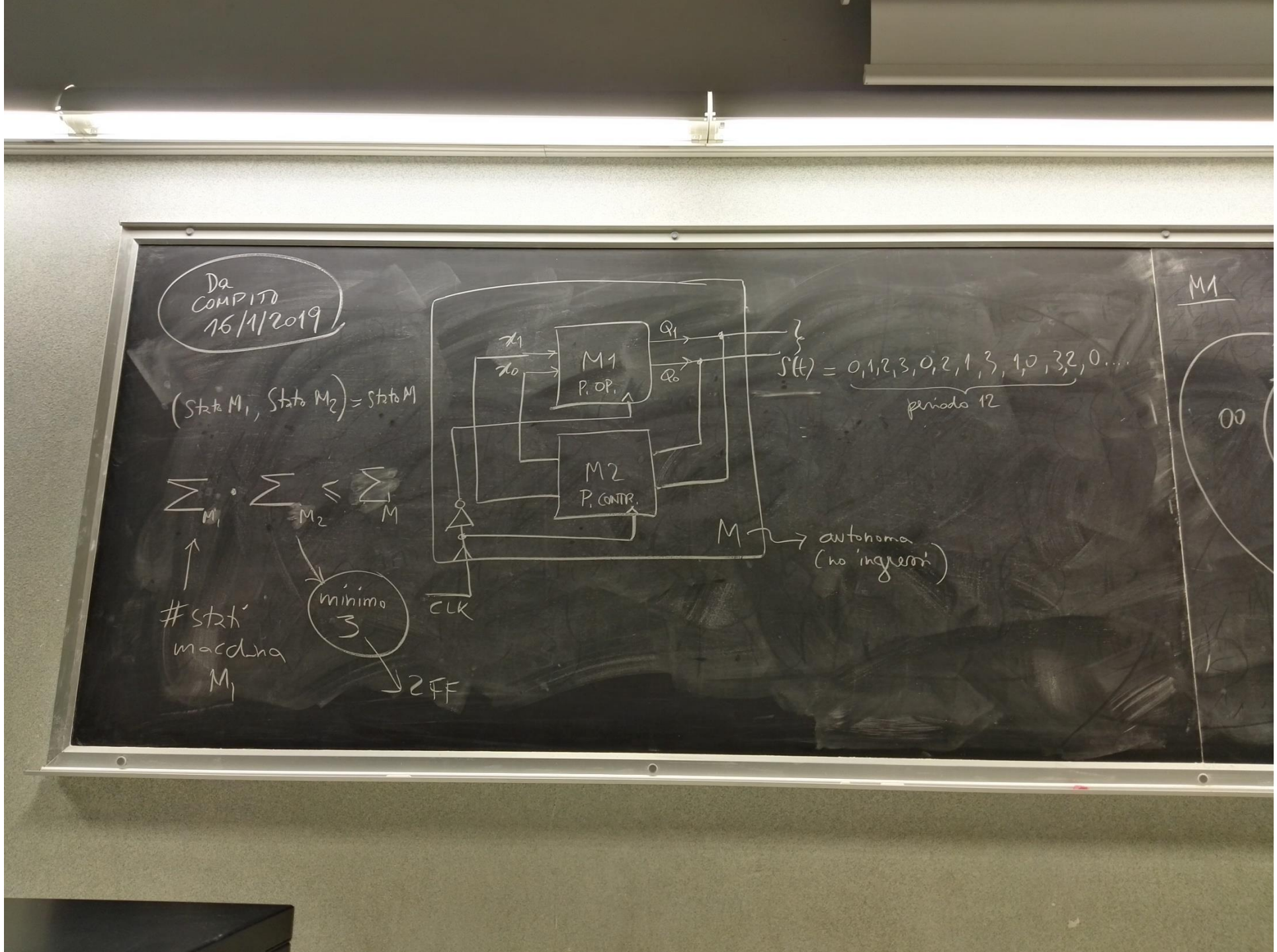
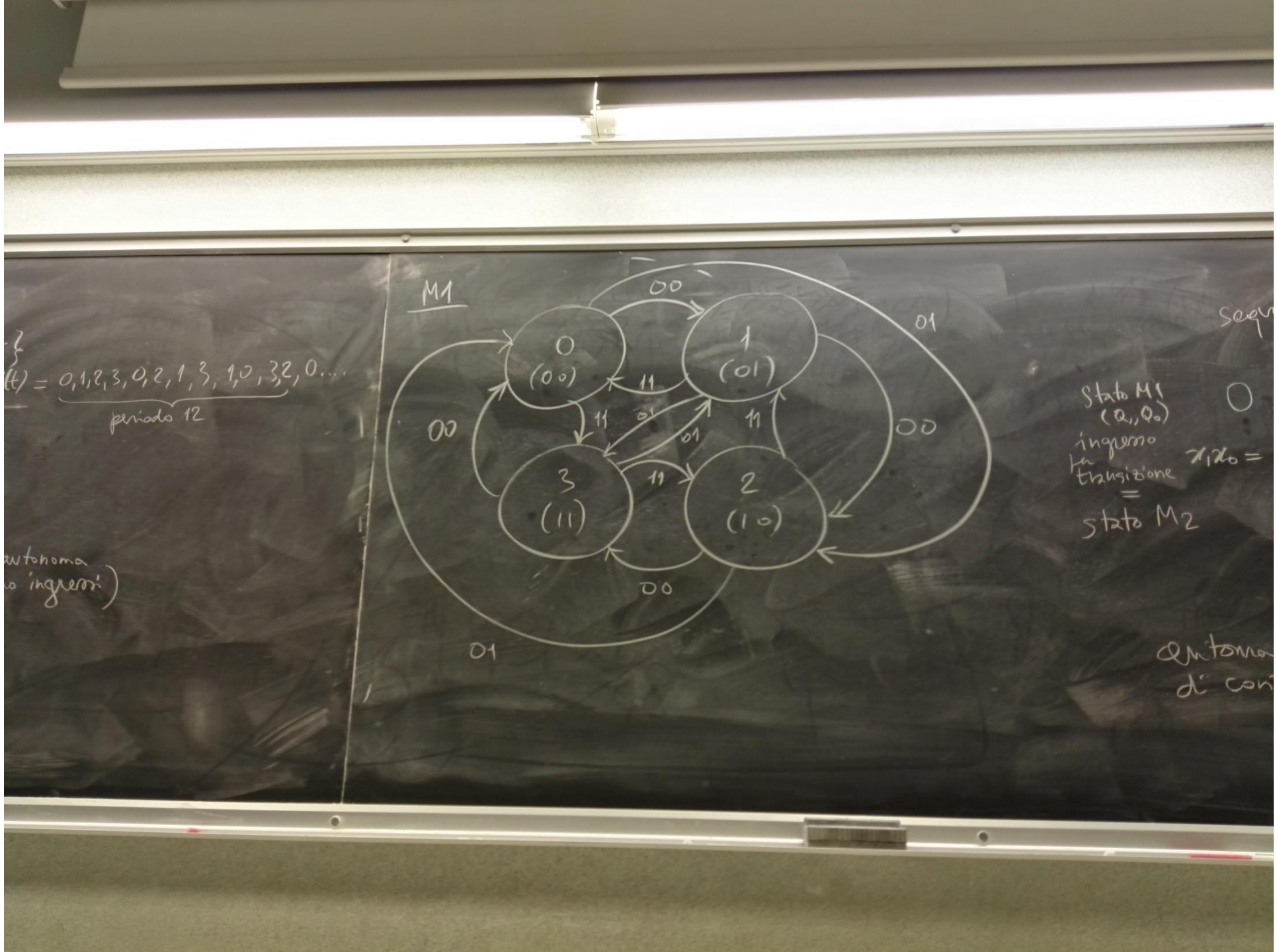
