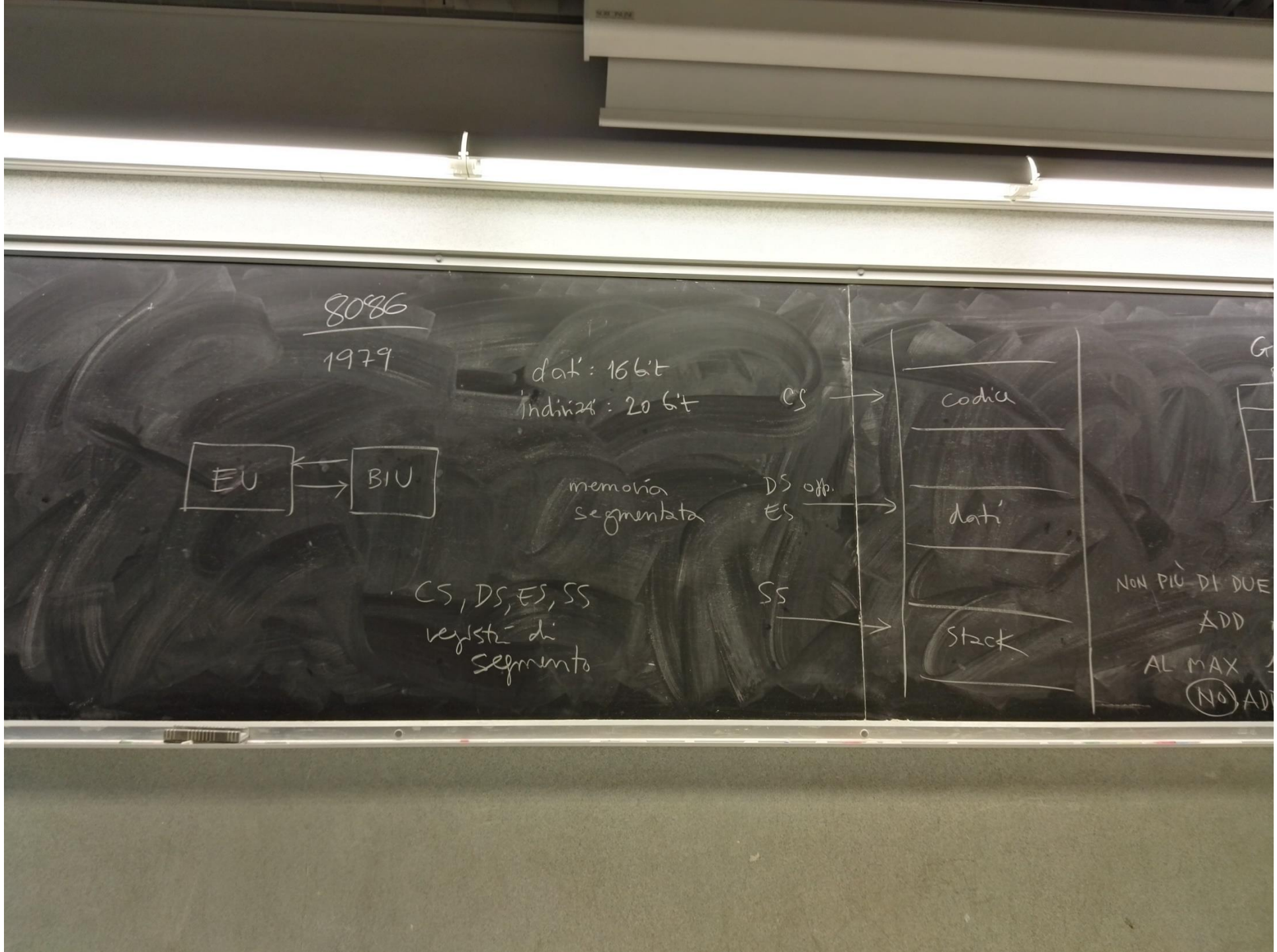
