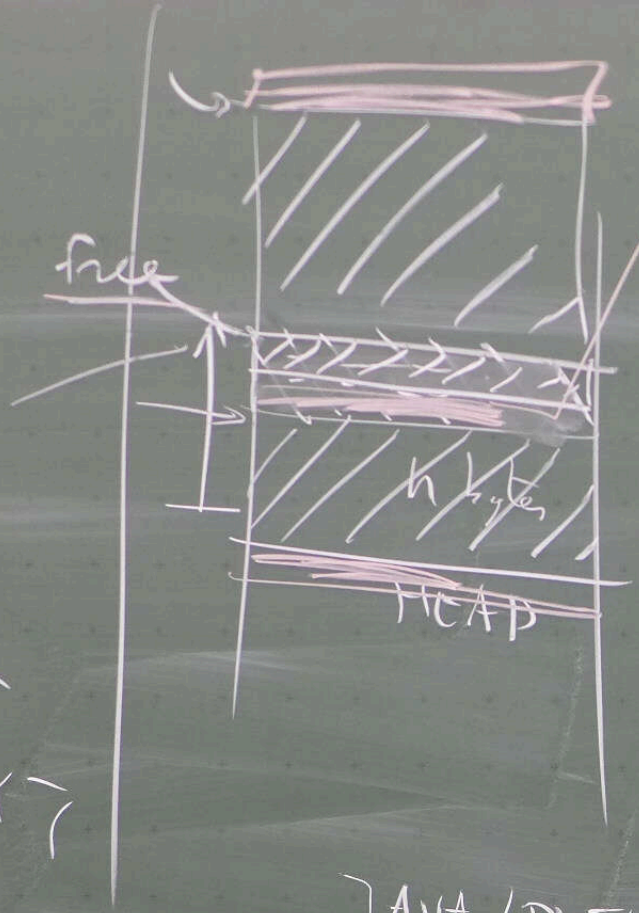
ST $t4, -4 ($SP)

SUBIU $SP, $SP, 4

ST $t0, 0 ($SP)

($SP) LAST OCCUPIED SPACE ON STACK

R/W)



$t0 [x]

TOS

$SP [ ]

malloc(n)

- STACK
- HEAP
- STATIC

— TEXT

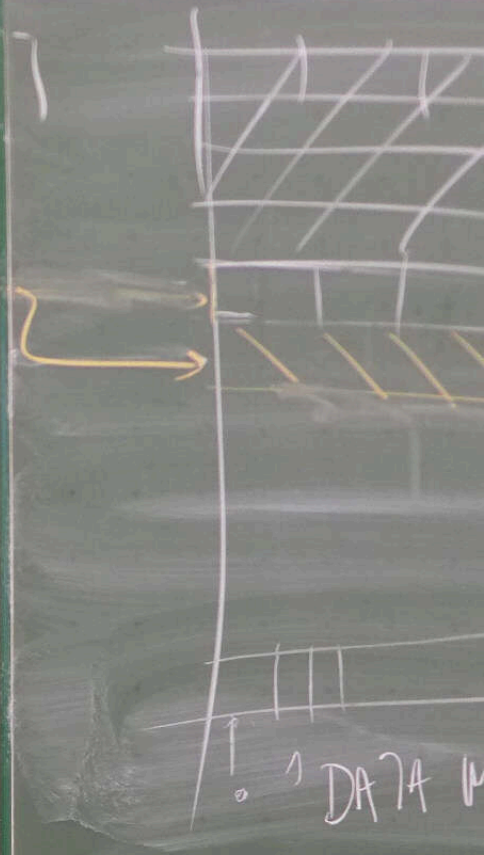
ED I/O

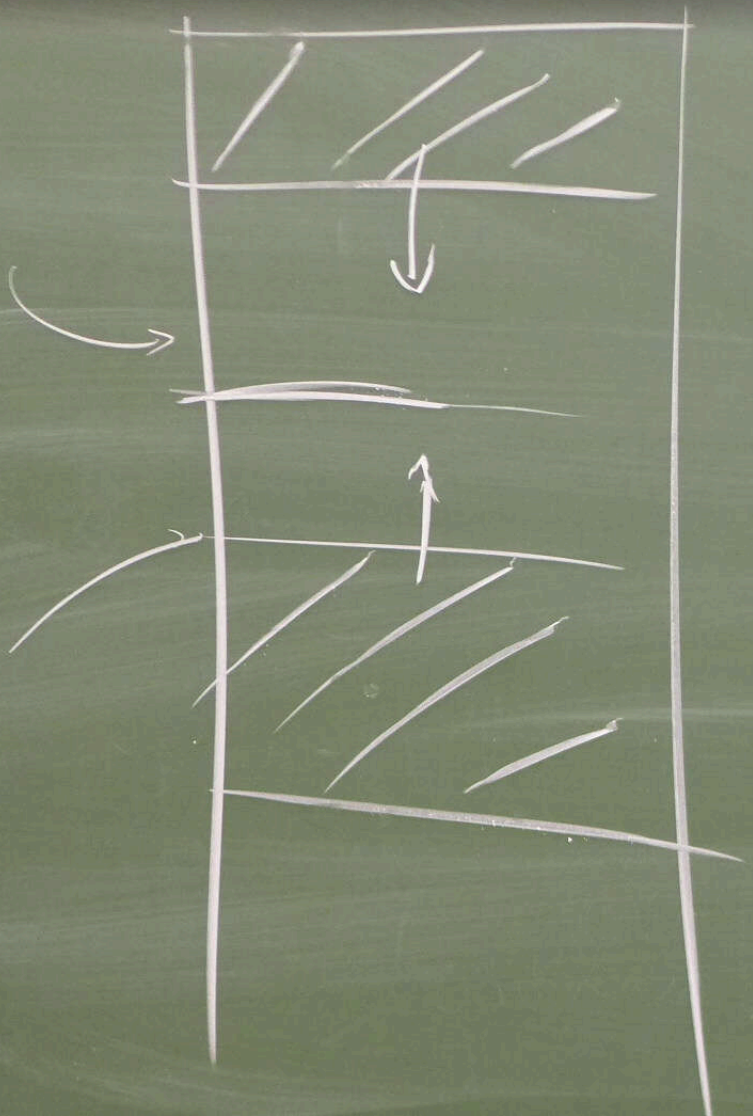
JAVA / PYTHON

MALLOC, ~~FREE~~

"GARBAGE COLLECTION"

MMU



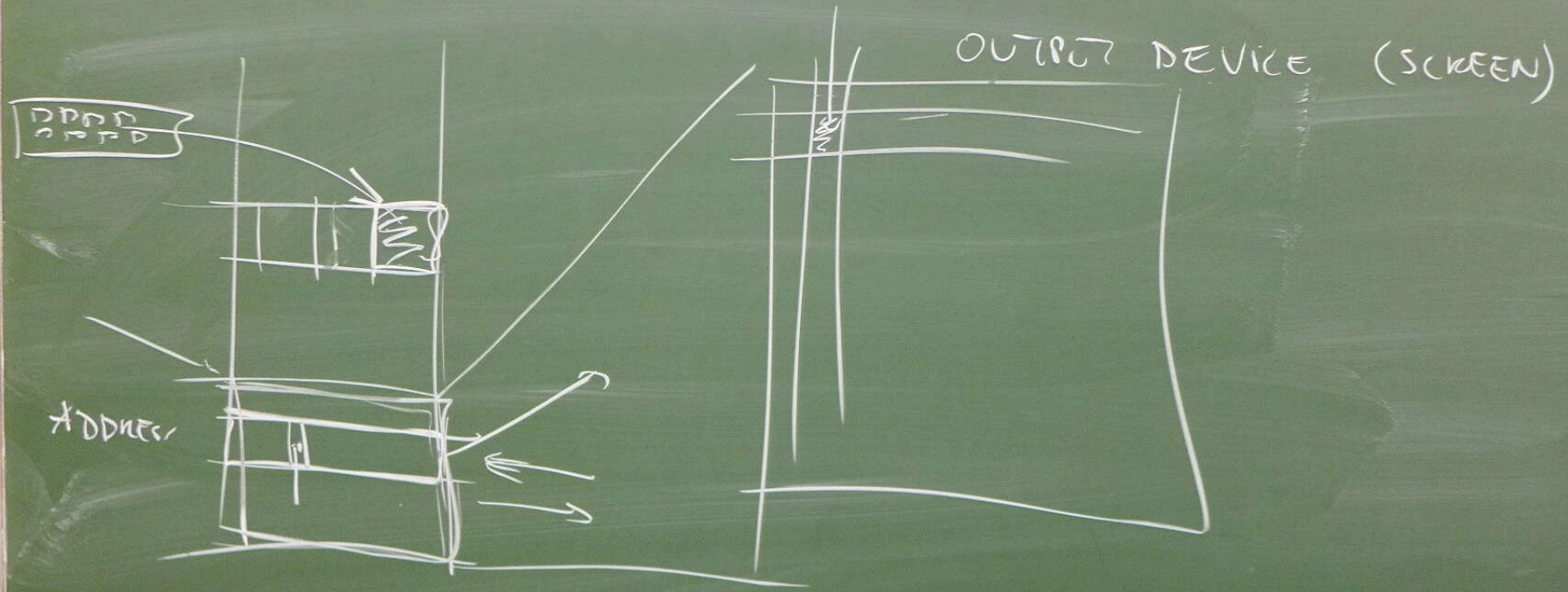


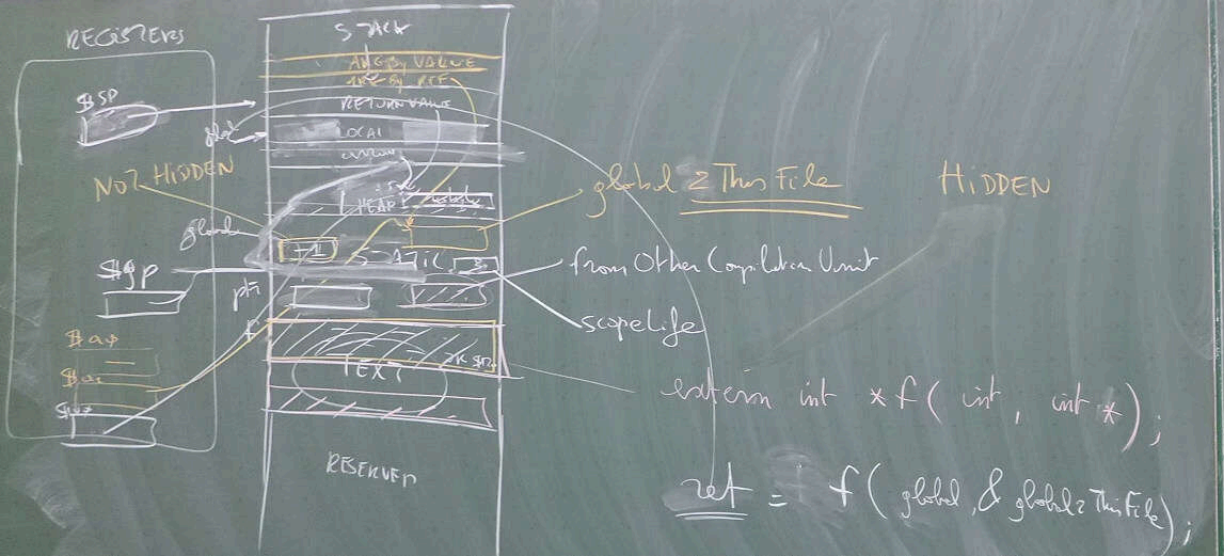
STACK-INTENSIVE
(RECURSIE)