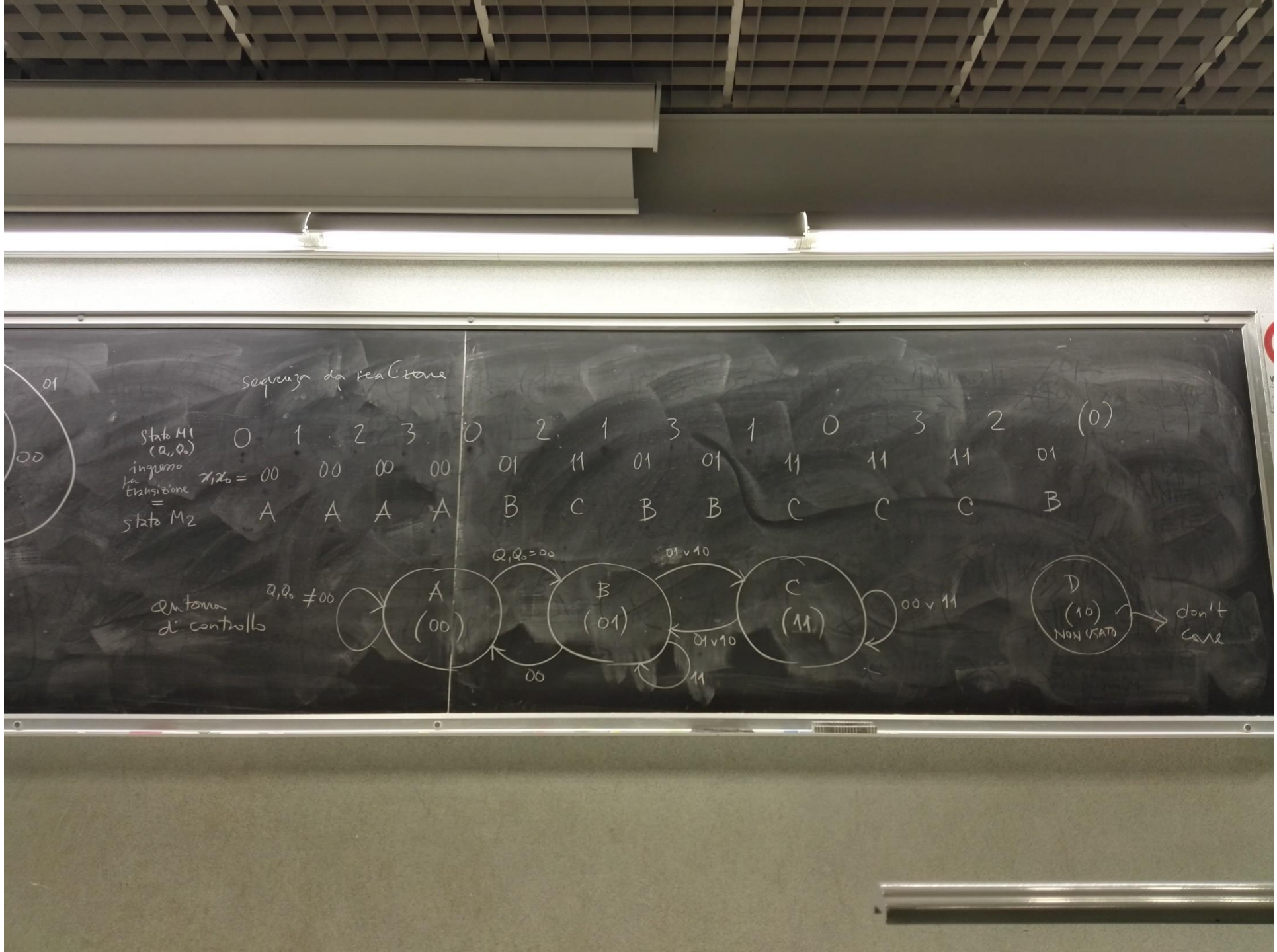
