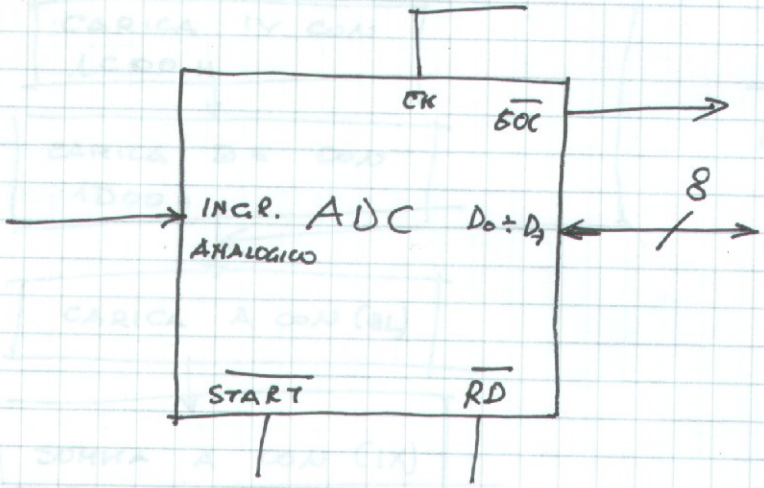
PROGRAMMA 51

Un µP Z80 controlla 128 sensori analogici tramite due multiplexer analogici ad 8 ingressi e due convertitori A/D; scrivere un programma che acquisisca i dati ogni minuto; si ha a disposizione un sottoprogramma RITARD. Acquisiti 100 dati, questi vengono inviati alla porta FOH e si ricomincia da capo.

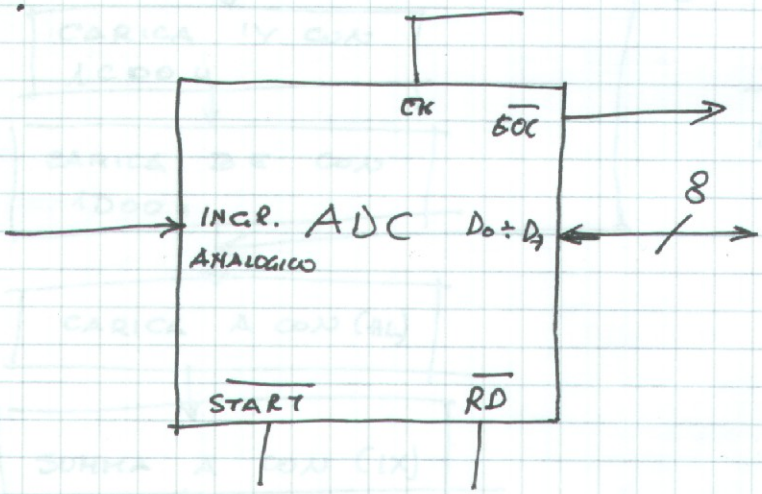
Questo esercizio implica delle considerazioni sull'hardware. Un convertitore A/D è un dispositivo elettronico che presenta un ingresso a cui giunge un segnale analogico il cui valore verrà convertito in un dato numerico. L'ADC presenta 8 linee di uscita digitali sulle quali scaricherà il dato numerico risultato della conversione. Queste uscite sono three-state in modo da consentire l'interfacciamento con il bus dati di un µP.