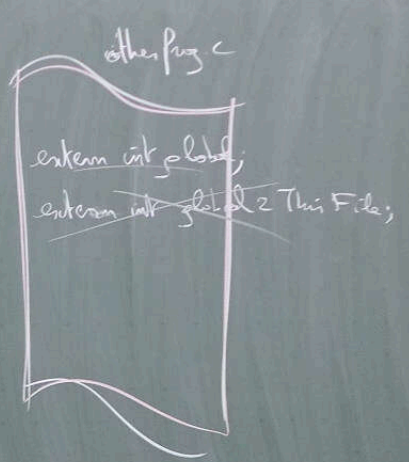
HEAP-INTENSIVE

RESERVED:

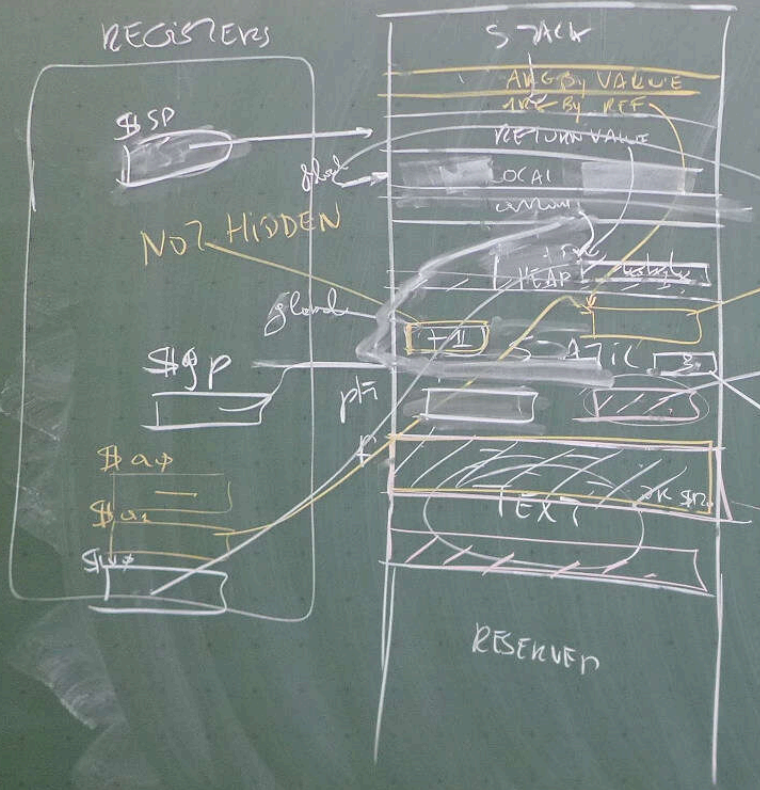
eg: MEMORY MAPPED I/O







NAMESPACE POLLUTION



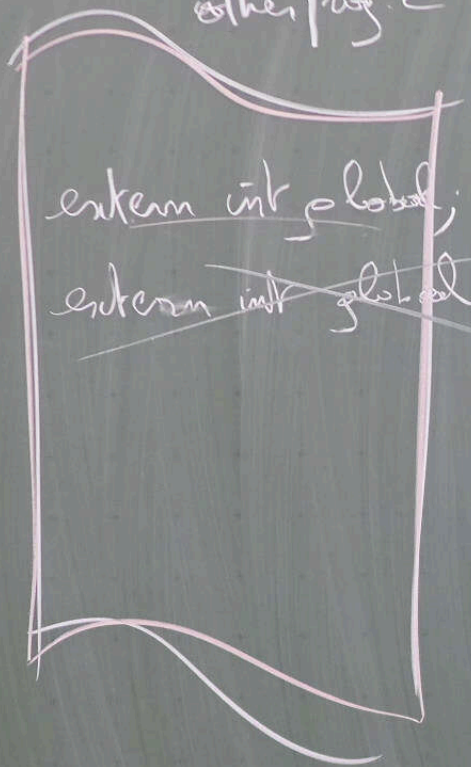
global2ThisFile — HIDDEN

from Other Compilation Unit

scope life

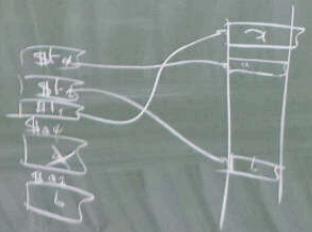
extern int *f(int, int *);

ret = f(global, &global2ThisFile);



NAMESPACE POLLUTION

LW $a0, ∅(sta)
LW $a1, ∅($t2)
res = f(a, b) - - - - - - JAL $RA

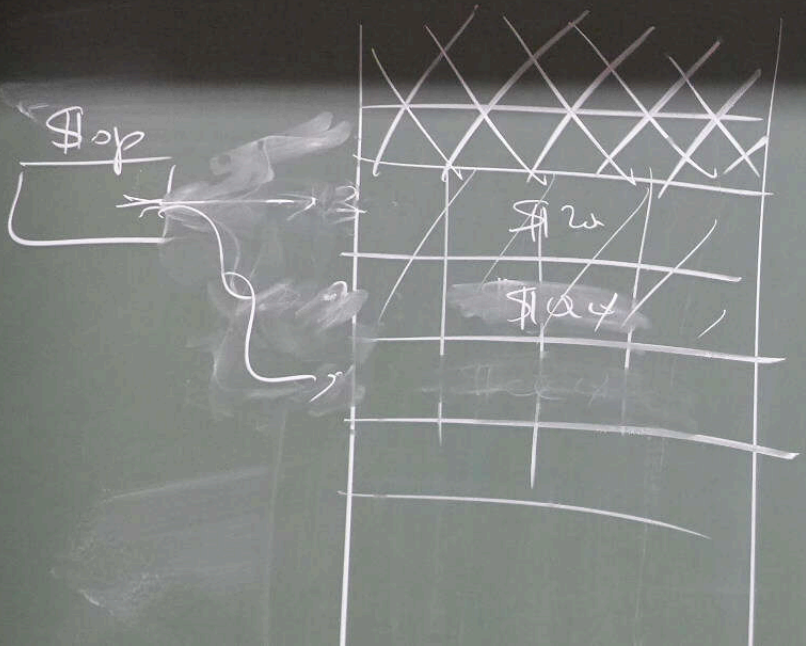


SAVE $a0 SW $a0, ∅($sp)
PUSH < SUBIU $sp, $sp, 4
LW $a0, ∅($t2)

f(- , -)
{
(g(x);
return
}

PUSH | SW $RA, ∅($sp)
SUBIU $sp, $sp, 4
SAVE-$RA
JAL $RA
RESTORE $RA
POP ADDIU $sp, $sp, 4
LW $RA, ∅($sp)
JR $RA
RESTORE $a0
POP

FOUT ⇒ STACK DE-SYNC




```
SUBIU   $SP, $sp, 8
SW      $RA, 8($sp)
SW      $a4, 4($sp)
```

SAVE
RESTORE

SEPARATE

COST PUSH / POP
#INSTR

$2 \times (2 + N)$

PER FCT

N VALUES TO SAVE

COST EFFICIENT
#INSTR

$2 \times (N + 1)$

STACK SIZE

FACT (n)

$2(n+1)$ bytes $\times 4$

fact(2)