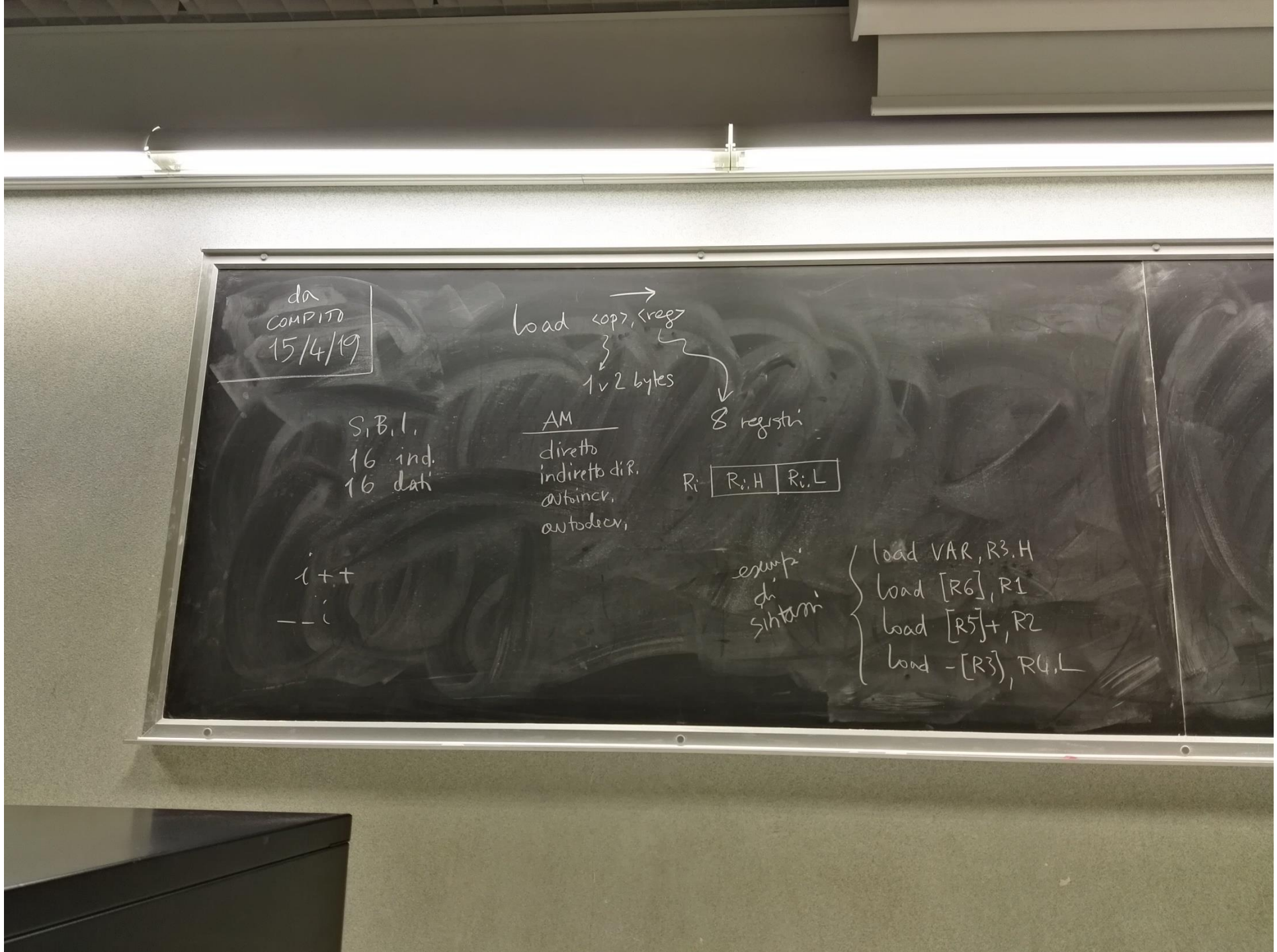
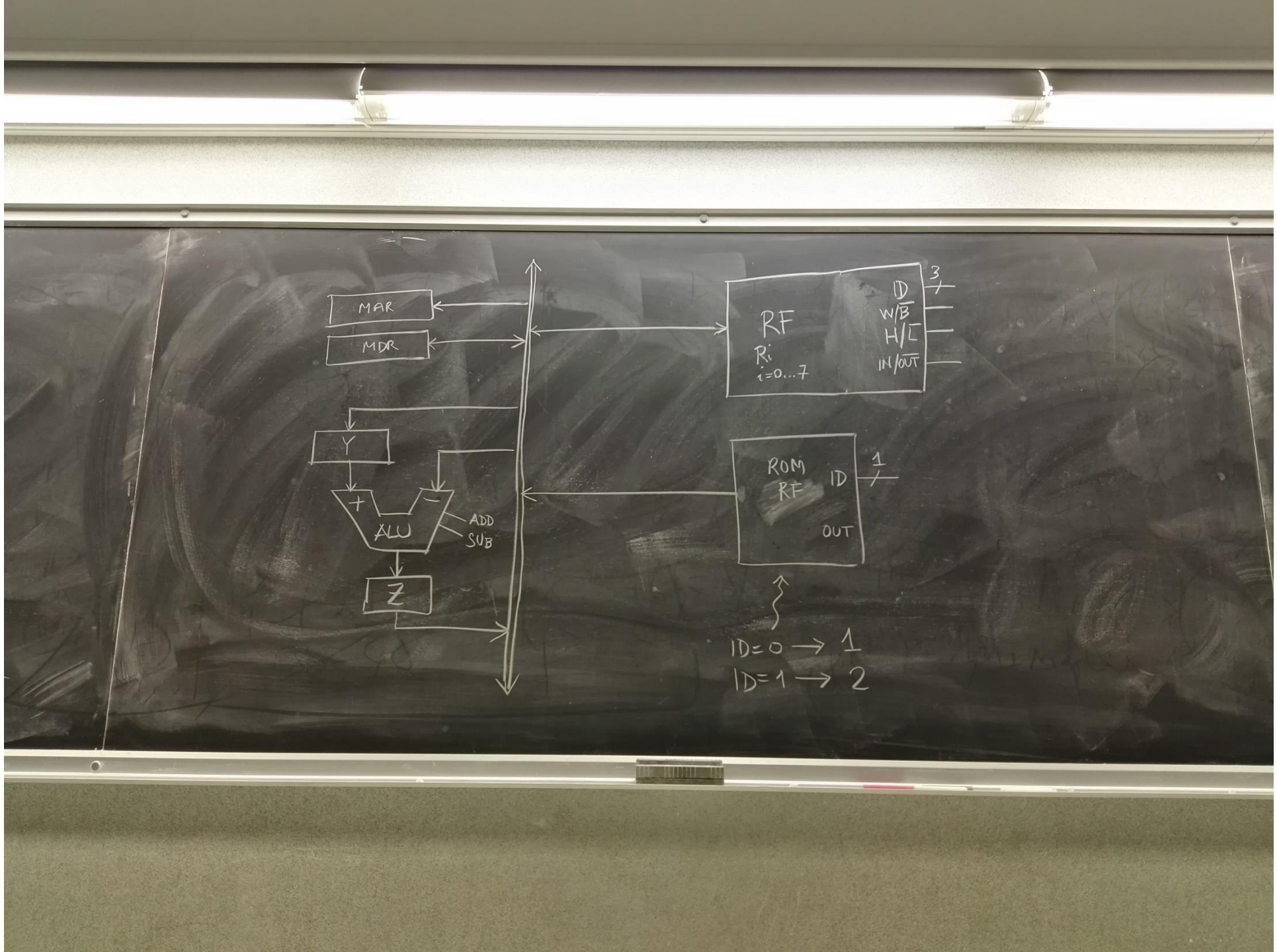