L'ADC presenta anche un ingresso di START per far partire la conversione e un ingresso di RD per leggere il risultato della conversione. Il processo di conversione è temporizzato da un segnale di clock la cui frequenza massima dipende dal tipo di ADC. La durata della conversione dipende dalla frequenza del clock ed è in genere espressa in N cicli di clock. In genere l'ADC

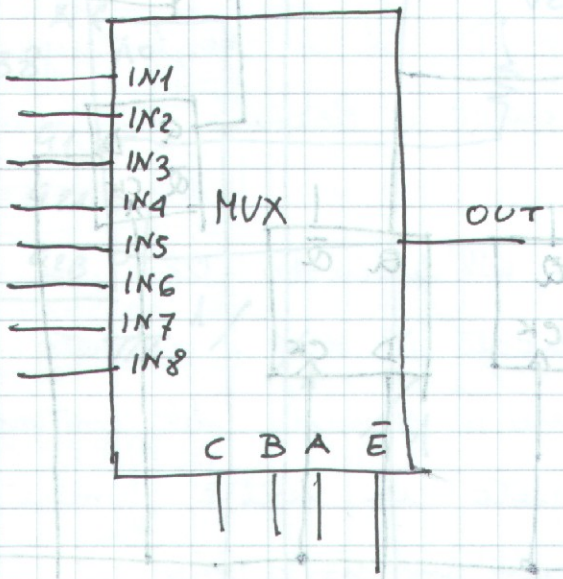
ad 8 ingressi e due convertitori A/D; scrivere un programma che acquisisca i dati ogni minuto; si ha a disposizione un sottoprogramma RITARD. Acquisiti 100 dati, questi vengono inviati alla porta F0H e si ricomincia da capo.

Questo esercizio implica delle considerazioni sull'hardware. Un convertitore A/D è un dispositivo elettronico che presenta un ingresso a cui