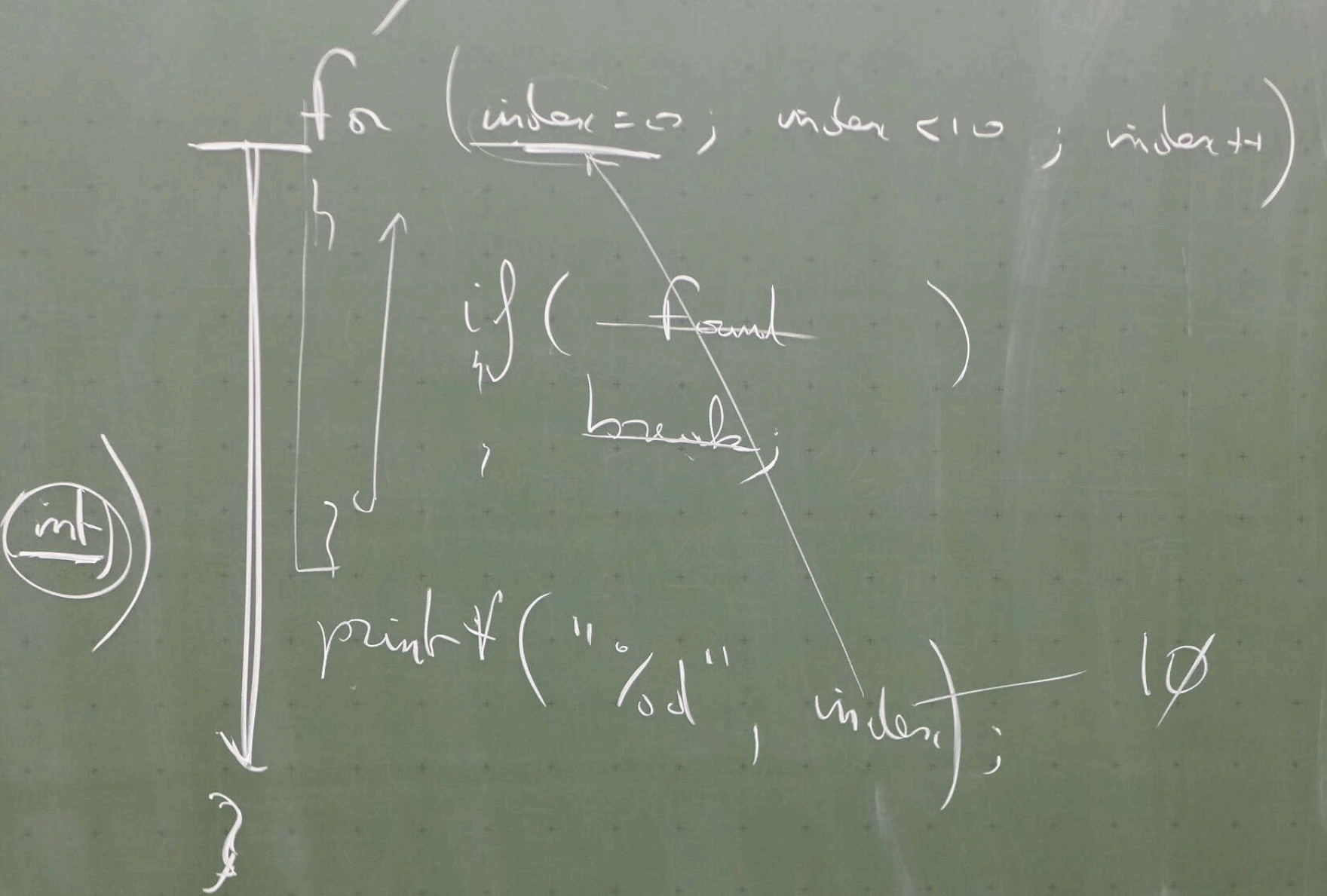
2

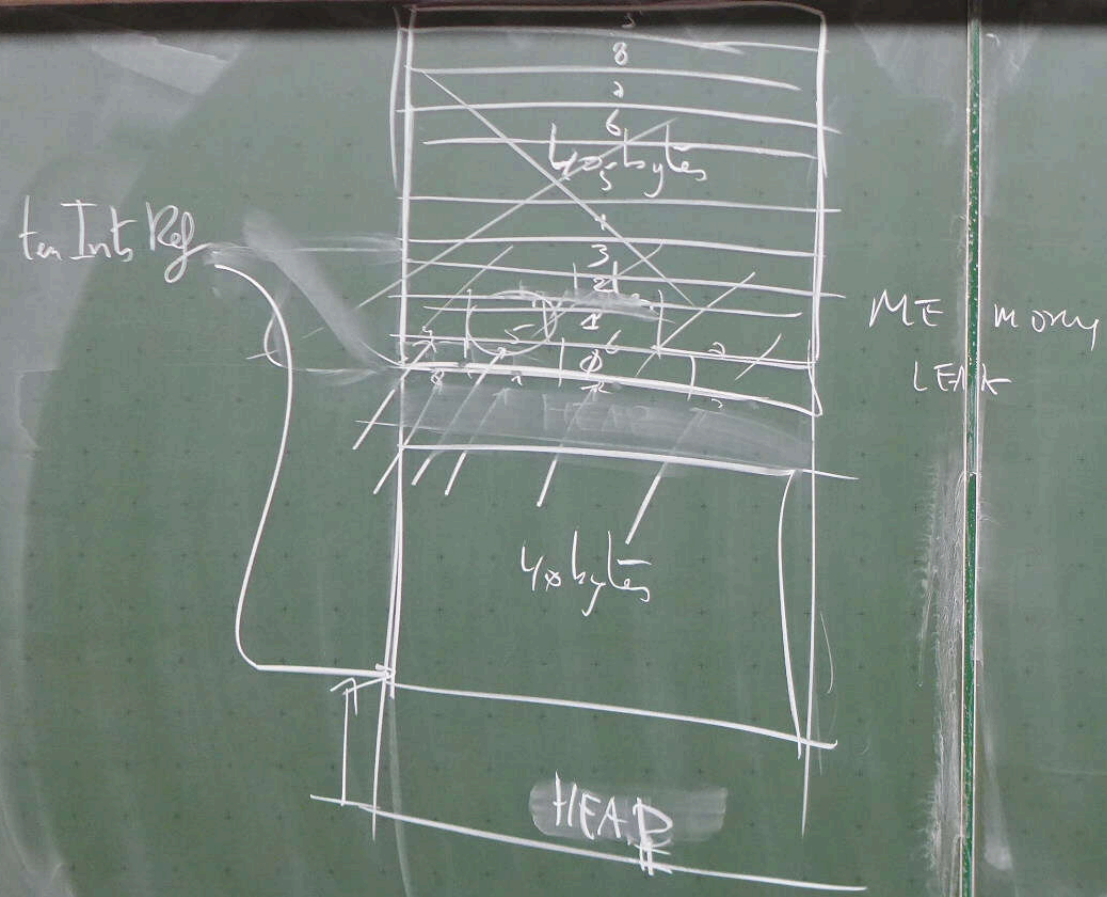
fact(1)

2

fact(0)

2

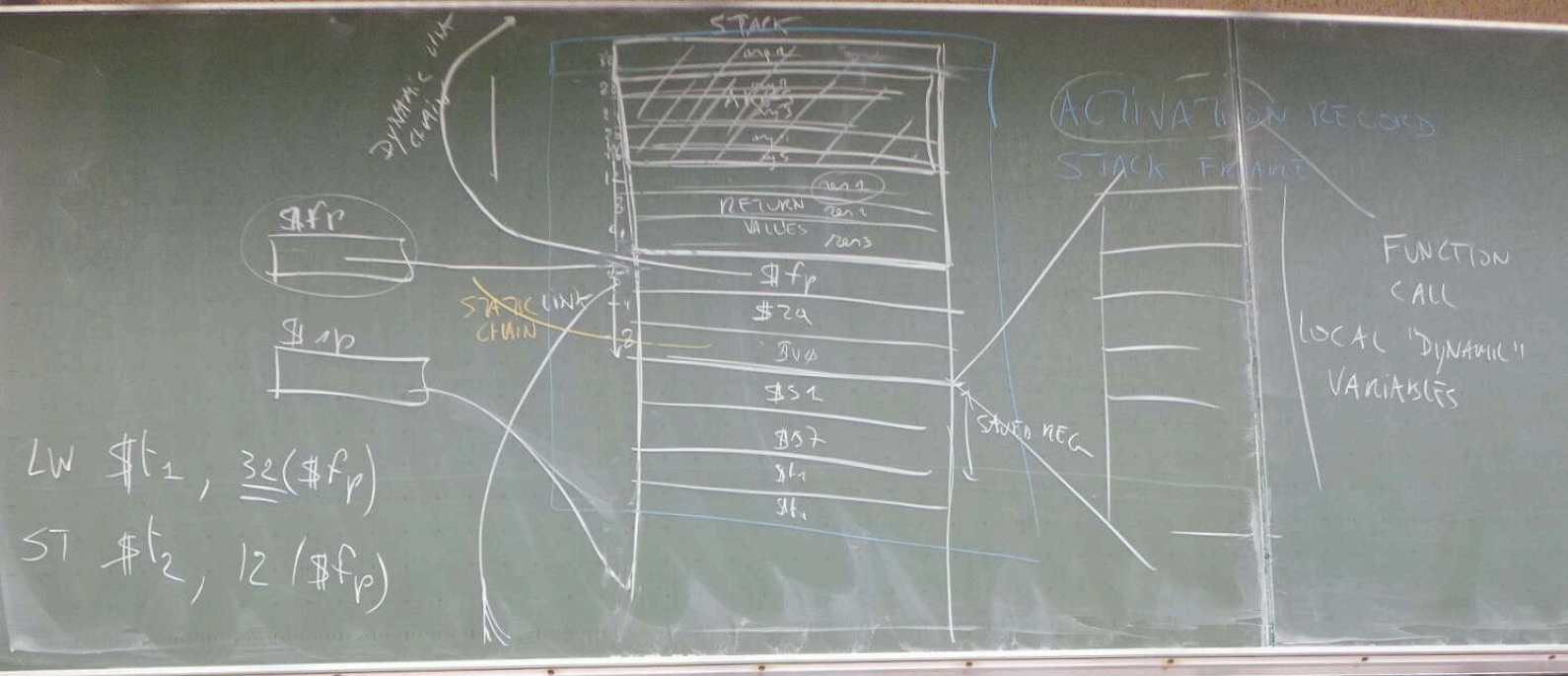

for (index=0; index<10; index++)
{
if(found)
bomb;
}
printf("%d", index);
10
int

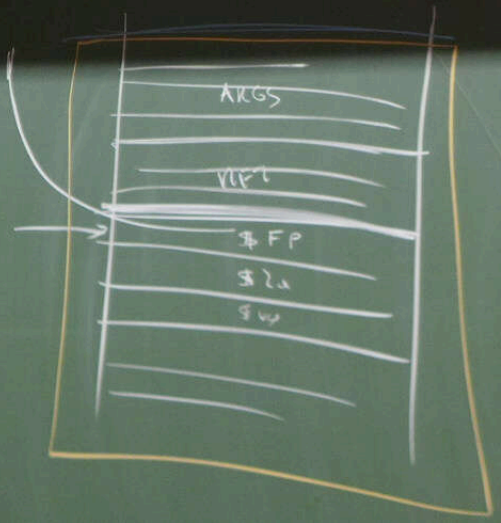


tenIntsRef [5]

5

*((tenIntsRef + 5 * sizeof(int))


STACK
DYNAMIC LINK CHAIN
ACTIVATION RECORD
STACK FRAME
RETURN VALUES
$fp
$ra
$v1
$s1
$s7
$t1
$t0
STATIC LINK CHAIN
$fp
$sp
SAVED REG
FUNCTION CALL
LOCAL "DYNAMIC" VARIABLES
LW $t1, 32($fp)
ST $t2, 12($fp)