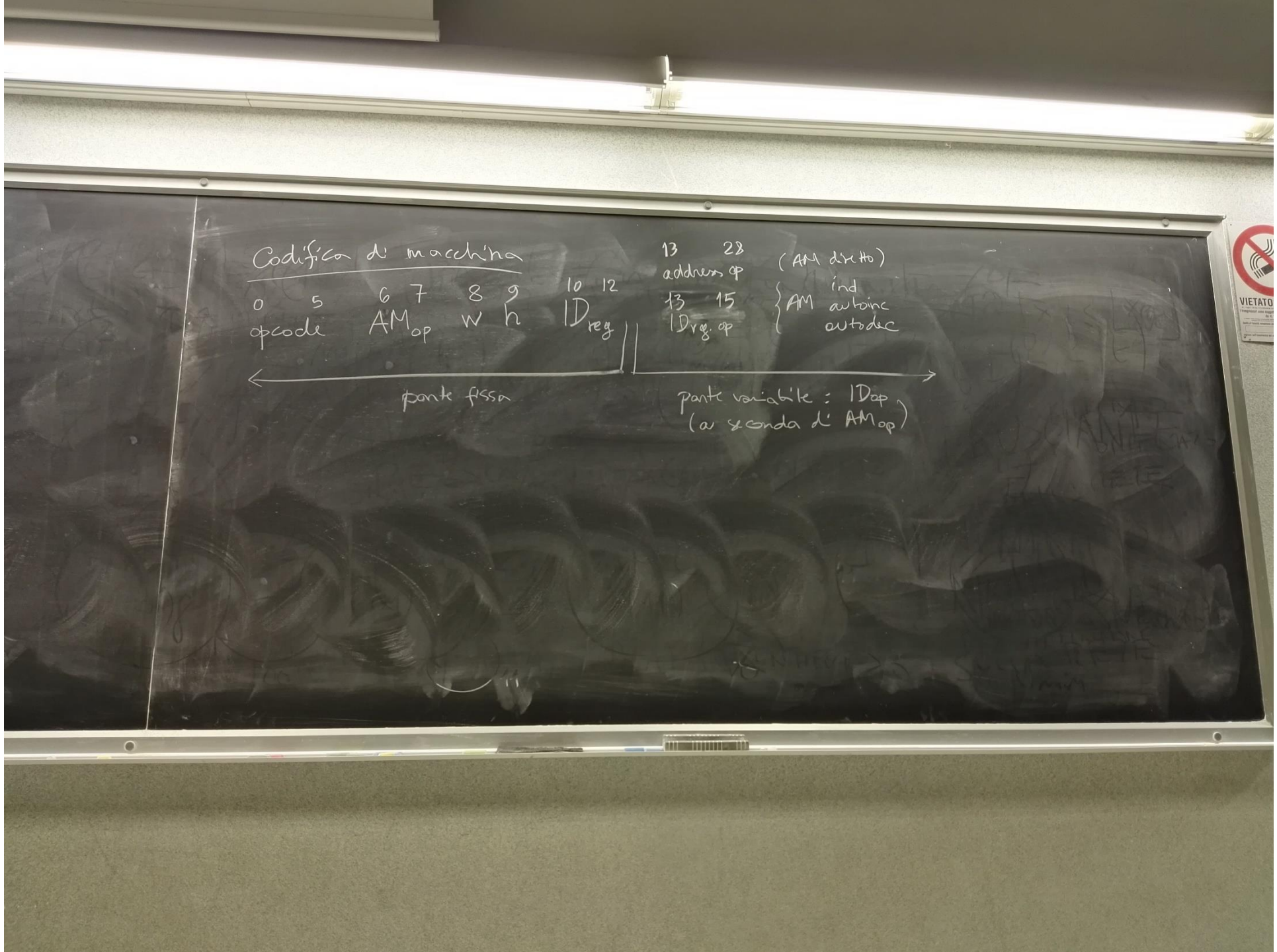