giunge un segnale analogico il cui valore verrà convertito in un dato numerico. L'ADC presenta 8 linee di uscita digitali sulle quali scaricherà il dato numerico risultato della conversione. Queste uscite sono three-state in modo da consentire l'interfacciamento con il bus dati di un µP.



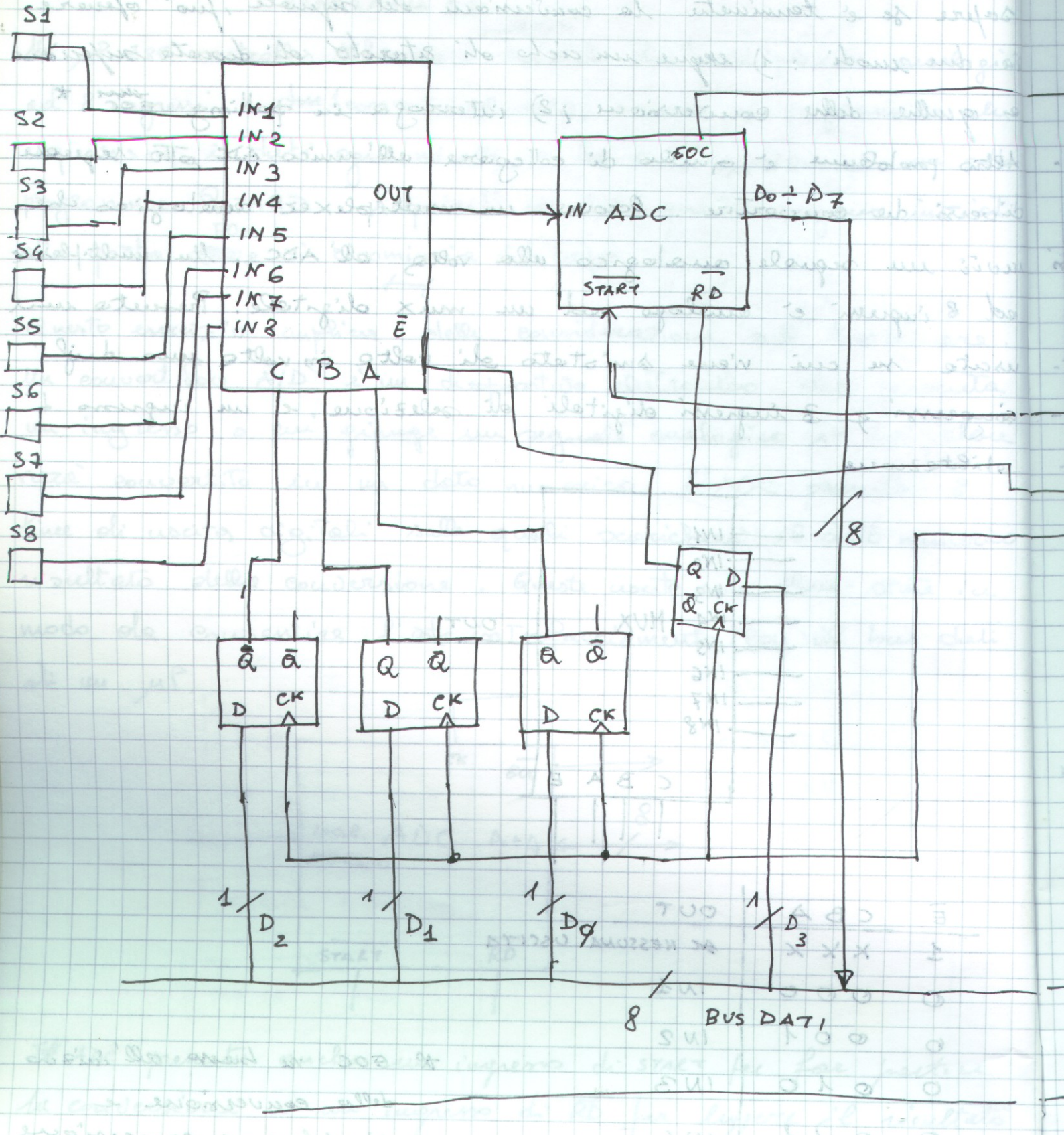
L'ADC presenta anche un ingresso di START per far partire la conversione e un ingresso di RD per leggere il risultato della conversione. Il processo di conversione è temporizzato da un segnale di clock la cui frequenza massima dipende dal tipo di ADC. La durata della conversione dipende dalla frequenza del clock ed è in genere espressa in N cicli di clock. In genere l'ADC presenta una uscita EOC che segnala la fine della conversione. Il µP, per