ARGS
VLP
$FP

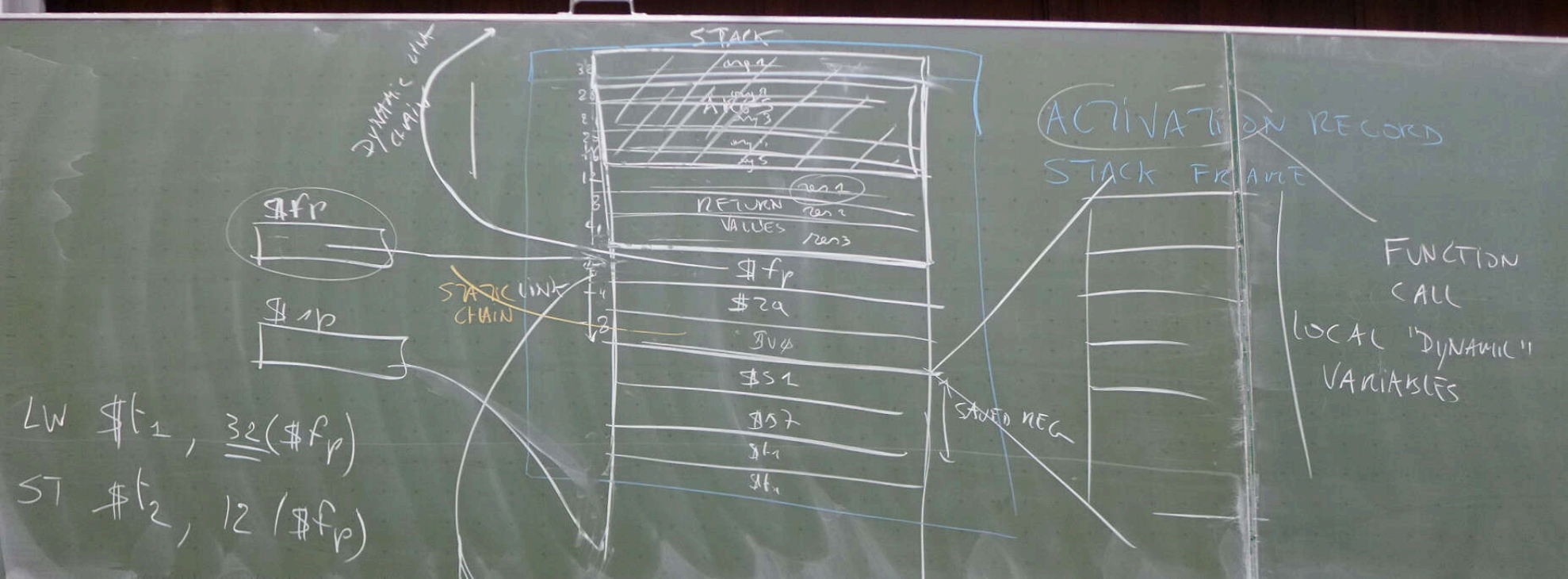
PROBLEM WITH $U64 — $G43, $UVF, $VS

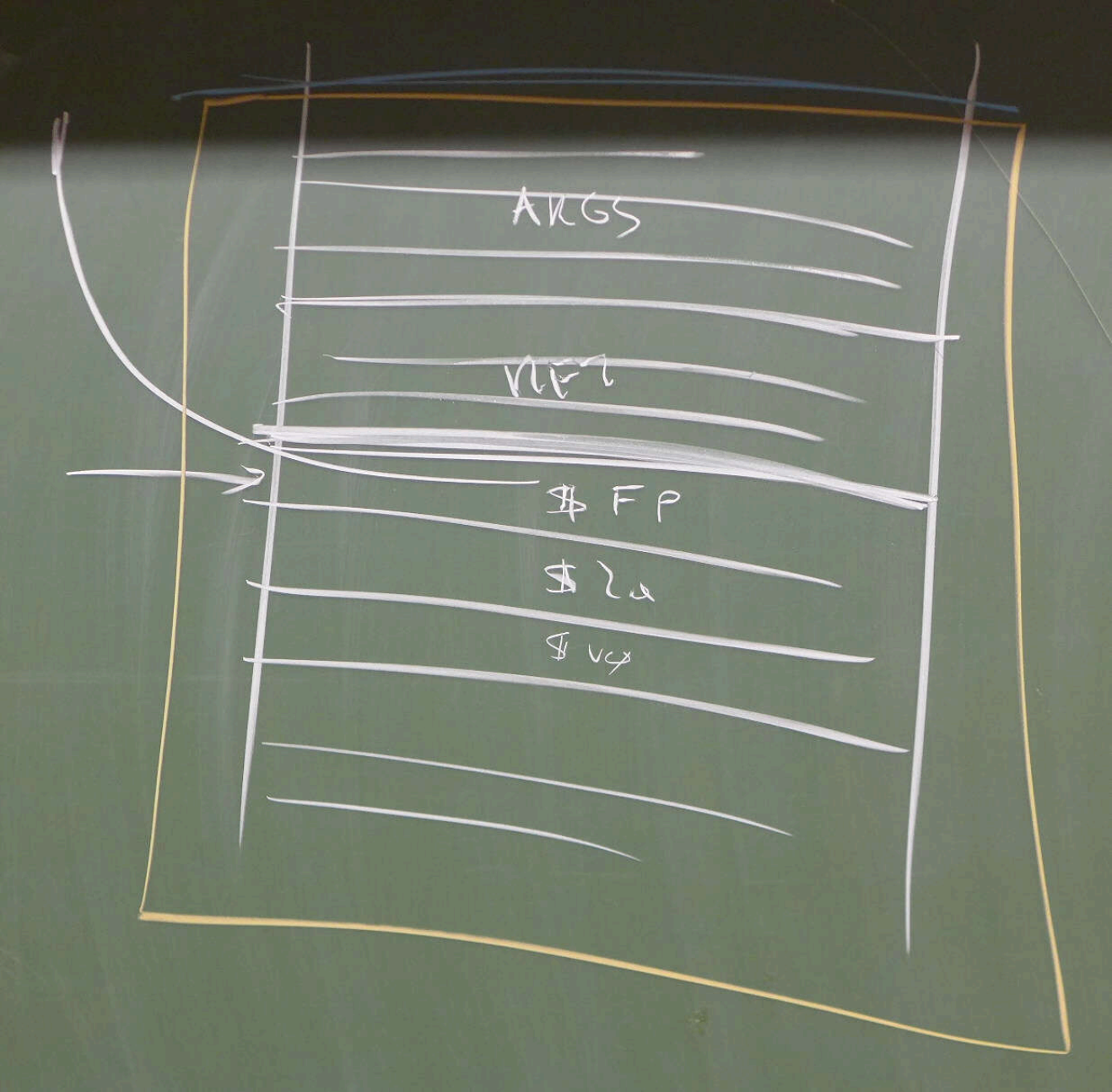
- # ARGS, # RET VALUES
  > 4 > 2
- SIZE OF ARGS, RET VALUES > 4 bytes

SOLUTION

① PASS REFERENCE TO STRUCTURE
OK FOR LEAF, NOK FOR NON-LEAF

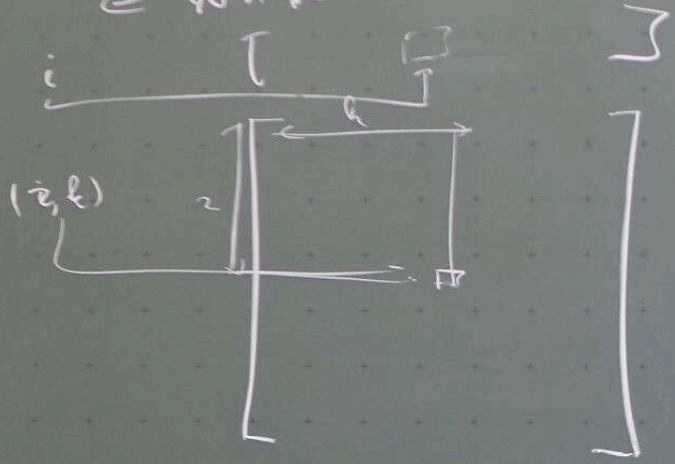
② PUSH ON STACK


STACK
DYNAMIC LINK CHAIN
$fp
$ra
STATIC LINK CHAIN
RETURN VALUES
$fp
$s0
$s1
$s7
ACTIVATION RECORD
STACK FRAME
FUNCTION CALL
LOCAL "DYNAMIC" VARIABLES
SAVED REG
LW $t1, 32($fp)
ST $t2, 12($fp)


ARGS
$FP

index $\in \mathbb{N}$

$\in \mathbb{N} \times \mathbb{N}$



T

$i \longrightarrow T$

v[i]

Vector
Matrix —— Array #DIM - - -

ELEMENT TYPE == $\forall$ elements

SIZE (ELEMENT TYPE)

ADDRESS (BASE ADDRESS)

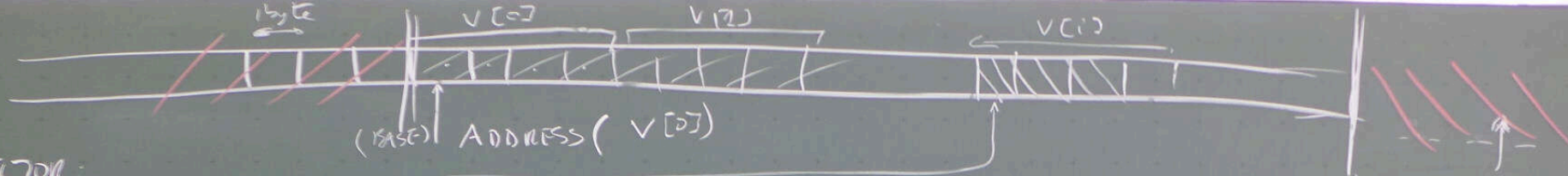
SIZE OF THE VECTOR, MATRIX ARRAY

DIMENSION

5 x 2 MATRIX OF INT

SIZE: 10 x SIZE(ELEMENT TYPE)

DIMENSION: 2



VECTOR

ADDRESS (V[i]) = ADDRESS(V[0]) + i x SIZE (ELEMENT TYPE)

int v[];

int v[10];

int *v;

v = 10 * malloc (sizeof (int))

SIZE (V) = LENGTH x SIZE (ELEMENT TYPE)

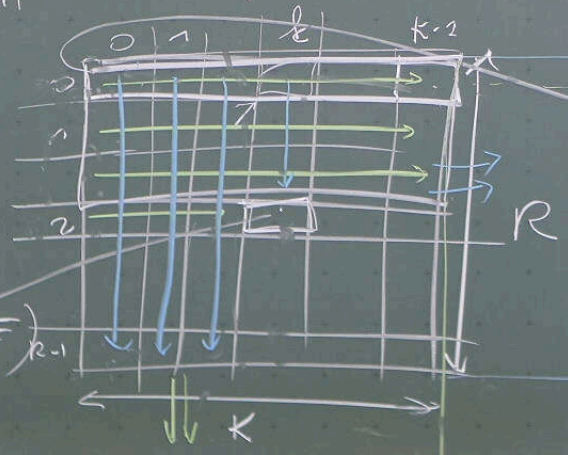
LENGTH

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

JAVA

MATRIX

ARRAY 2D



ADDRESS (a[r,k])

ROWS = R

COLUMNS = K

SIZE (ELEMENT TYPE)

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

SIZE (a) = R x K x SIZE (ELEMENT TYPE)