Da COMPITO 16/1/2019
(Stato M1, Stato M2) = Stato M
M1
P. OP.
M2
P. CONTR.
M
autonoma (no ingressi)
CLK
S(t) = 0,1,2,3,0,2,1,3,1,0,3,2,0....
periodo 12
#stati macchina M1
minimo 3
2 FF
M1
00

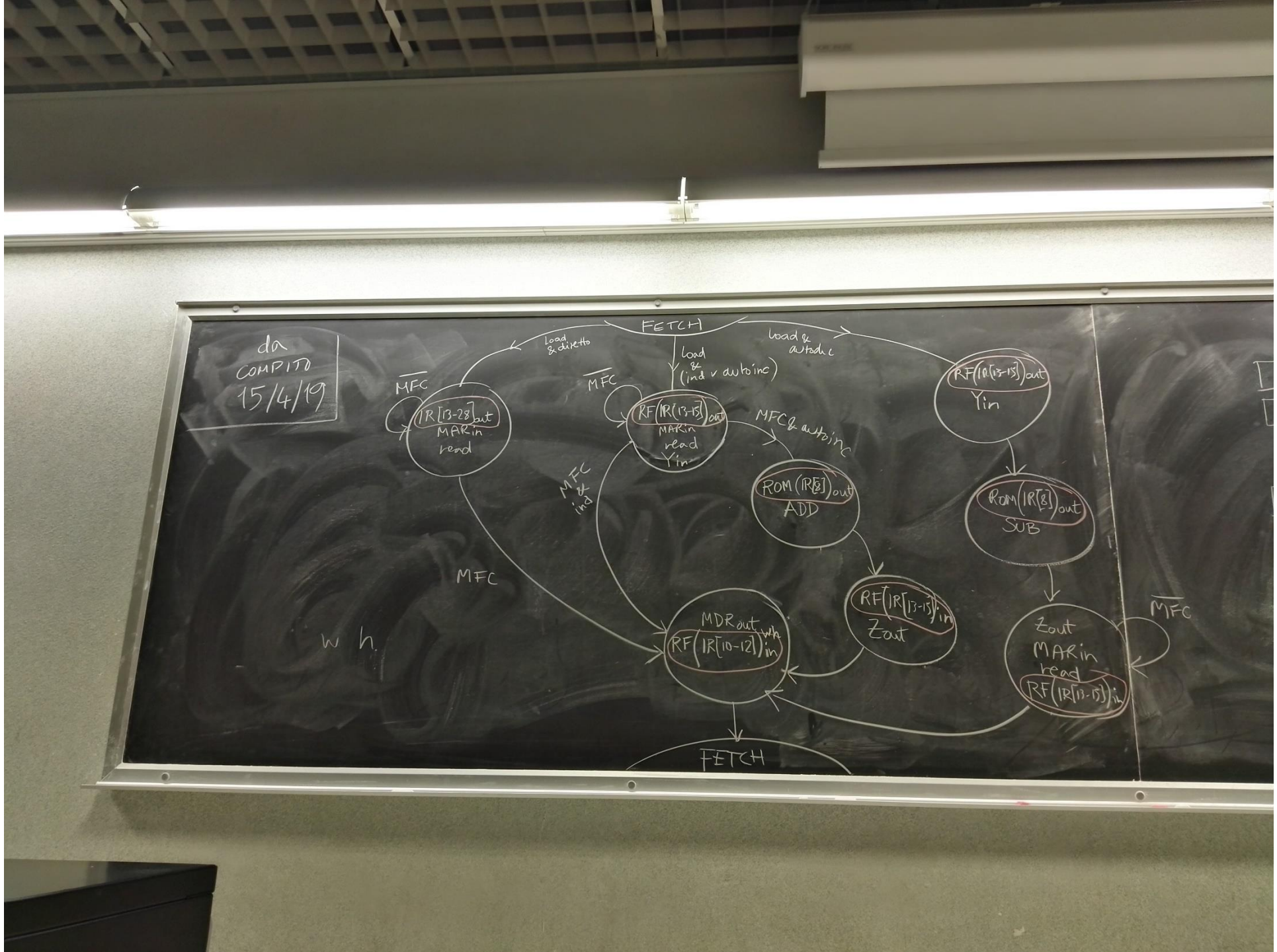

M1
0
(00)
1
(01)
3
(11)
2
(10)
00
01
11
Stato M1
(Q1,Q0)
ingresso
la transizione
=
stato M2
x1x0 =
0,1,2,3,0,2,1,3,1,0,3,2,0....
periodo 12

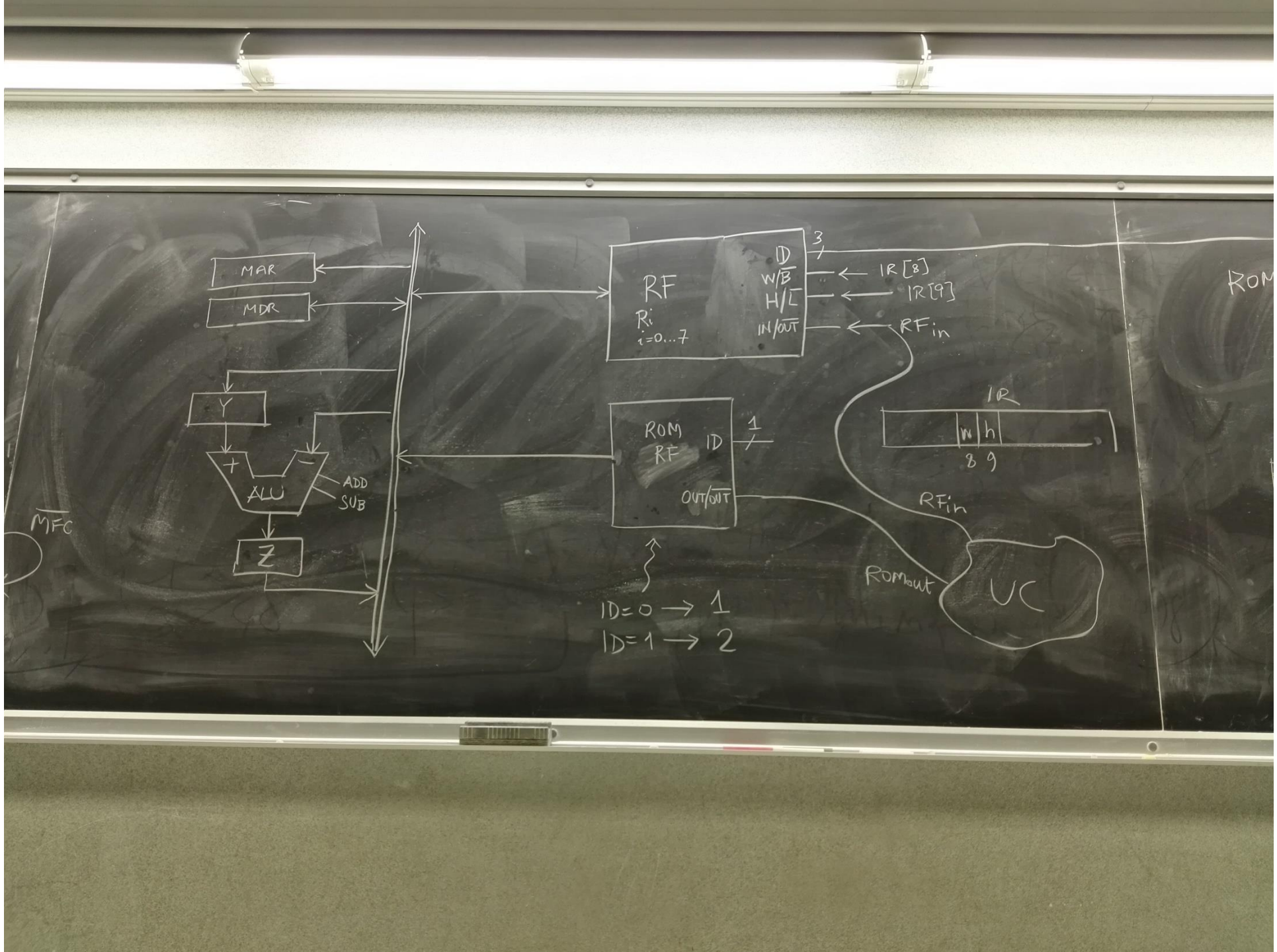

01
00
sequenza di reazione
Stato M1 (Q1,Q0)
ingresso per transizione = stato M2
x1x0 =
0 1 2 3
00 00 00 00
A A A A
0 2 1 3 1 0 3 2 (0)
01 11 01 01 11 11 11 01
B C B B C C C B
Q1Q0 = 00
01 v 10
Q1Q0 ≠ 00
Automa di controllo
A (00)
B (01)
C (11)
00 v 11
01 v 10
00
11
D (10) NON USATO
don't care

