

UNIVERSITA' DEGLI STUDI DI BOLOGNA

Facoltà di Scienze Fisiche, Matematiche e Naturali

Corso di Laurea in Fisica

**APPLICAZIONI DEL MICROPROCESSORE
INTEL 8080**

Tesi di Laurea

di

PAOLO MULAZZANI

Relatori:

Chiar.mo Prof Ing. GIANFRANCO SINIGAGLIA

Dott. ENZO GANDOLFI

Anno Accademico 1977 - 78

UNIVERSITA' DEGLI STUDI DI BOLOGNA

Facoltà di Scienze Fisiche, Matematiche e Naturali

Corso di Laurea in Fisica

**APPLICAZIONI DEL MICROPROCESSORE
INTEL 8080**

Tesi di Laurea

di

PAOLO MULAZZANI

Relatori:

Chiar.mo Prof Ing. GIANFRANCO SINIGAGLIA

Dott. ENZO GANDOLFI

Anno Accademico 1977 - 78

I N D I C E

- INTRODUZIONE	I
- CONVERTITORE DIGITALE-ANALOGICO	1
Conversione analogico-digitale	5
-DESCRIZIONE DEL CRAMER KIT E DELLE MODIFICHE APPORTATE	17
- RICOSTRUZIONE DI FUNZIONI PERIODICHE SECONDO FOURIER	22
Istruzioni per l'uso del programma	31
Esempi di funzioni, tabelle e grafici	33
-MEDIA DI IMPULSI QUASI PERIODICI DI TIPO BATTITO CARDIACO	47
Uso pratico del programma	58
-APPENDICE	64
-CONCLUSIONE	90
-BIBLIOGRAFIA	91

INTRODUZIONE

Nella primavera del 1976 l'Istituto di Fisica ha acquistato un microprocessore (μ P) sotto forma di kit di montaggio venduto dalla CRAMER (CRAMER KIT). Lo scopo era di tenere aggiornato l'insegnamento dell'elettronica, visto che i μ P erano un prodotto nuovo e ancora poco conosciuto. Il loro specifico campo di applicazione è controllare continuamente e in tempo reale dei processi fisici attraverso opportuni sensori e di fornire delle risposte, a programma, in funzione dei cambiamenti che il processo stesso subisce. La logica cablata, o fissa, o hardware, potrebbe fornire le stesse prestazioni ma i μ P permettono di poter variare il tipo di controllo senza rifare il progetto e il montaggio: semplicemente sostituendo il programma inserito o la memoria che lo contiene.

Il nostro lavoro è partito dal CRAMER KIT già montato e funzionante. Però le uniche possibilità che aveva

di comunicare con l'utente erano degli interruttori per scegliere funzioni o per introdurre dei dati e due display per visualizzare altri. Il primo passo é stato quindi quello di modificarlo e renderlo capace di ricevere e di produrre tensioni analogiche.

Montando un convertitore Digitale-Analogico si realizza la seconda richiesta molto semplicemente. E' nel dotarlo di un convertitore Analogico-Digitale che si vede la flessibilit  dei μ P. Innanzitutto la conversione stessa (da tensione analogica a numero binario) é possibile eseguirla con il μ P stesso. Inoltre utilizzando due metodi diversi di conversione, contemporaneamente presenti in memoria sotto forma di due distinti programmi, e mandando in esecuzione l'uno o l'altro alternativamente abbiamo potuto scegliere il metodo migliore nei vari casi. La seconda applicazione del CRAMER KIT é l'uso del convertitore D-A in uscita per visualizzare, a scopo didattico, su registratore scrivente su carta o su oscilloscopio la somma di un numero qualunque di funzioni sinu-

soidali di frequenza, ampiezza e sfasamento qualunque, da impostare come INPUT. Tipicamente si può scegliere un certo numero di termini dello sviluppo in serie di Fourier di una funzione periodica e studiare la convergenza delle serie di Fourier di funzioni diverse. In Istituto esiste già una apparecchiatura che produce e somma i primi cinque termini della serie di Fourier dell'onda quadra, ma le limitazioni nella scelta della funzione e nel numero di armoniche sommabili sono evidenti.