MATRIX

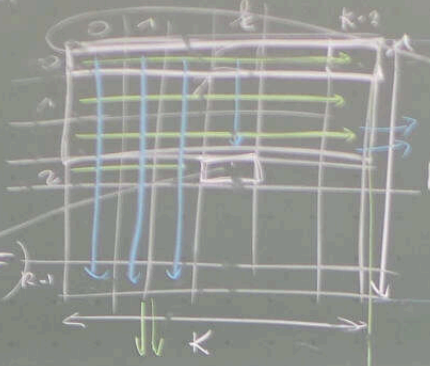
ARRAY 2D

ADDRESS(a[i,k])

ROWS = R

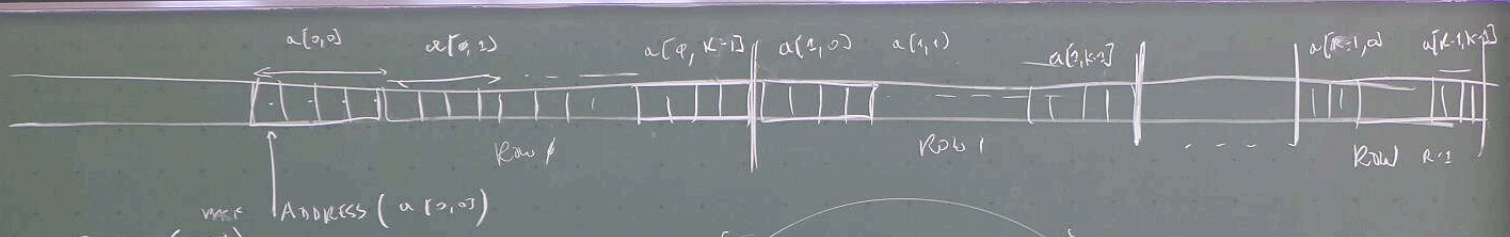
COLUMNS = K

SIZE (ELEMENT TYPE)



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

SIZE (a) = R × K × SIZE (ELEMENT TYPE)



ROW-MAJOR (C)

ADDRESS (a[i,k]) = ADDRESS (a[0,0]) + (k + i × K) × SIZE (ELEMENT TYPE)

(0 ≤ i ≤ R)

0 ≤ k < K

CAVEAT: malloc

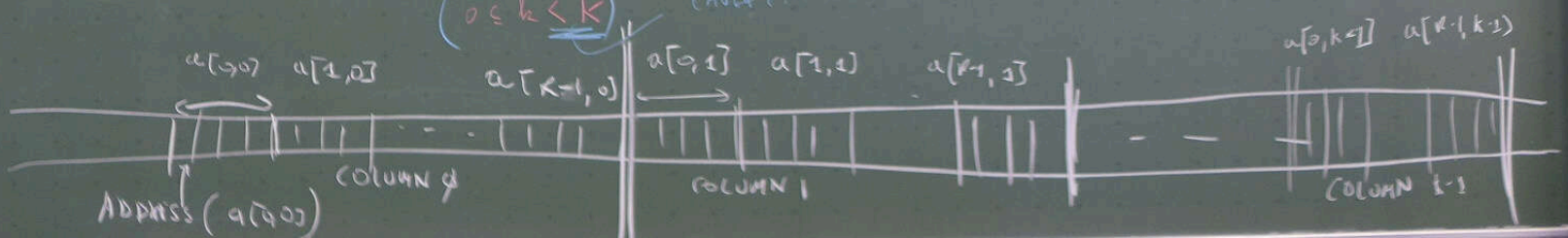
COLUMN-MAJOR (FORTRAN)

ADDRESS (a[i,k]) = ADDRESS (a[0,0]) + (i + k × R) × SIZE (ELEMENT TYPE)

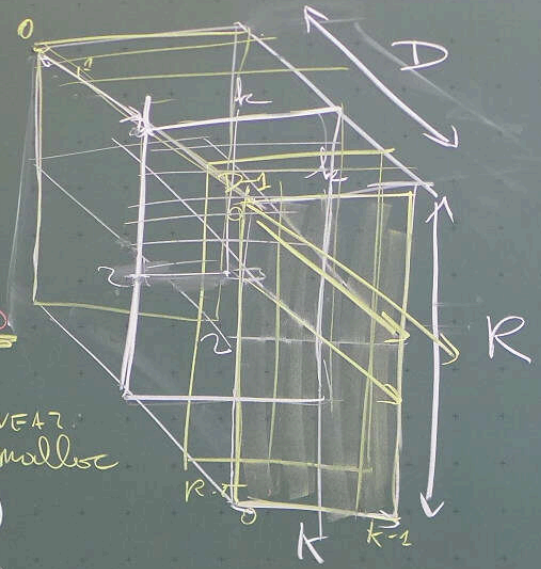
0 ≤ i < R

(0 ≤ k ≤ K)

CAVEAT: malloc



3D ARRAY



a[r, k, d]

$0 \le r < R$

$0 \le k < K$

$0 \le d < D$

0 → R-1

0 → K-1

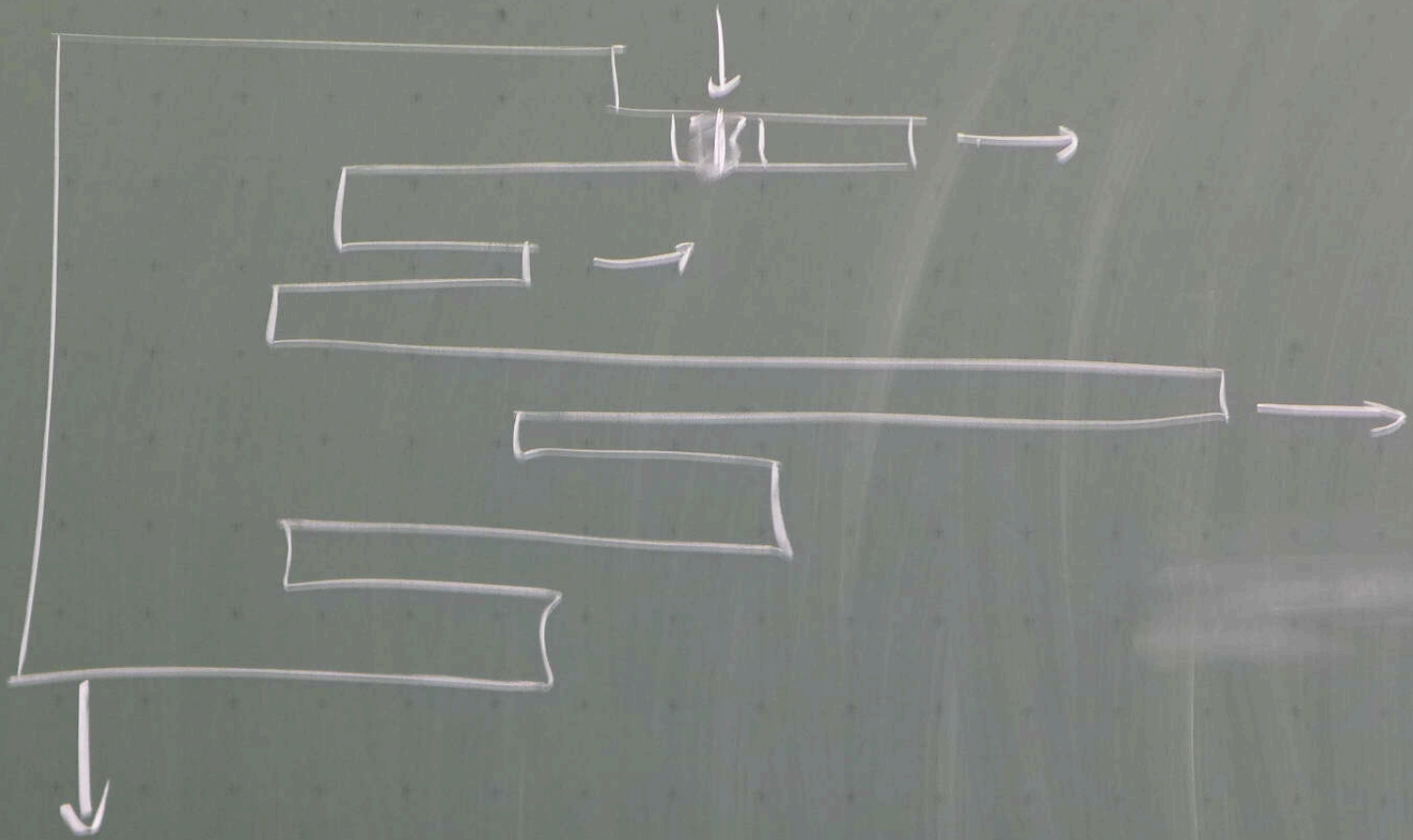
0 → D-1

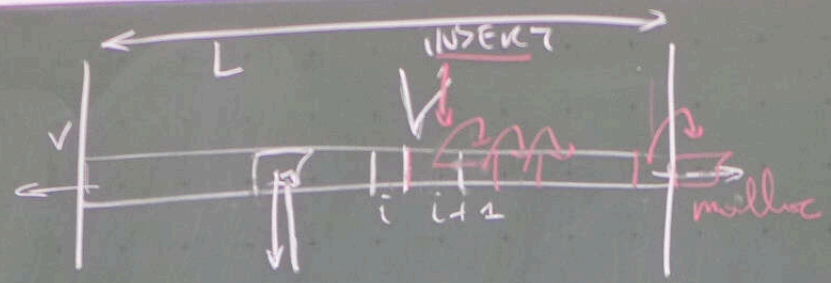
CAVEAT

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

SIZE (a) = R x K x D x SIZE (ELEMENT TYPE)

ADDRESS (a[r,k,d]) = ADDRESS (a[0,0,0]) + (k + r x K + d x R x K) x SIZE (ELEMENT TYPE)





① malloc under L

②

③ INSERT i+1

j → j+1 $j \in [i+1, i+2, \ldots, L-1)$

DYNAMIC SIZE VECTOR (LENGTH)

ALTERNATIVES:

SIZE (V) = L × SIZE(ELEMENT TYPE)