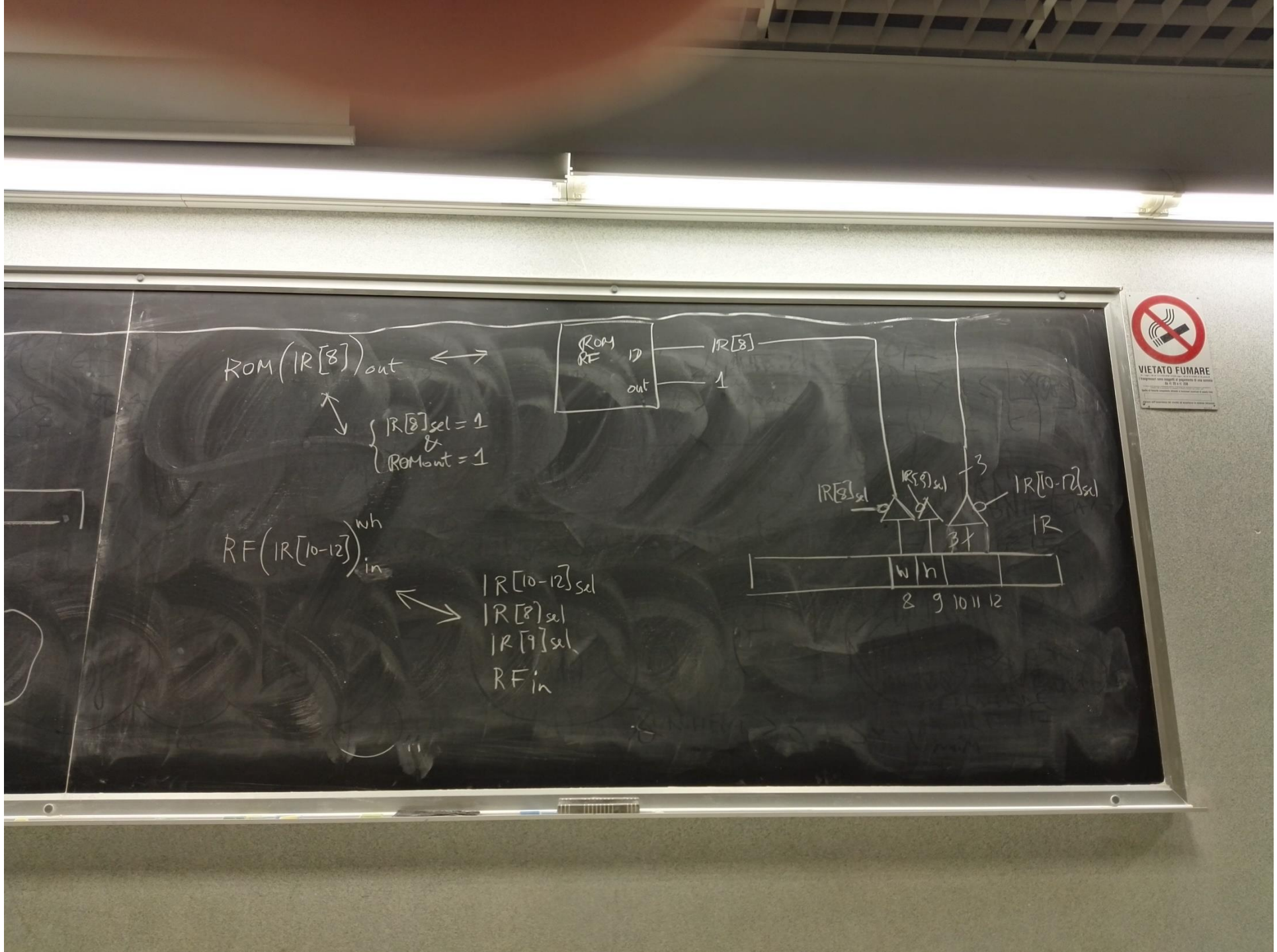

da
COMPITO
15/4/19
load <op>, <reg>
1 v 2 bytes
8 registri
S, B, I,
16 ind.
16 dati
AM
diretto
indiretto dir.
autoincr,
autodecr,
Ri
Ri.H
Ri.L
i++
--i
esempi di sintassi
load VAR, R3.H
load [R6], R1
load [R5]+, R2
load -[R3], R4.L