Lo spunto per la terza applicazione proviene dalle seguenti considerazioni. In cardiologia si studiano i tracciati dell'elettrocardiogramma, del fonocardiogramma e dell'ecocardiogramma. Da tutti questi tracciati si nota una mancanza di periodicità degli impulsi e anche una leggera variazione della larghezza dell'impulso dovuto anche alle irregolarità nella durata della sistole. A parte i tracciati dell'ecocardiogramma, che presentano problematiche più complesse, per gli altri due potrebbe

IV

essere utile avere un sistema che accetti come INPUT un certo numero di impulsi, che, ripetiamo, sono non esattamente periodici nè tutti della stessa durata, ne faccia la media e la visualizzi in modo stabile su un apposito ^{regi} registratore. Tutto questo in tempi brevissimi purchè le caratteristiche del segnale in esame non vari in modo macroscopico. Il CRAMER KIT è stato perciò programmato per registrare, in generale, un numero qualunque di impulsi entranti alla frequenza di circa 1 Hz, di forma leggermente variabile, farne la media con una risoluzione programmabile e visualizzare il risultato su oscilloscopio o su registratore a pennino scrivente.

CONVERTITORE DIGITALE-ANALOGICO

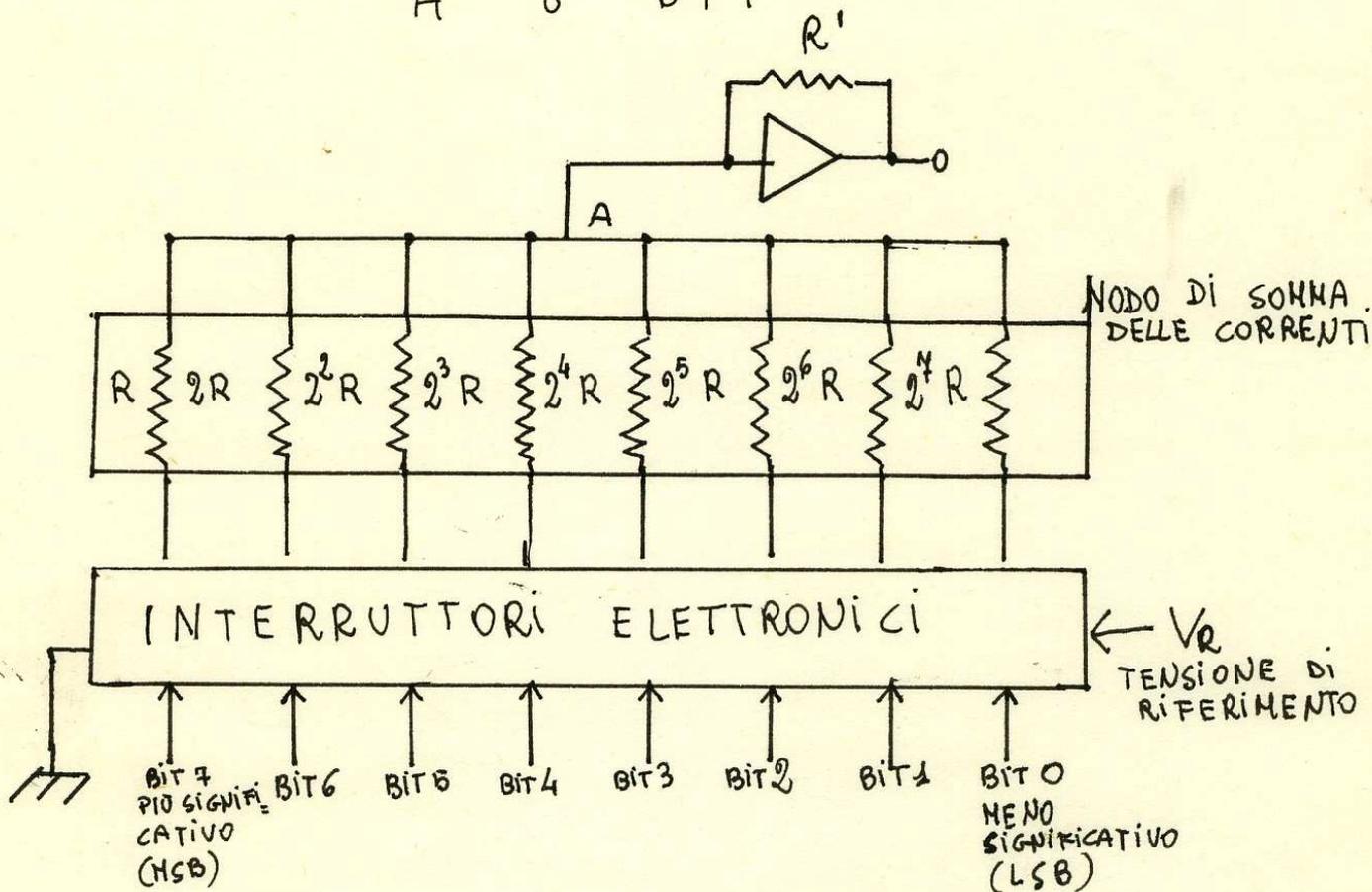
Un convertitore digitale-analogico (DAC) é un sistema che accetta una configurazione di bit e la converte in una tensione (o corrente) analogica.