#OP → ELEM i = 1

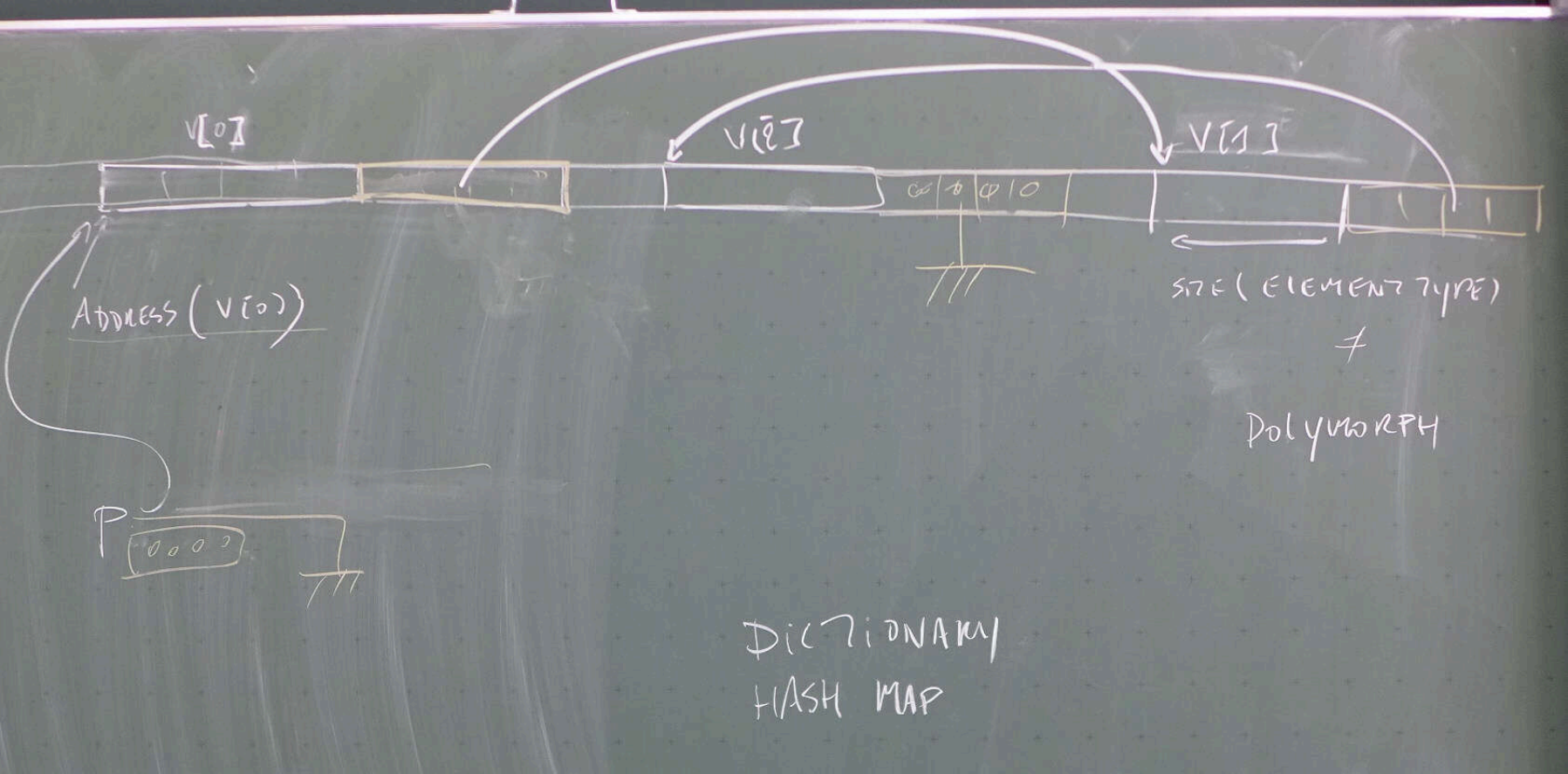
MEM REFS = 1

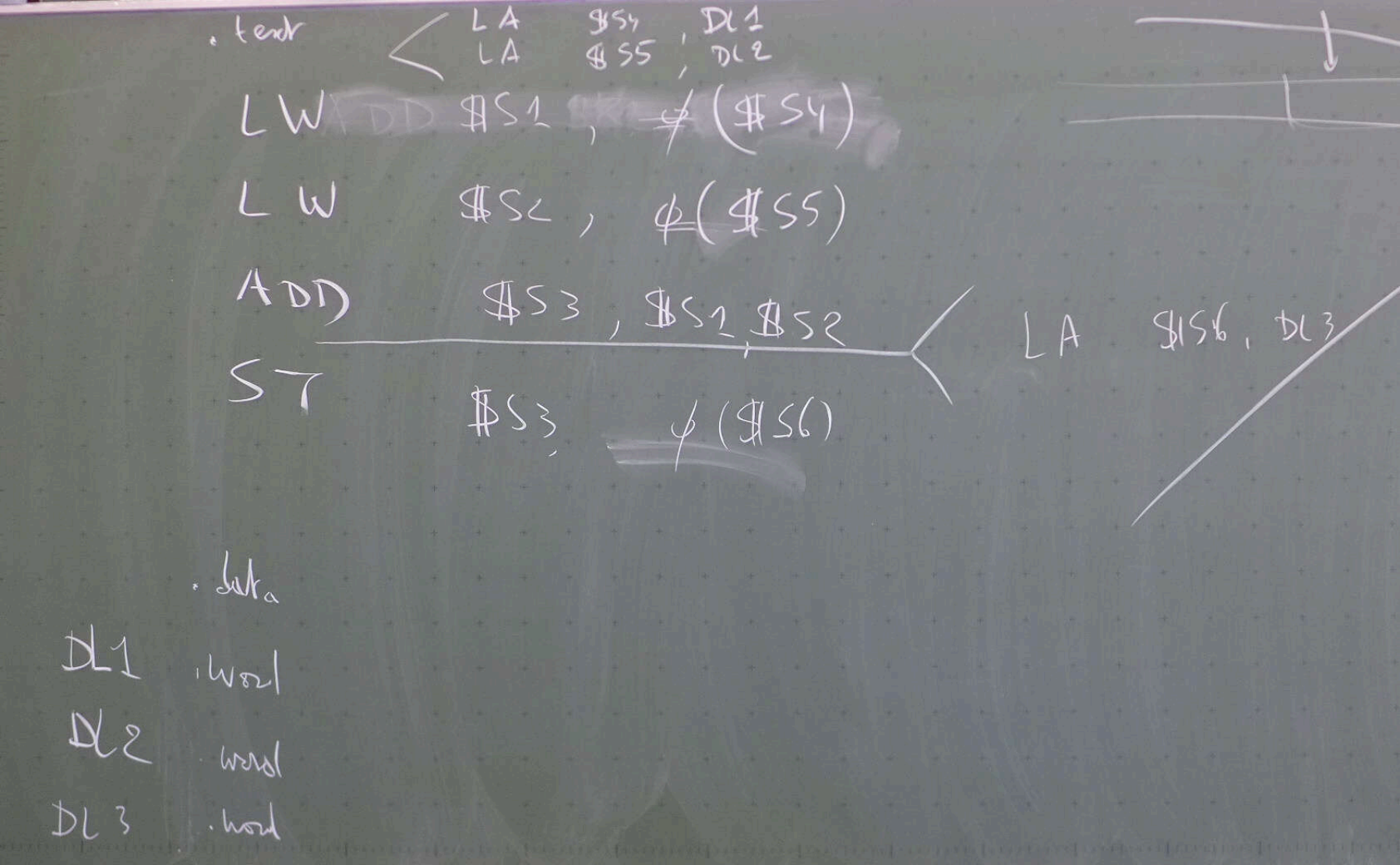
LINKED LIST

SIZE(V) = L × (SIZE(ELEMENT TYPE) + SIZE(PTR))

#OP → ELEM i =

MEM REFS = 2 × i


V[0]
V[2]
V[1]
ADDRESS (V[0])
SIZE (ELEMENT TYPE)
≠
POLYMORPH
P
DICTIONARY
HASH MAP

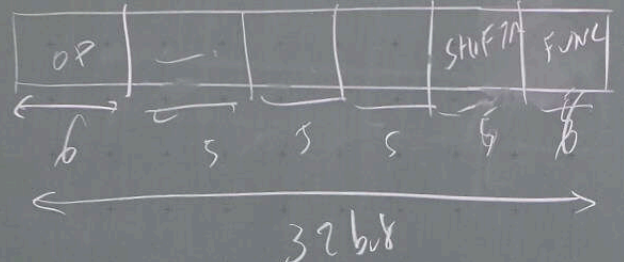

.text
LA $S4 , DL1
LA $S5 , DL2
LW $S1 , 0($S4)
LW $S2 , 0($S5)
ADD $S3 , $S1 , $S2
ST $S3 , 0($S6)
LA $S6 , DL3
.data
DL1 .word
DL2 .word
DL3 .word