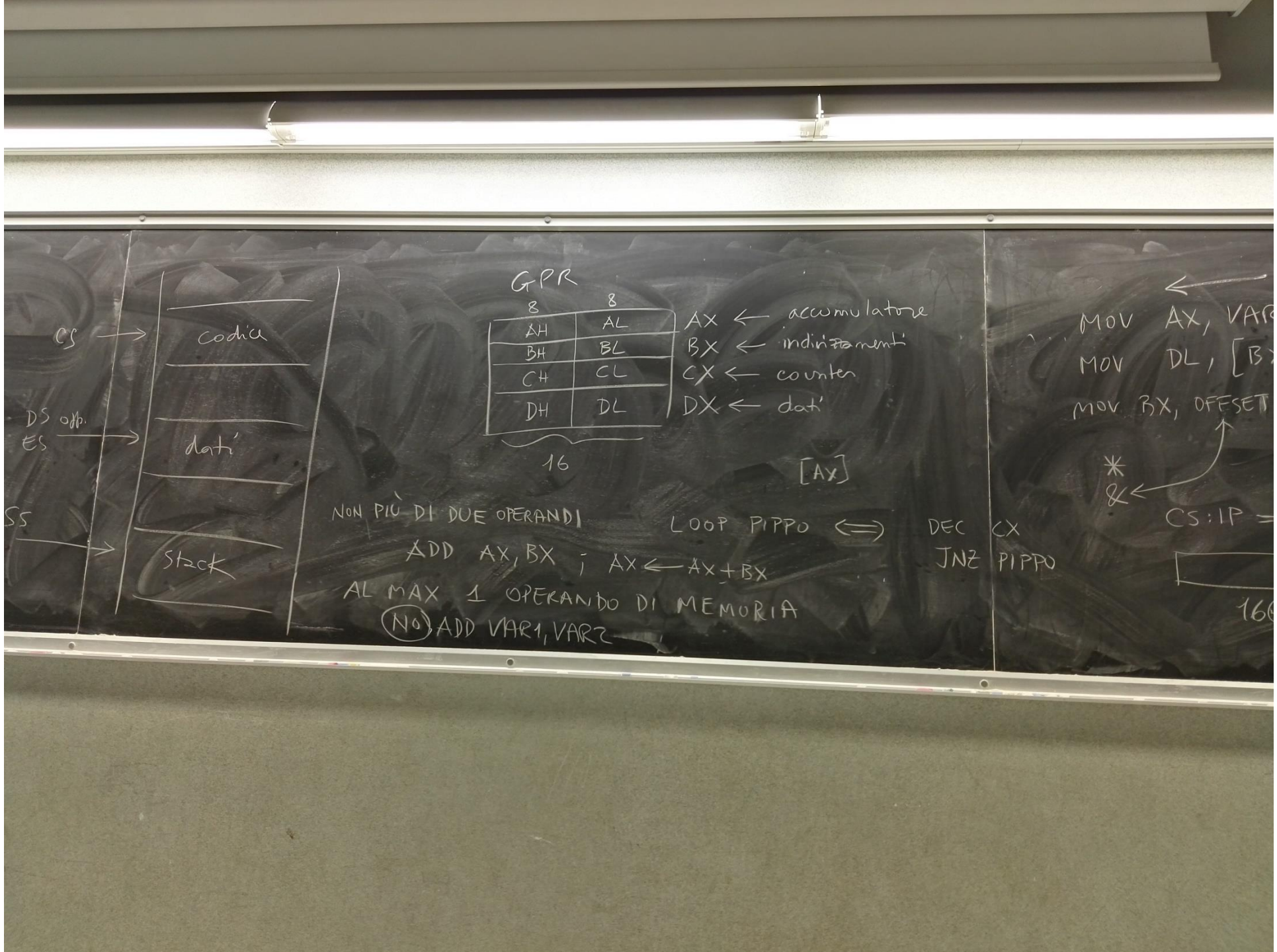
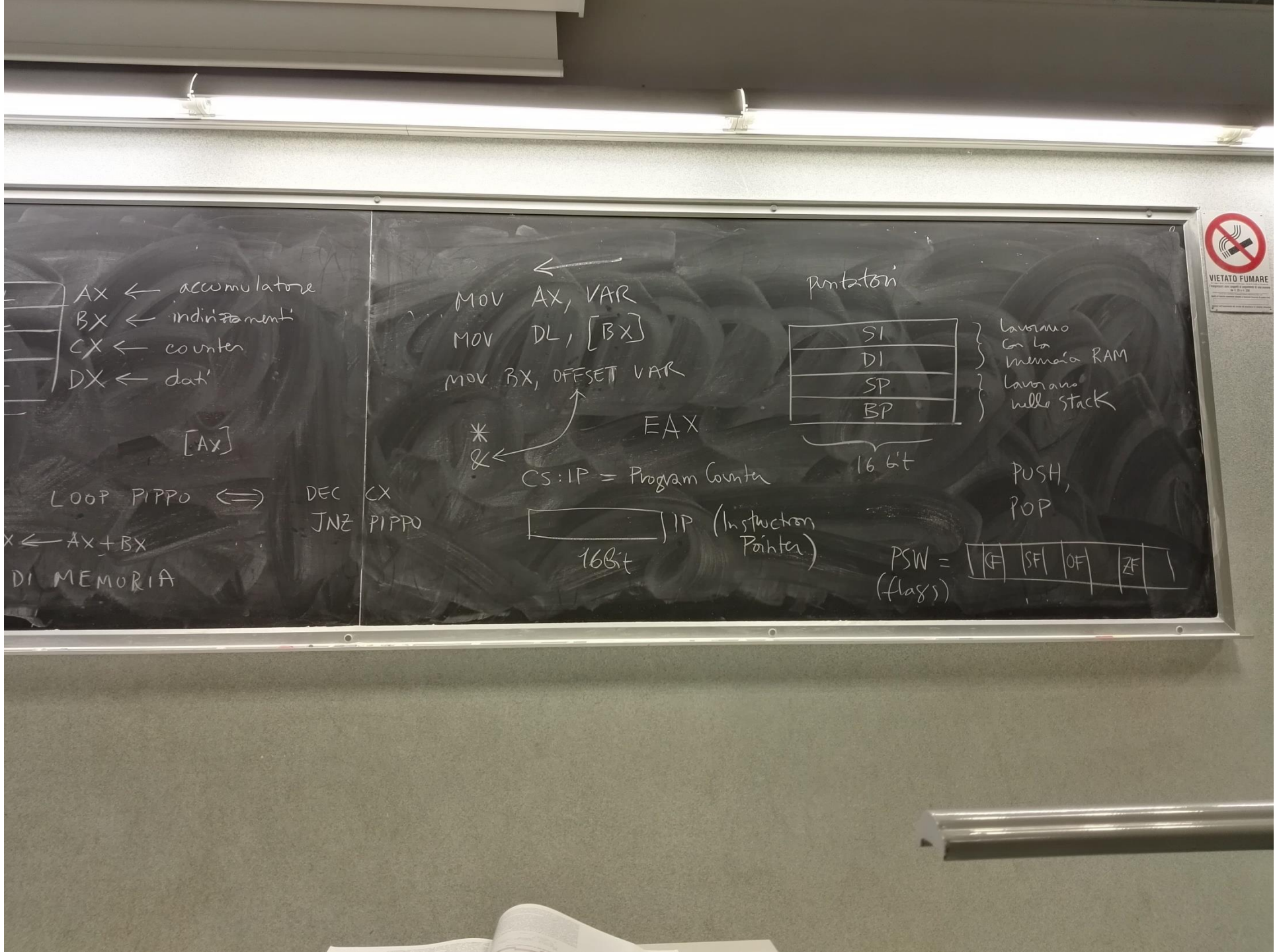