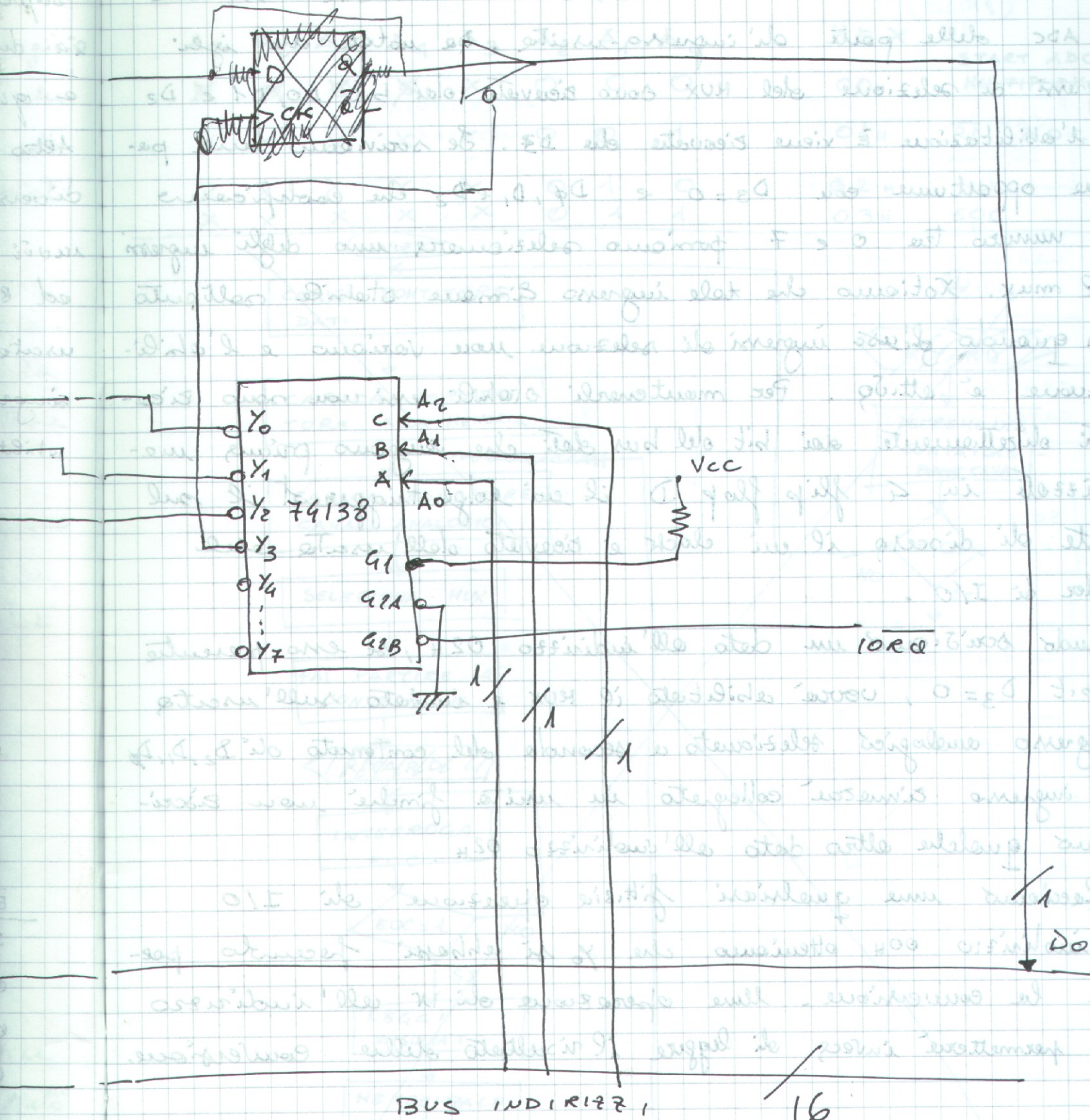
sapere se è terminata la conversione del segnale, può operare in due modi: 1) esegue un ciclo di ritardo di durata superiore a quella della conversione; 2) interroga in polling $\overline{EOC}$.*
Altro problema è quello di collegare all'unico ADC otto segnali diversi da convertire. Occorre un multiplexer analogico che invii un segnale analogico alla volta all'ADC. Un multiplexer ad 8 ingressi è analogo ad un mux digitale. Premesso una usata su cui viene suddiviso di volta in volta uno degli ingressi a 3 ingressi digitali di selezione e un ingresso di abilitazione