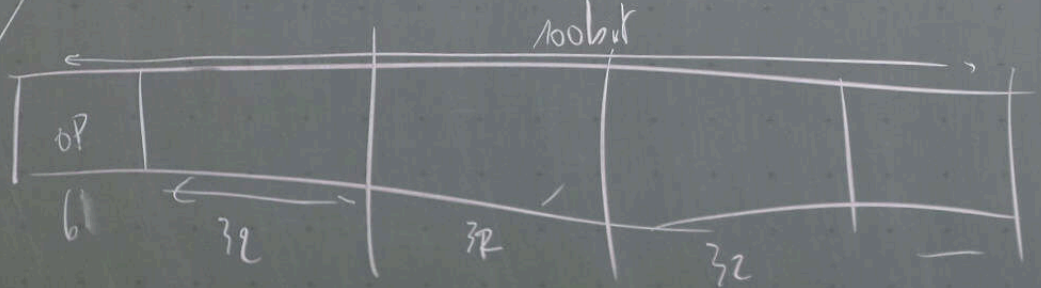
ADD $t3, $t2, $t2

ADD $s3, $s2, $s2

Δt ① DURATION OF INSTRUCTIONS

② SIZE

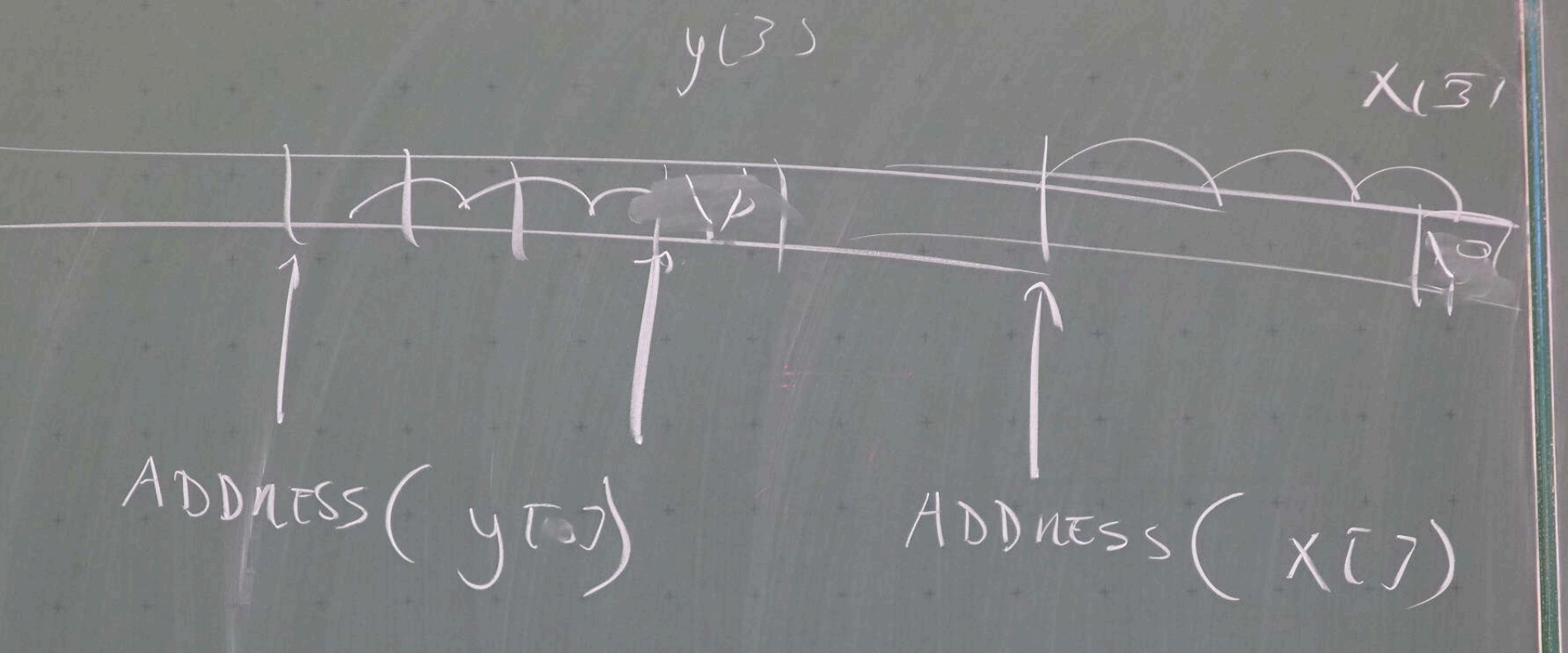


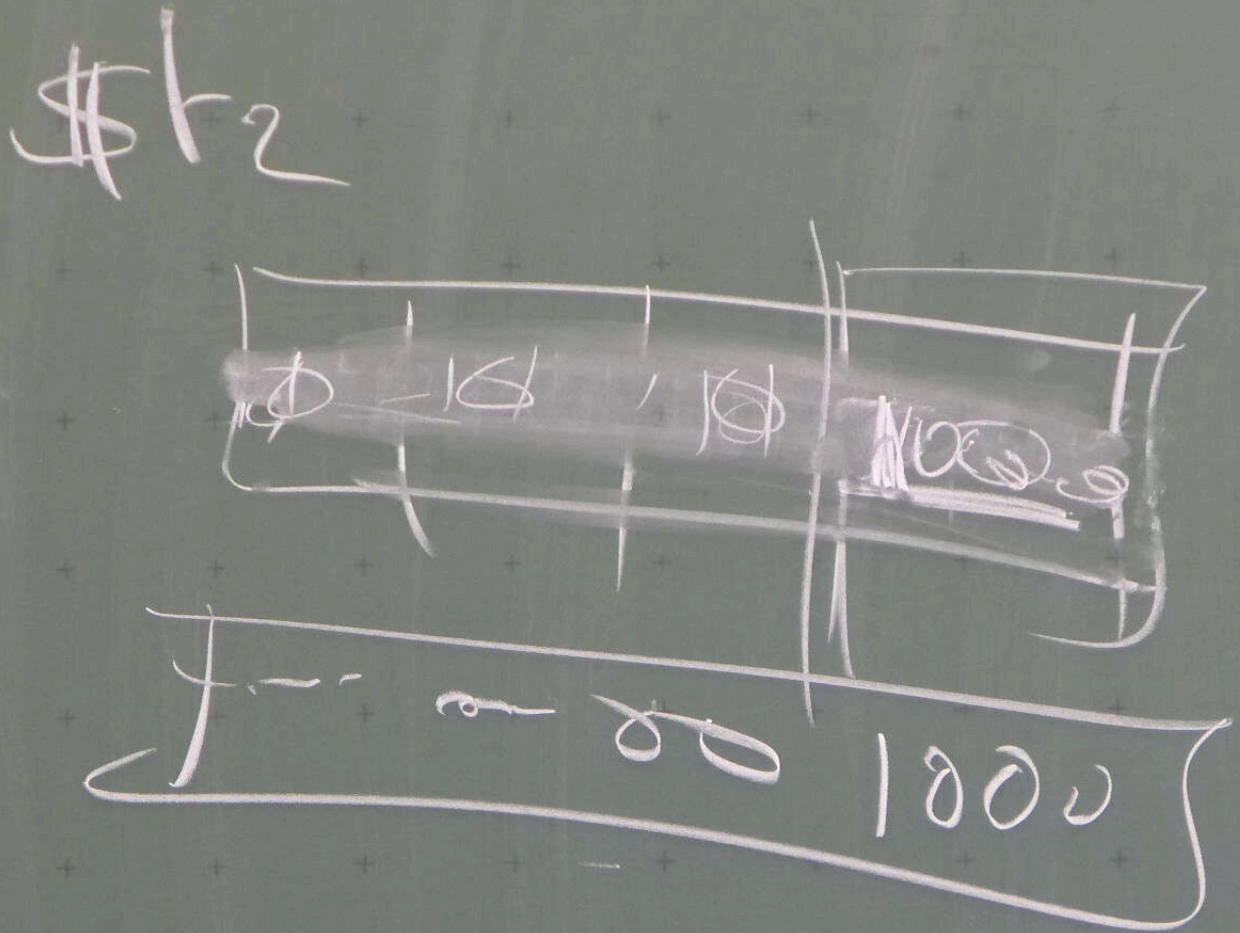


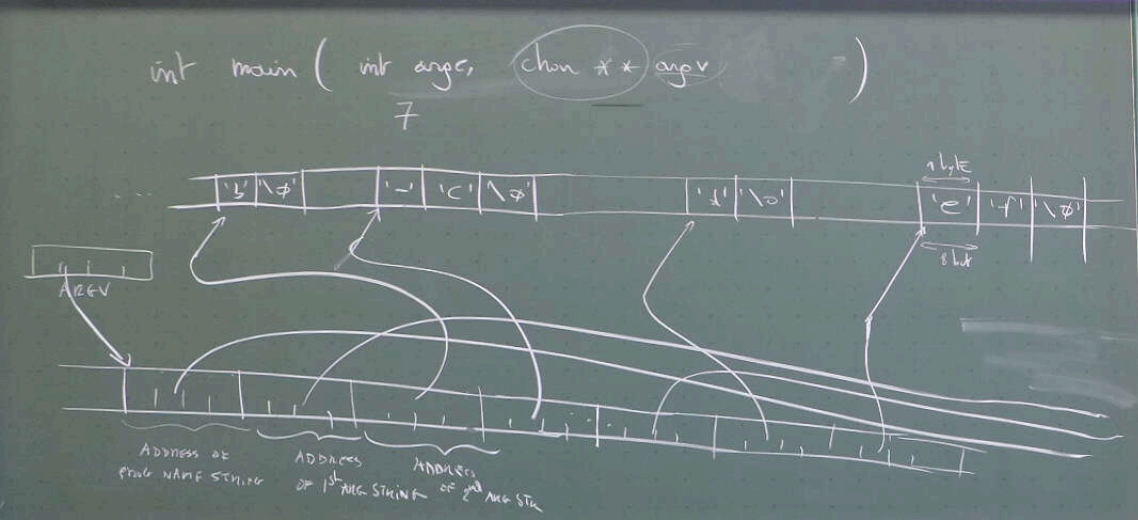
if ((a=f) || (c=d) || (a=b))