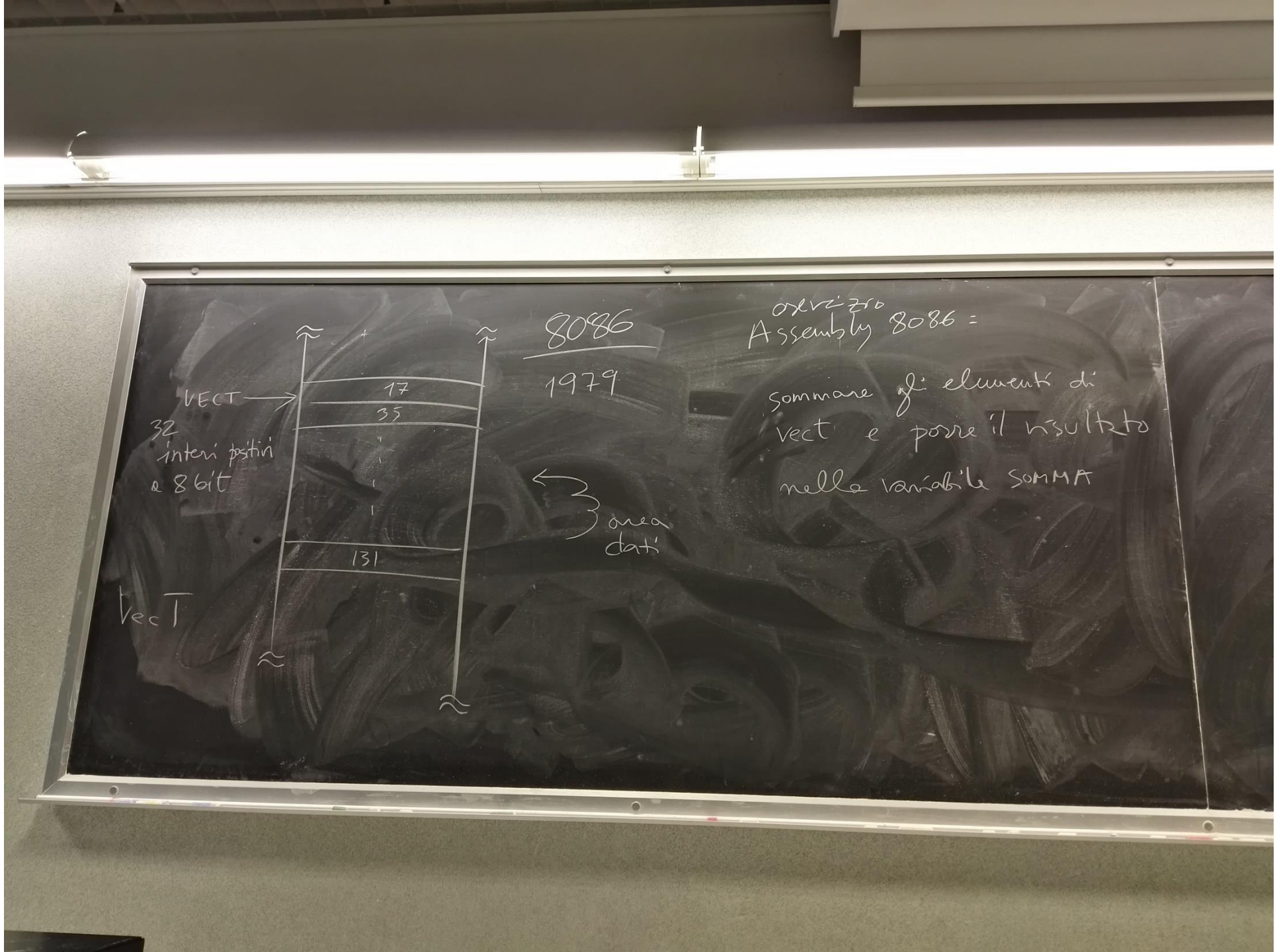