Ē	C B A	OUT
1	X X X	~~X~~ NESSUNA USCITA
0	0 0 0	IN1
0	0 0 1	IN2
0	0 1 0	IN3
0	0 1 1	IN4
⋮	⋮	⋮
0	1 1 1	IN8

* EOC va basso all'inizio della conversione e si rialza a conversione terminata


S1
S2
S3
S4
S5
S6
S7
S8
IN1
IN2
IN3
IN4
IN5
IN6
IN7
IN8
OUT
C B A
Ē
EOC
IN ADC
D0 ÷ D7
START
RD
8
Q D
Q̄ CK
Q̄ Q̄
D CK
Q Q̄
Q Q̄
1
1/ D2
1/ D1
1/ D0
1/ D3
8
BUS DATI


D
Q
CK
Q
Y0
Y1
Y2
Y3
Y4
Y7
74138
C
B
A
G1
G2A
G2B
A2
A1
A0
Vcc
IORQ
1
1
1
1
D0
BUS INDIRIZZI
16

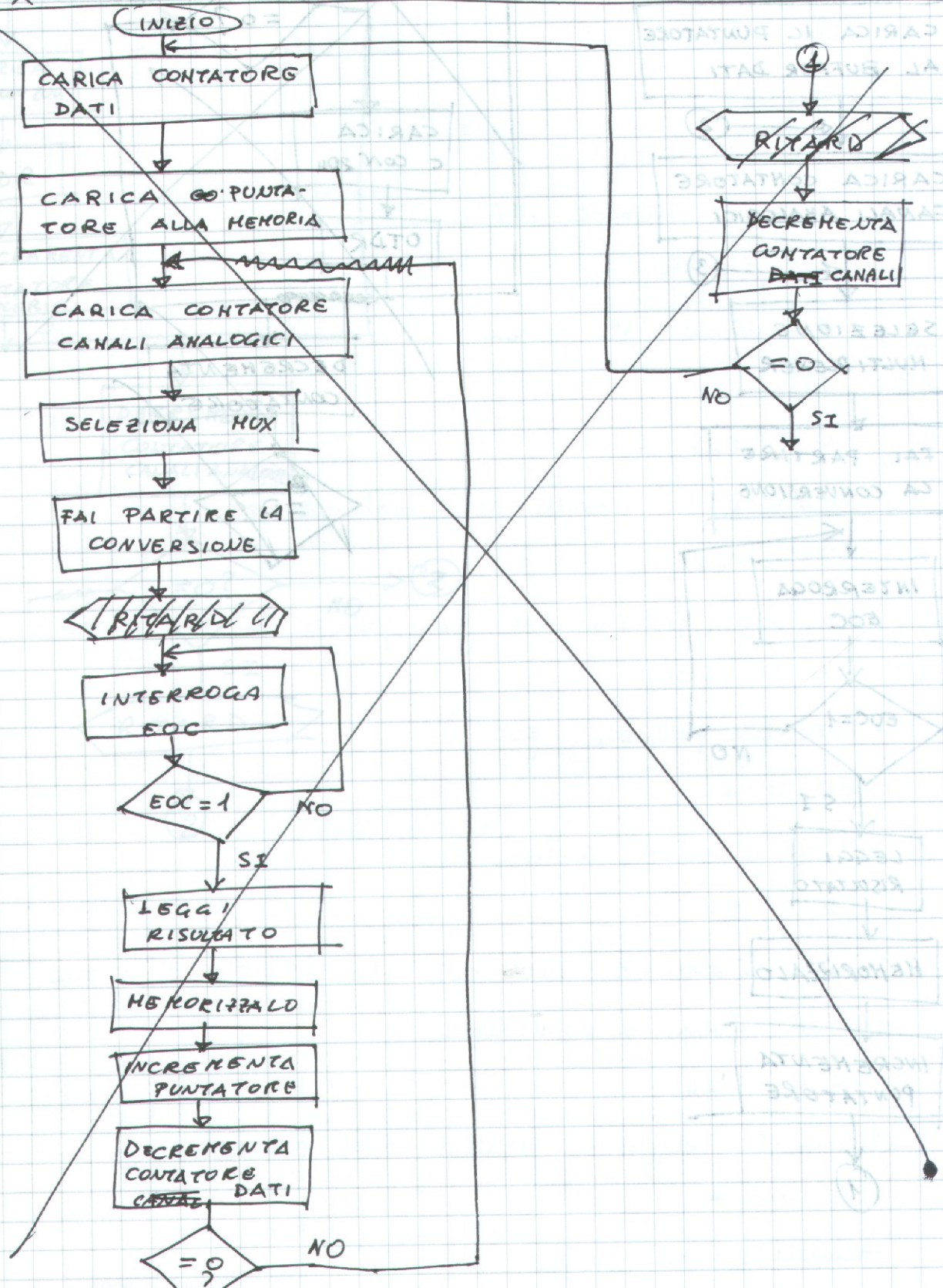
Nello schema vediamo che abbiamo reso sia il MUX che l'ADC delle porte di ingresso-uscita. Da notare che gli ingressi di selezione del MUX sono ricevuti dai bit D0, D1 e D2 e l'abilitazione E viene ricevuta da D3. Se scriviamo una parola opportuna con $D_3 = 0$ e D_0, D_1 e D_2 che codificano un numero tra 0 e 7 possiamo selezionare uno degli ingressi del mux. Notiamo che tale ingresso rimane stabile soltanto fin quando gli ingressi di selezione non variano e l'abilitazione è attiva. Per mantenerli stabili essi non sono ricevuti direttamente dai bit del bus dati che vengono prima memorizzati in 4 flip flop D il cui edge triggered il fronte di discesa il cui clock è ricevuto dall'uscita del decoder di I/O.