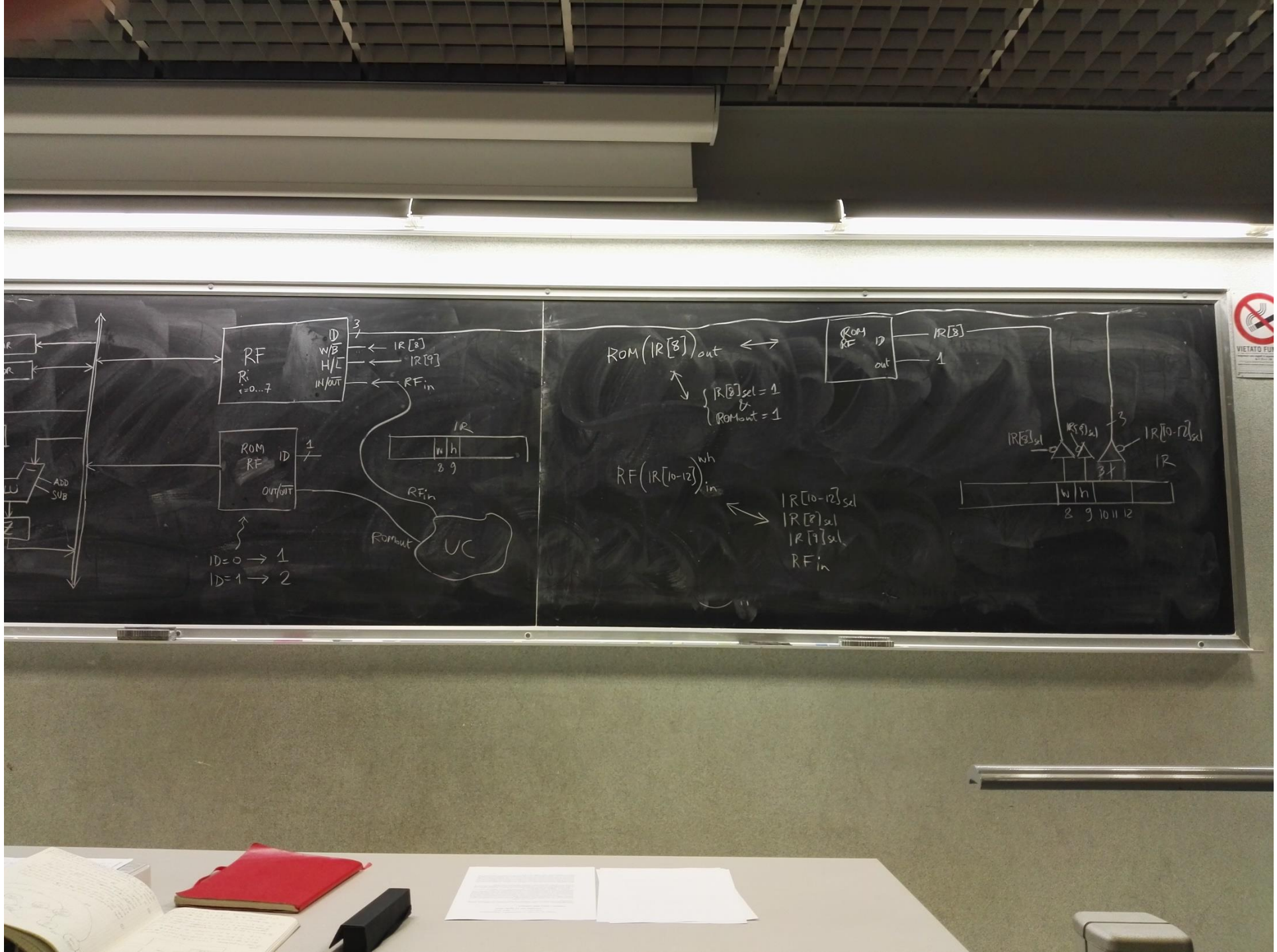

MAR
MDR
RF
R_i
i=0...7
ID
W/B
H/L
IN/OUT
3
Y
+
−
ALU
ADD
SUB
Z
ROM
RF
ID
OUT
1
ID=0 → 1
ID=1 → 2

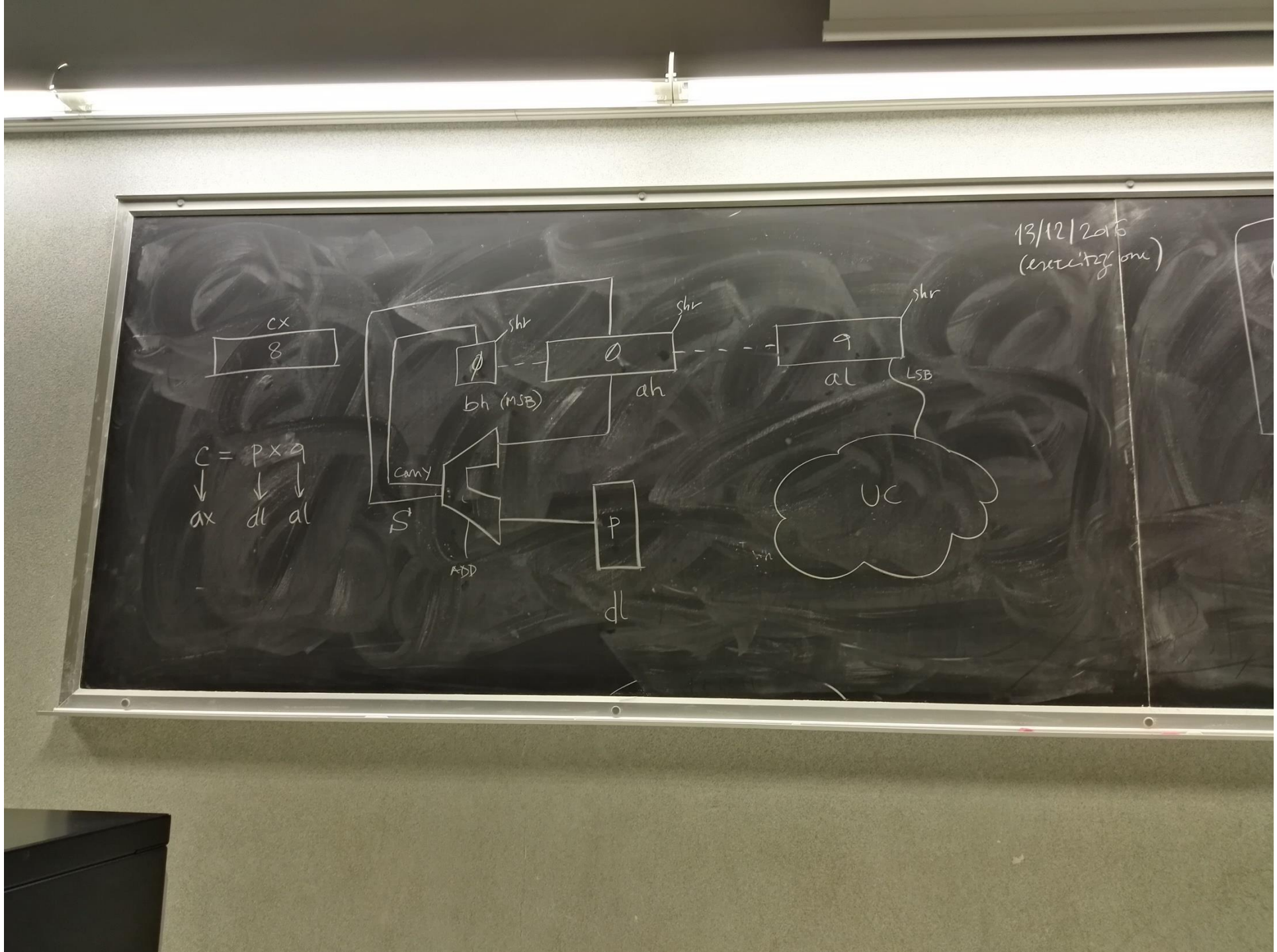

Codifica di macchina
0 5 6 7 8 9 10 12
opcode AM op w h ID reg
13 23 (AM diretto)
address op
13 15 AM ind autoinc autodec
ID reg. op
parte fissa
parte variabile: IDop
(a seconda di AMop)
VIETATO


da
COMPITO
15/4/19
FETCH
load & diretto
load & (ind v autoinc)
load & autoinc
MFC
MFC
MFC
IR[13-28]out
MARin
read
RF(IR[13-15])out
MARin
read
Yin
MFC & autoinc
MFC & ind
RF(IR[13-15])out
Yin
ROM(R[i])out
ADD
ROM(IR[i])out
SUB
MFC
MDRout
RF(IR[10-12])in
RF(IR[13-15])in
Zout
Zout
MARin
read
RF(IR[13-15])in
MFC
FETCH