8086
1979
dati: 16 bit
indirizzi: 20 bit
EU
BIU
memoria segmentata
CS
DS off.
ES
SS
CS, DS, ES, SS
registri di segmento
codice
dati
stack
NON PIÙ DI DUE
ADD
AL MAX
NO


CS
codice
DS opp.
ES
dati
SS
stack
GPR
8
8
AH
AL
BH
BL
CH
CL
DH
DL
16
AX ← accumulatore
BX ← indirizzamenti
CX ← counter
DX ← dati
[AX]
NON PIÙ DI DUE OPERANDI
ADD AX, BX ; AX ← AX + BX
AL MAX 1 OPERANDO DI MEMORIA
NO ADD VAR1, VAR2
LOOP PIPPO ⟺ DEC CX
JNZ PIPPO
MOV AX, VAR
MOV DL, [BX]
MOV BX, OFFSET
*
&
CS:IP =
16

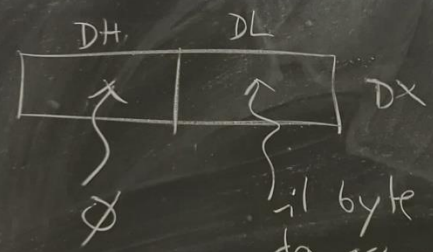

AX ← accumulatore
BX ← indirizzamenti
CX ← counter
DX ← dati
[AX]
LOOP PIPPO ⟺ DEC CX
JNZ PIPPO
X ← AX + BX
DI MEMORIA
MOV AX, VAR
MOV DL, [BX]
MOV BX, OFFSET VAR
EAX
CS:IP = Program Counter
IP (Instruction Pointer)
16Bit
puntatori
SI
DI
SP
BP
Lavorano con la memoria RAM
Lavorano nello stack
16 bit
PUSH,
POP
PSW = (flags)
CF
SF
OF
ZF
VIETATO FUMARE