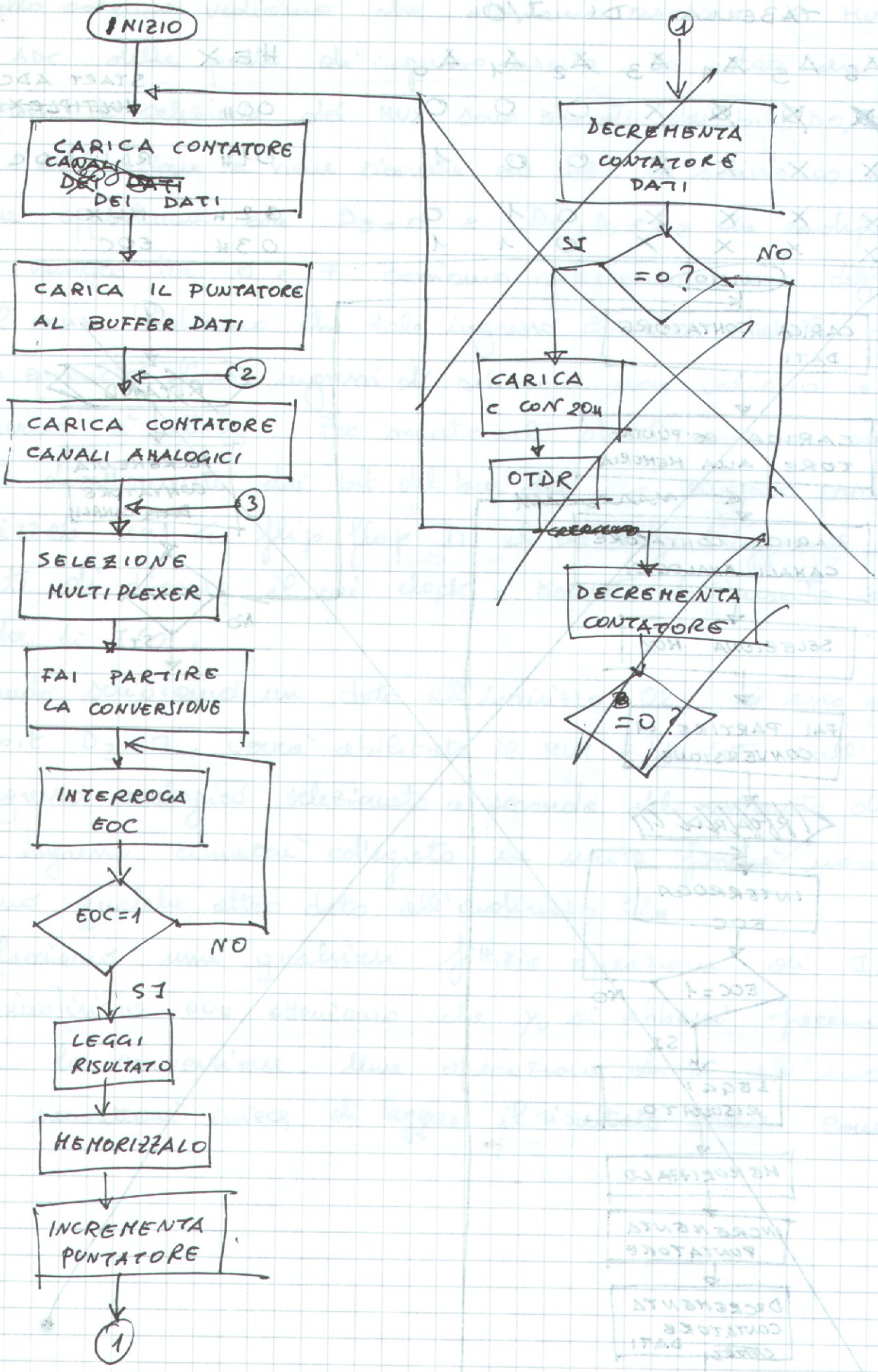
Quando scriviamo un dato all'indirizzo 02_H, se esso presenta il bit $D_3 = 0$, verrà abilitato il MUX e inviato sull'uscita l'ingresso analogico selezionato a seconda del contenuto di D_2, D_1. Tale ingresso rimarrà collegato in uscita finché non riscriveremo qualche altro dato all'indirizzo 02_H.