MAR
MDR
Y
+
−
ALU
ADD
SUB
Z
MFC
RF
R_i
i=0...7
ID
W/B
H/L
IN/OUT
3
IR[8]
IR[9]
RFin
ROM
RF
ID
1
OUT/OUT
ID=0 → 1
ID=1 → 2
IR
w
h
8 9
RFin
ROMout
UC
ROM


ROM(IR[8])out
IR[8]sel = 1
&
ROMout = 1
ROM RF ID
out
IR[8]
1
RF(IR[10-12])wh in
IR[10-12]sel
IR[8]sel
IR[9]sel
RFin
IR[8]sel
IR[9]sel
3
IR[10-12]sel
IR
3↑
w
h
8 9 10 11 12
VIETATO FUMARE


RF
Ri
i=0...7
ID
W/B
H/L
IN/OUT
3
IR[8]
IR[9]
RFin
ROM
RF
ID
OUT/OUT
1
IR
w
h
8 9
RFin
ROMout
UC
ADD
SUB
ID=0 → 1
ID=1 → 2
ROM(IR[8])out
IR[8]sel = 1
&
ROMout = 1
RF(IR[10-12])wh in
IR[10-12]sel
IR[8]sel
IR[9]sel
RFin
ROM
RF
ID
out
IR[8]
1
IR[8]sel
IR[9]sel
IR[10-12]sel
3
3↑
IR
w
h
8 9 10 11 12


13/12/2016
(esercizio one)
cx
8
shr
shr
shr
0
0
9
bh (MSB)
ah
al
LSB
C = P × 9
ax dl al
carry
S
ADD
P
dl
UC

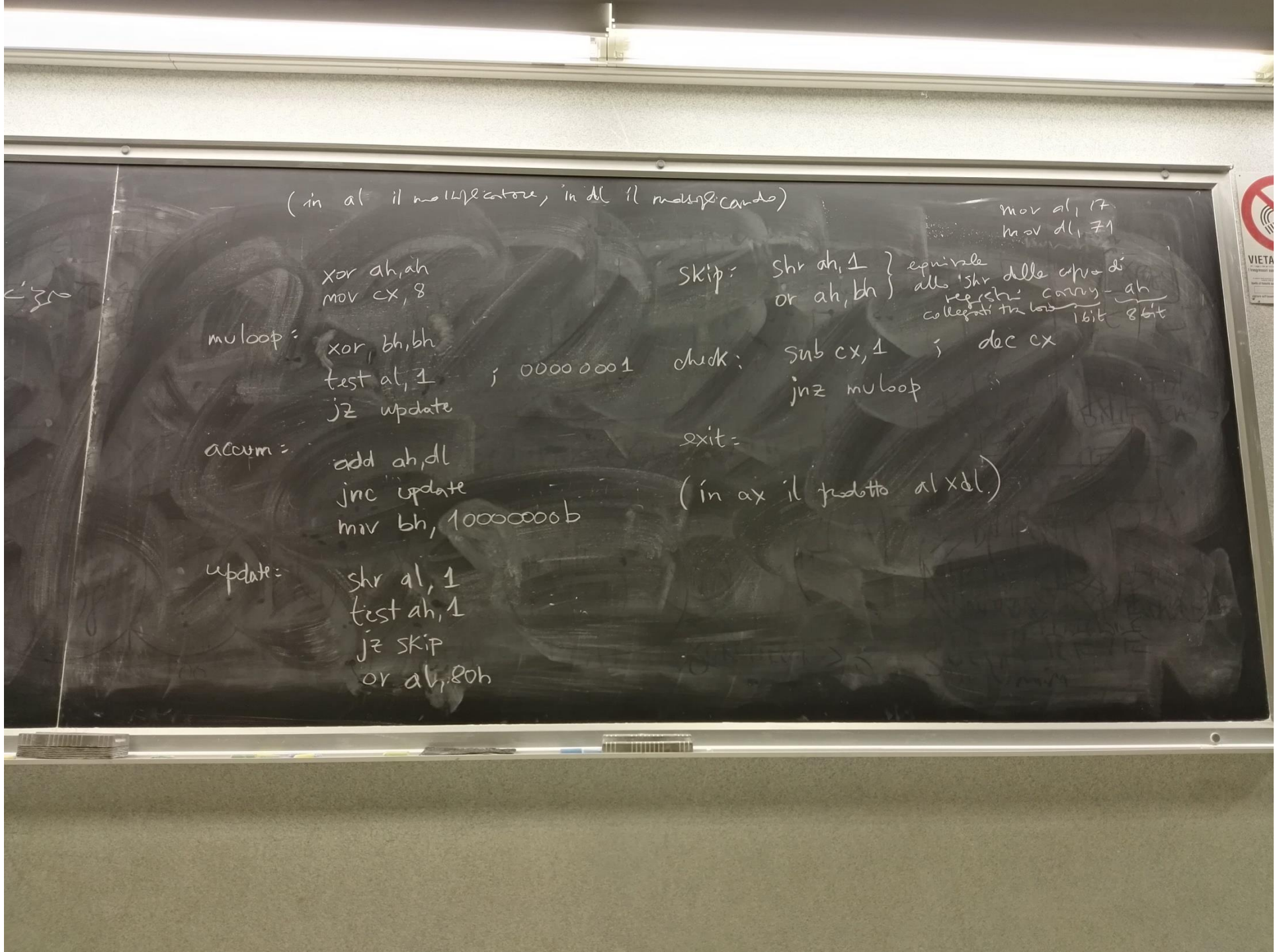
/2016
izione)

Ipotesi: l'8086 non ha la moltiplicazione
(falsa!)

⇒ realizzare un programma ASM 86
che moltiplica al e dl (8 bit)
e pone il risultato in ax (16 bit)

Studiare la
Sol. dell'esercizio
di R.L.
17/6/2019

$\sqrt{N} = (a, b)$


(in al il moltiplicatore, in dl il moltiplicando)
mov al, 17
mov dl, 71
xor ah,ah
mov cx, 8
muloop: xor bh,bh
test al, 1 ; 00000001
jz update
accum: add ah,dl
jnc update
mov bh, 10000000b
update: shr al, 1
test ah,1
jz skip
or ah,80h
skip: shr ah, 1
or ah, bh
equivale alla shr della cpu di registri collegati tra loro
carry 1bit
ah 8bit
check: sub cx, 1 ; dec cx
jnz muloop
exit:
(in ax il prodotto al x dl)