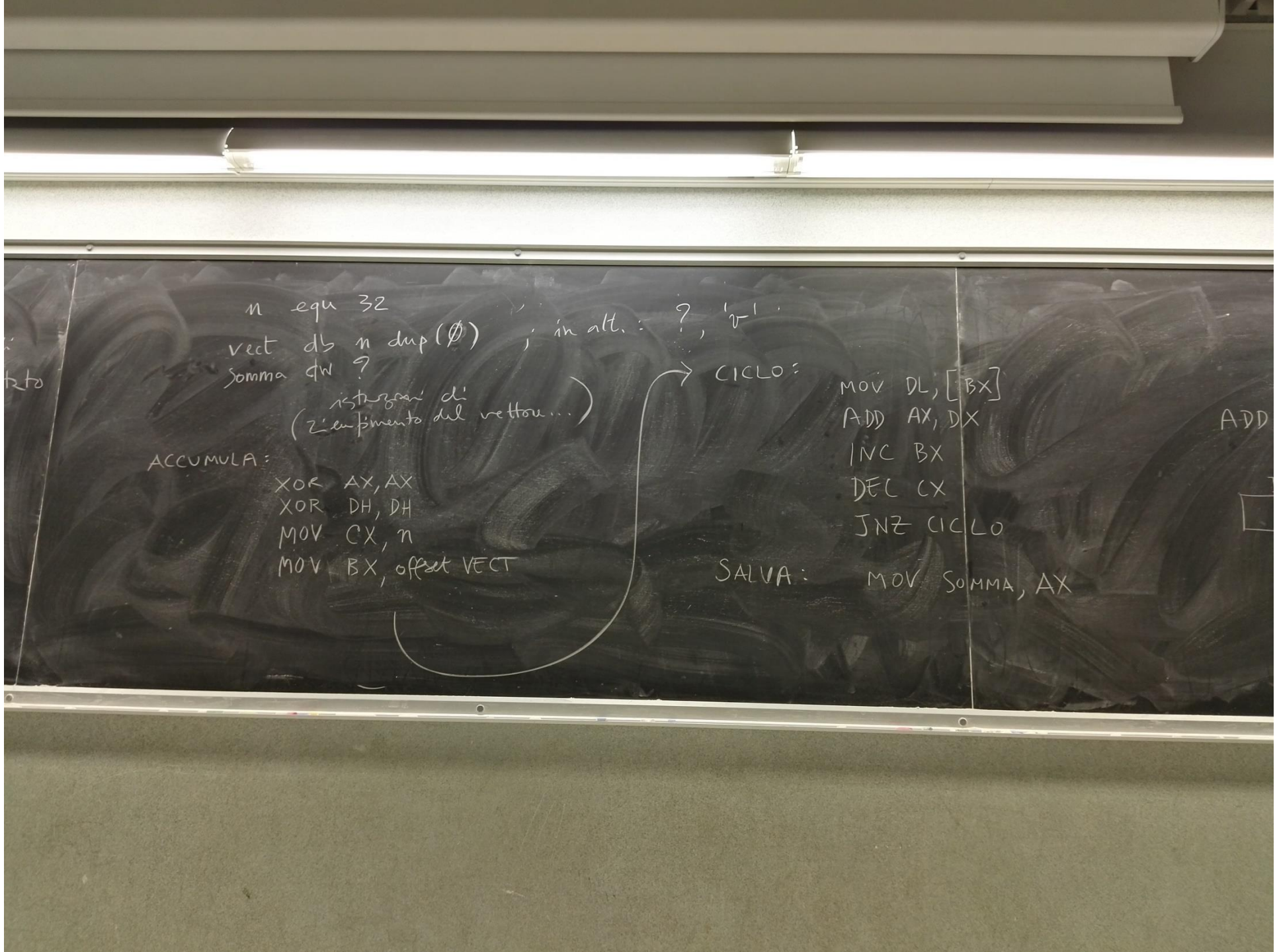

8086
1979
esercizio
Assembly 8086 =
sommare gli elementi di
vect e porre il risultato
nella variabile SOMMA
VECT
32
interi positivi
a 8 bit
17
35
131
area
dati
VecT