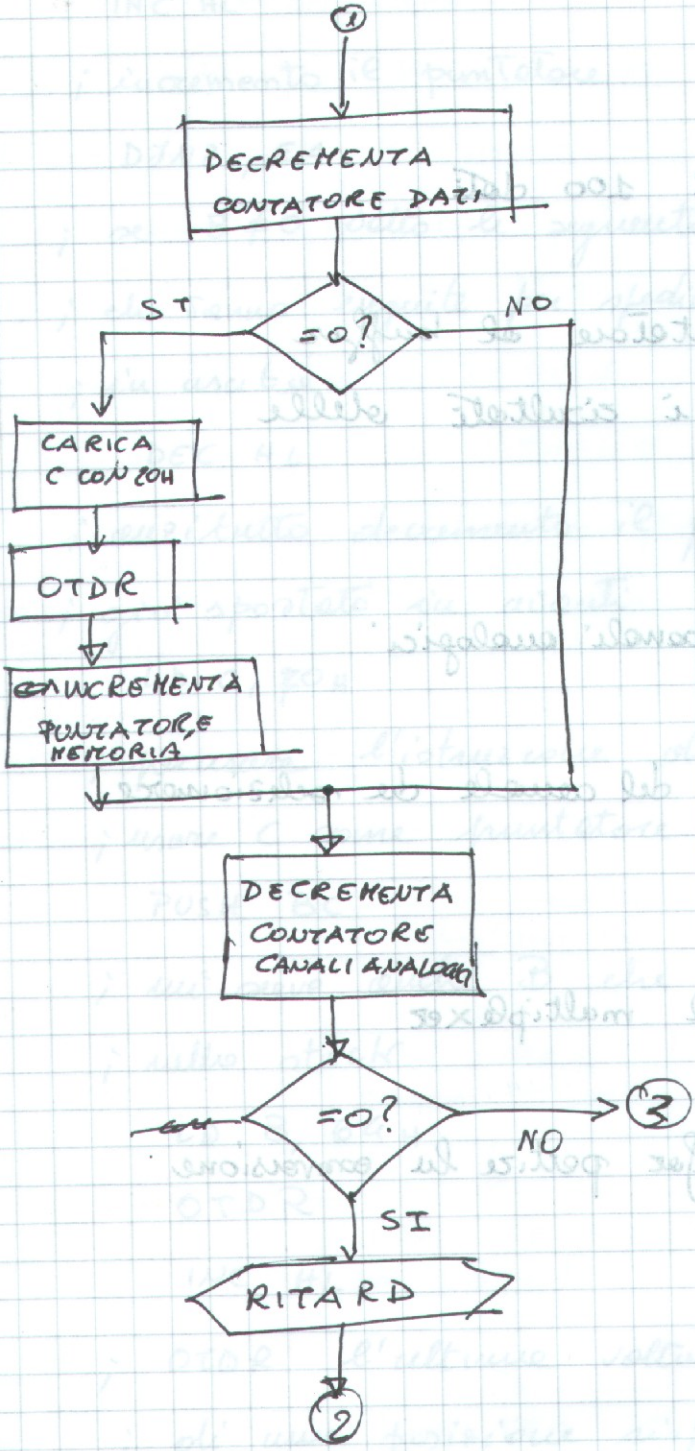
Se facciamo una qualsiasi fittizia operazione di I/O all'indirizzo 00_H otteniamo che Vo si abbassi facendo partire la conversione. Una operazione di IN all'indirizzo 01_H permetterà invece di leggere il risultato della conversione