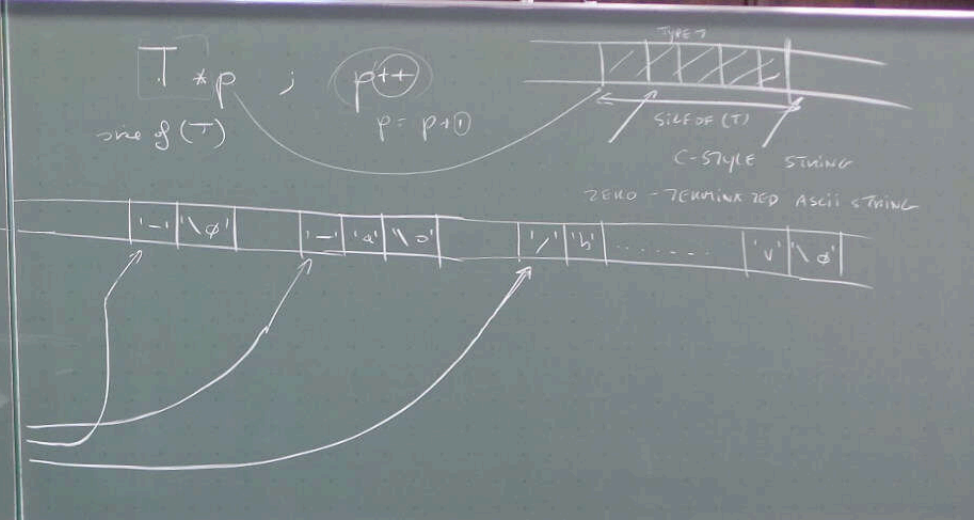
gcc -O3

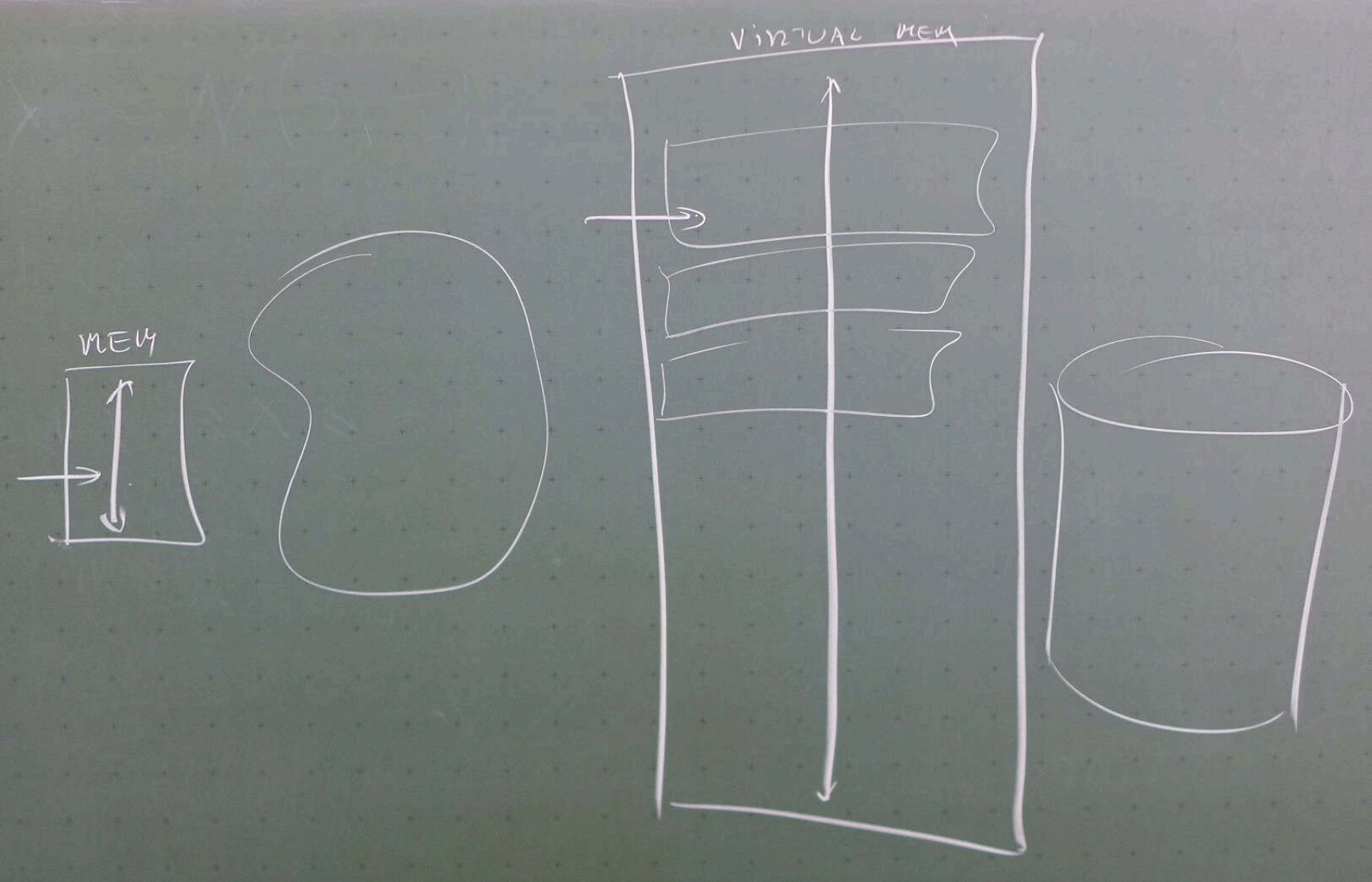
ADDRESS (V[x]) + (i) * SIZE (EL TYPE)

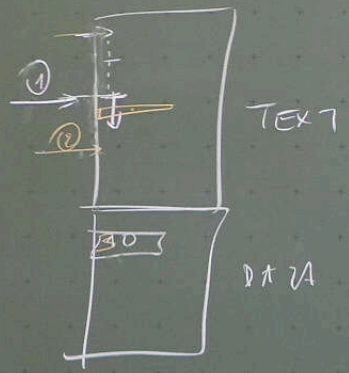

y(3)
x(3)
ADDRESS(y[0])
ADDRESS(x[])

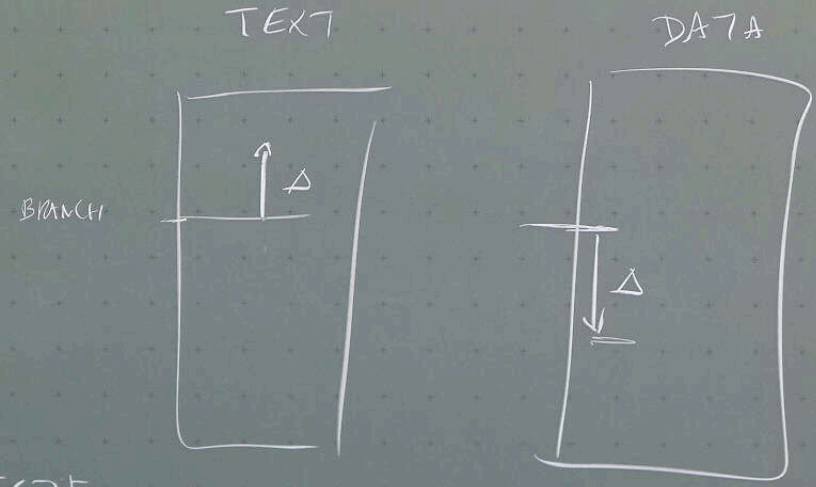



int main (int argc, char ** argv)
argc
1 byte
8 bit
ARGV
Address of prog NAME string
Address of 1st ARG string
Address of 2nd ARG str


T *p ; p++
size of (T)
p = p + 1
sizeof (T)
C-style string
ZERO - TERMINATED ASCII STRING
'v' '\0'


VIRTUAL MEM
MEM

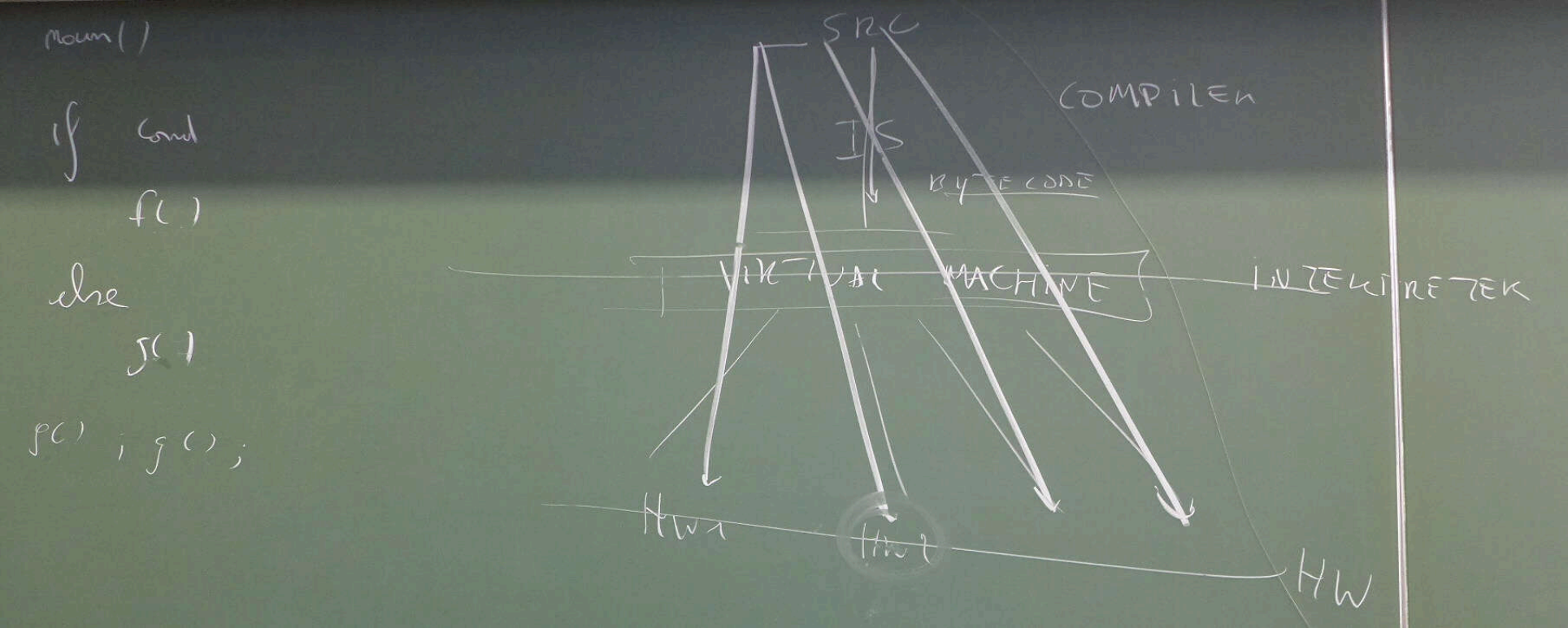




INDIRECTE ++ RELATIVE ADDRESSING → PIC

(- LONG DISTANCE → MORE INSTRUCTIONS)

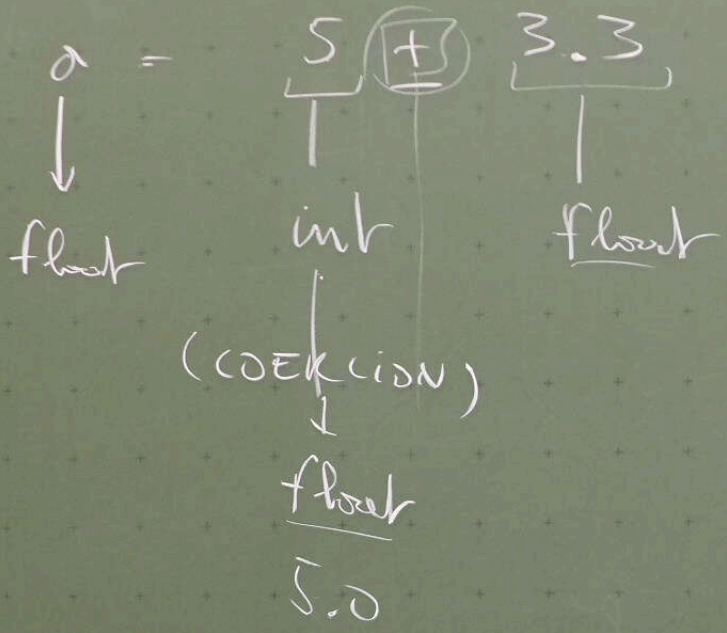
LOCALITY


main()
if Cond
f()
else
g()
f() ; g() ;
SRC
COMPILER
BYTECODE
VIRTUAL MACHINE
INTERPRETER
HW1
HW2
HW

-5 $\quad -: \mathbb{Z} \to \mathbb{Z}$ unary Min (i)

$3 - 5$ $\quad -: \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ binary Min (i , j)


MOI



$+ : int \times int \longrightarrow int$

$+ : float \times float \longrightarrow float$

RPN

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

PREFIX $+ \; 2 \; \times \; 3 \; 4$

→ POSTFIX