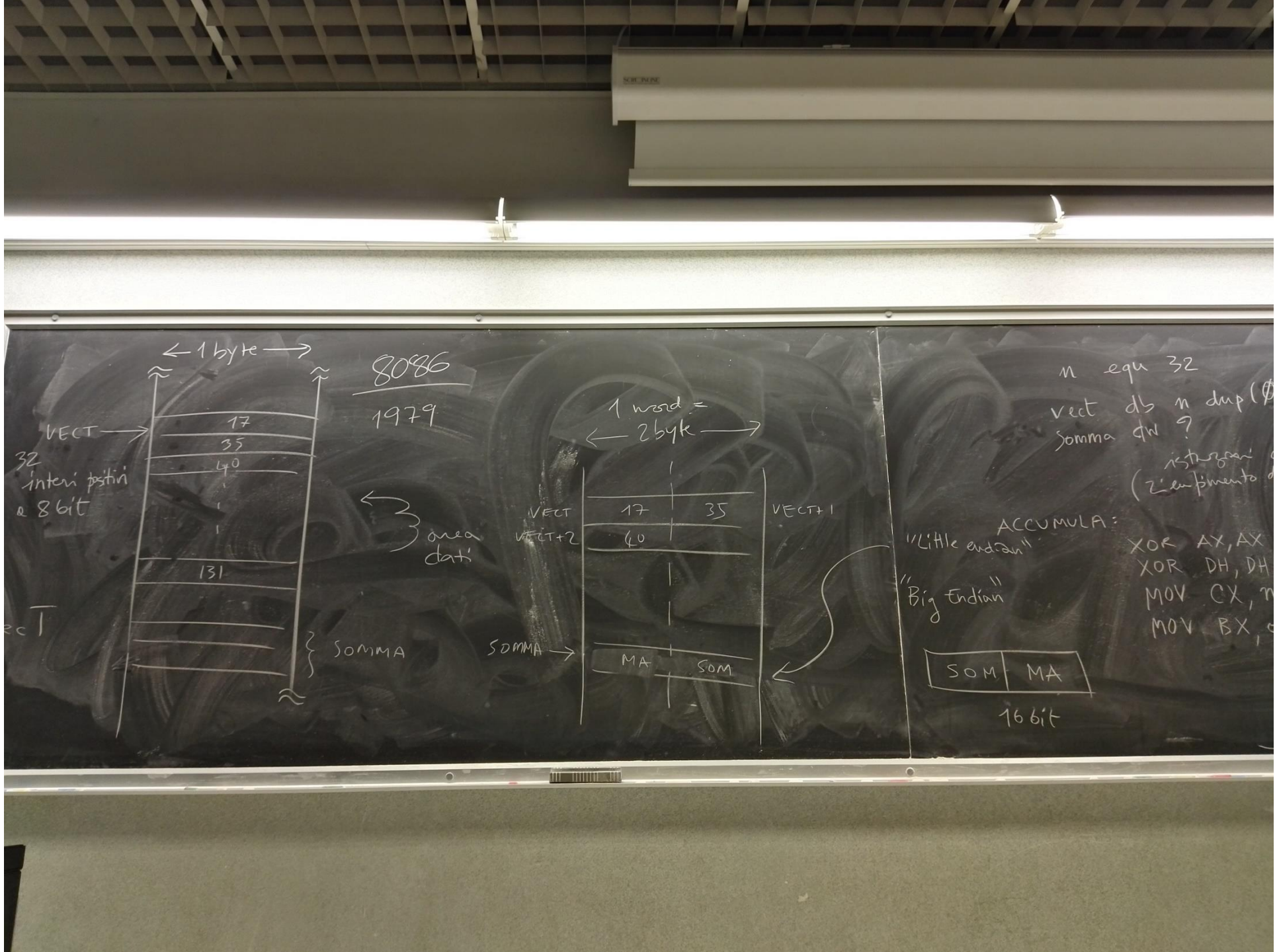
DL, [BX]
AX, DX
BX
CX
CICLO

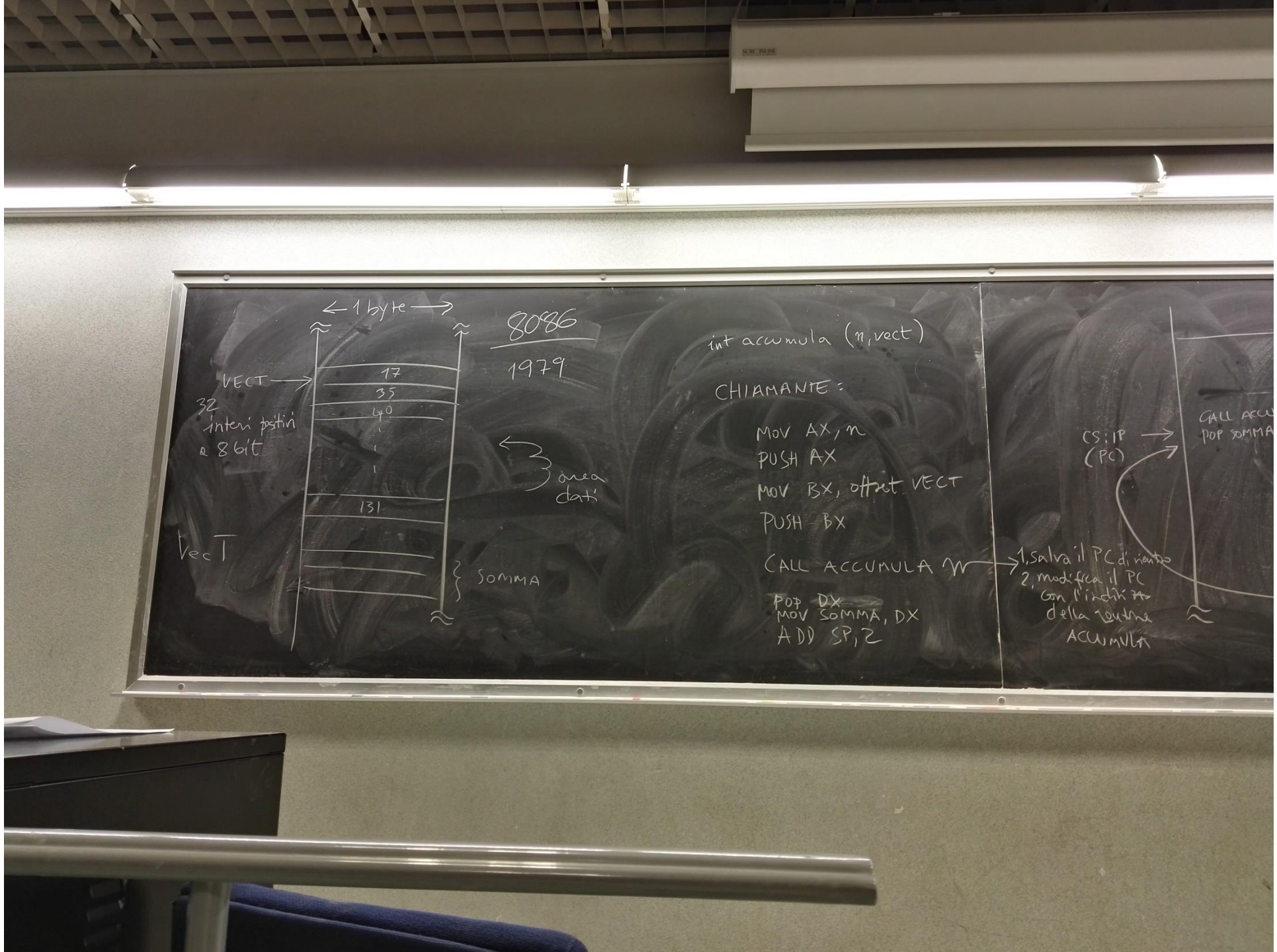
SOMMA, AX

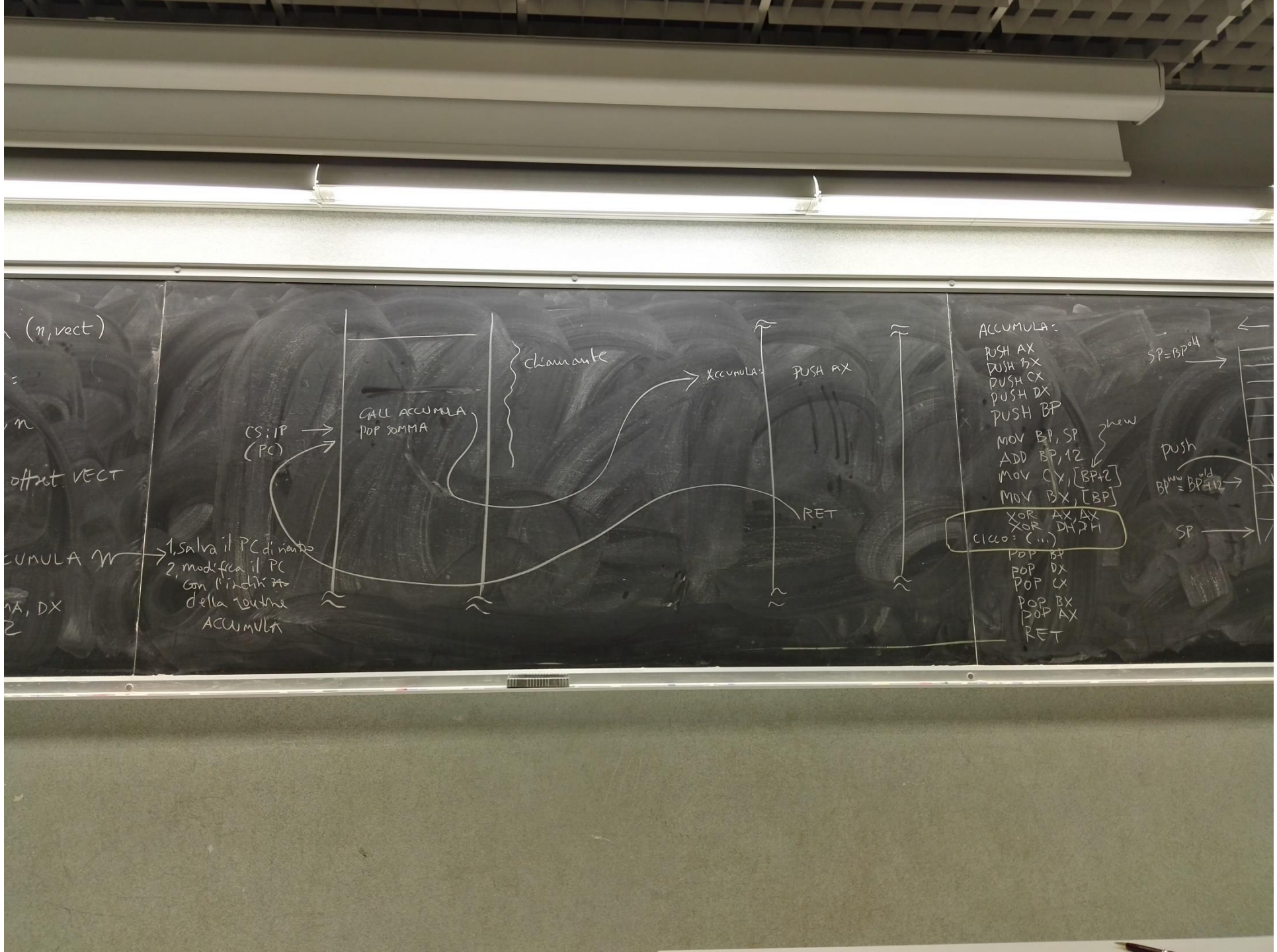
ADD AX, DX accumulazione

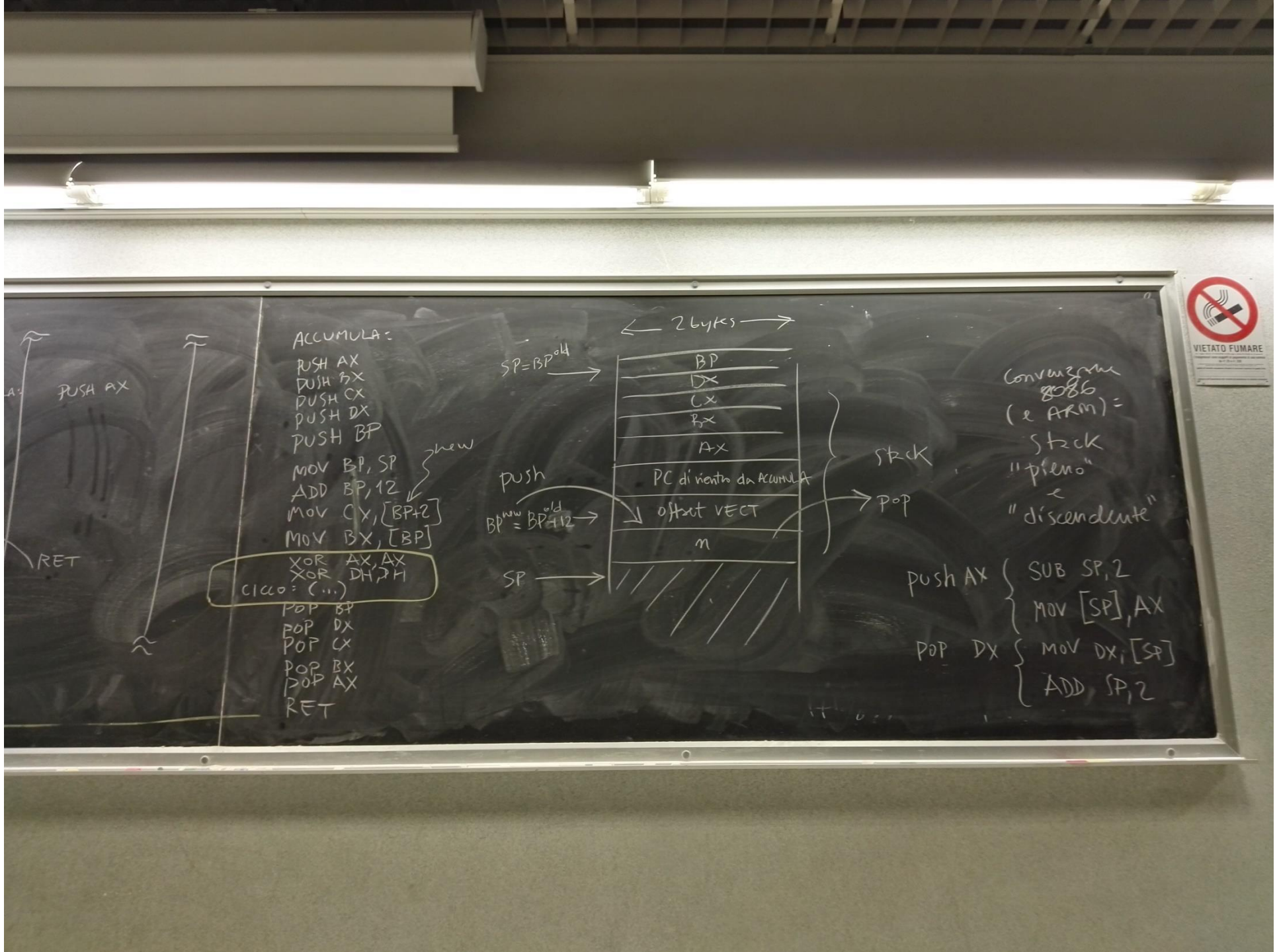



n equ 32
vect db n dup(Ø) ; in alt.: ?, '0'
somma dw ?
(istruzioni di riempimento del vettore...)
ACCUMULA:
XOR AX,AX
XOR DH,DH
MOV CX,n
MOV BX, offset VECT
CICLO:
MOV DL,[BX]
ADD AX,DX
INC BX
DEC CX
JNZ CICLO
SALVA: MOV SOMMA, AX
ADD


← 1 byte →
8086
1979
VECT
32 interi positivi a 8 bit
17
35
40
131
SOMMA
area dati
1 word = 2byte
VECT
VECT+1
VECT+2
17
35
40
SOMMA
MA
SOM
n equ 32
vect db n dup(
somma dw ?