TABELLA DI I/O

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	HEX	
~~X~~	~~X~~	~~X~~	~~X~~	X	0	0	0	00H	START ADC ~~MULTIPLEXER~~
X	X	X	X	X	0	0	1	01H	$\overline{RD}$ ADC
X	X	X	X	X	0	1	0	02H	MUX
X	X	X	X	X	0	1	1	03H	EOC




INIZIO
CARICA CONTATORE CANALI DEI DATI
CARICA IL PUNTATORE AL BUFFER DATI
2
CARICA CONTATORE CANALI ANALOGICI
3
SELEZIONE MULTIPLEXER
FAI PARTIRE LA CONVERSIONE
INTERROGA EOC
EOC=1
NO
SI
LEGGI RISULTATO
MEMORIZZALO
INCREMENTA PUNTATORE
1
1
DECREMENTA CONTATORE DATI
SI
=0 ?
NO
CARICA c CON 20μ
OTDR
DECREMENTA CONTATORE
=0 ?


1
DECREMENTA CONTATORE DATI
=0?
SI
NO
CARICA C CON 20H
OTDR
INCREMENTA PUNTATORE MEMORIA
DECREMENTA CONTATORE CANALI ANALOGICI
=0?
3
NO
SI
RITARD
2

```
        ORG  XXXX;
        LD   B, 64H;
    ; B è il contatore dei 100 dati
        LD   HL, 1800H
    ; usiamo HL come puntatore al buffer
    ; che riempiremo con i risultati delle
    ; conversioni
E2 :    LD   D, 08H
    ; D è il contatore dei canali analogici
        LD   E, 00H
    ; E contiene il numero del canale da selezionare
E3 :    LD   A, E
        OUT (02H), A
    ; selezione il canale del multiplexer
        OUT (00H), A
    ; scrittura fittizia per far partire la conversione
E1 :    IN   A, (03H)
    ; lettura per prelevare EOC
        BIT  0, A
    ; il bit 0 di A contiene EOC
        JR   Z, E1
    ; se flag di zero = 1 vuol dire che il bit 0 è a zero
    ; quindi ritorno a controllare
        IN   A, (01H)
    ; altrimenti leggo il risultato della conversione
        LD   (HL), A
    ; memorizzo nel buffer
```

```
        INC HL
    ; incremento il puntatore
        DJNZ, E4
    ; se B ≠ 0 salto le seguenti istruzioni
    ; che sono eseguite per spedire i 100 dati accumulati
    ; in uscita
        DEC HL
    ; anzitutto decremento il puntatore che bravo
    ; già spostato in avanti
        LD C, 70H
    ; per usare l'istruzione di out ricordo devo
    ; usare C come puntatore alla porta di I/O
        PUSH BC
    ; mi serve anche B che salvo momentaneamente
    ; nello stack
        LD B, 64H
        OTDR
        INC HL
    ; OTDR l'ultima volta sposta HL all'indietro
    ; di una posizione rispetto all'inizio del
    ; buffer
        POP BC
    ; ripristino B
E4 : DEC D
    ; decremento contatore cicli esterno
        INC E
    ; incremento l'indirizzo del ciclo esterno
```

```
CALL   RITARD
; aspetta un minuto
JP , E2
```