ACCUMULA:
XOR AX,AX
XOR DH,DH
MOV CX,n
MOV BX,
"Little endian"
"Big Endian"
SOM
MA
16 bit


← 1 byte →
8086
1979
VECT
32 interi positivi a 8 bit
17
35
40
131
area dati
SOMMA
Vect
int accumula (n, vect)
CHIAMANTE:
MOV AX, n
PUSH AX
MOV BX, offset VECT
PUSH BX
CALL ACCUMULA
POP DX
MOV SOMMA, DX
ADD SP, 2
CS:IP (PC)
1, salva il PC di ritorno
2, modifica il PC con l'indirizzo della routine ACCUMULA


(n, vect)
offset VECT
CUMULA
NA, DX
2
CS:IP
(PC)
CALL ACCUMULA
POP SOMMA
chiamante
ACCUMULA:
PUSH AX
RET
1, salva il PC di ritorno
2, modifica il PC
con l'indirizzo
della routine
ACCUMULA
ACCUMULA:
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH BP
MOV BP, SP
ADD BP, 12
MOV CX, [BP+2]
MOV BX, [BP]
new
XOR AX, AX
XOR DH, DH
ciclo: (...)
POP BP
POP DX
POP CX
POP BX
POP AX
RET
SP=BP-04
Push
BP new = BP old - 12
SP


PUSH AX
RET
ACCUMULA:
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH BP
MOV BP, SP
ADD BP, 12
MOV CX, [BP+2]
MOV BX, [BP]
XOR AX, AX
XOR DH, DH
CICLO: (...)
POP BP
POP DX
POP CX
POP BX
POP AX
RET
new
SP=BP^old
2 bytes
BP
DX
CX
BX
AX
PC di ritorno da ACCUMULA
offset VECT
n
SP
push
BP^new = BP^old + 12
stack
pop
Convenzione 8086 (e ARM): stack "pieno" e "discendente"
push AX
SUB SP, 2
MOV [SP], AX
POP DX
MOV DX, [SP]
ADD SP, 2
VIETATO FUMARE