Di solito il dato in ingresso é presentato sotto forma di un numero binario puro e l'uscita analogica é proporzionale al valore che la configurazione assume se considerata come un numero scritto in base due.

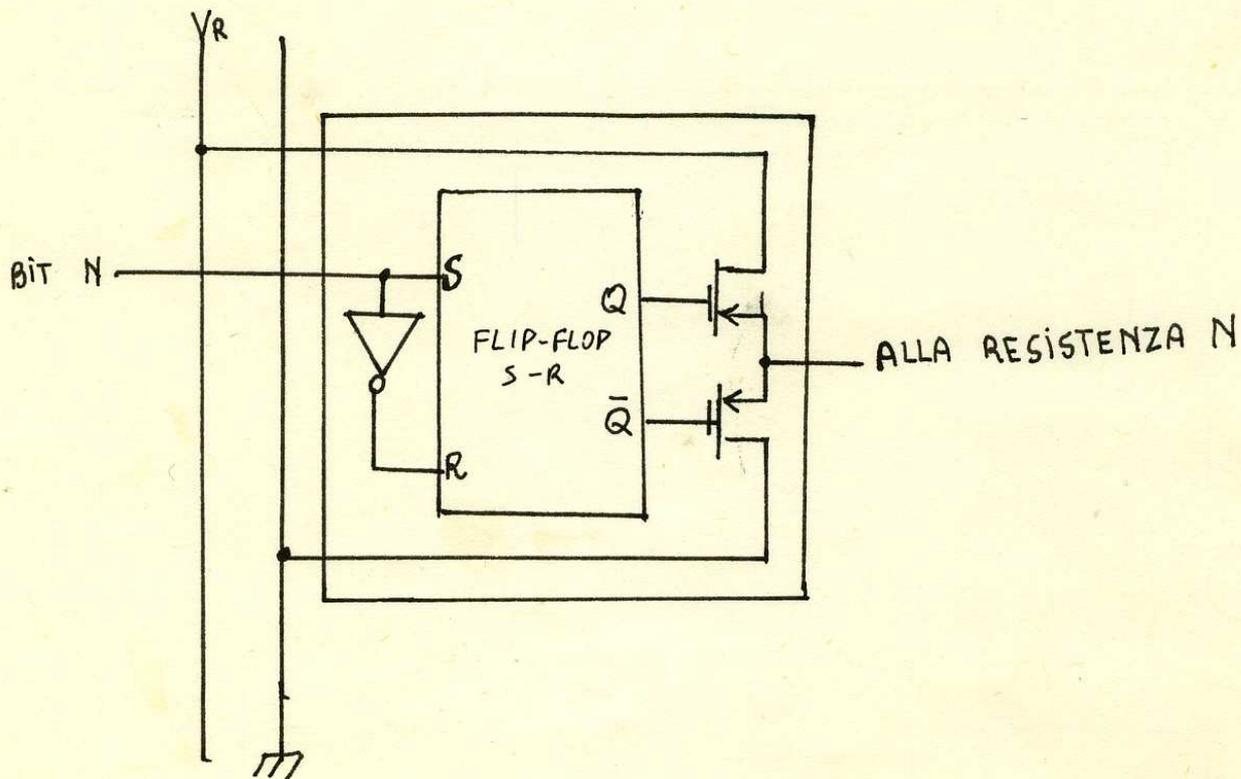
Sono possibili in ingresso altri codici oltre al binario puro : complemento due, BCD che ha una notevole ridondanza , l'offset binary, ecc. Il sistema usato per la conversione consiste in una rete di resistenze che ha il compito di pesare ciascun bit a seconda della posizione occupata. Uno schema di un DAC a 8 bit é riportato a tavola I. Ciascun interruttore connette il filo a V_r se il bit corrispondente é=1; lo connette a massa se il bit é=0. Così nel nodo A (massa virtuale) confluiscono delle correnti date dalla somma:

TAVOLA I

SCHEMA DI UN CONVERTITORE D-A
A 8 BIT



UN FLIP-FLOP MOS E DUE MOSFET COSTITUISCONO CIASCUN INTERRUTTORE.



$$(1) \quad V_R \left(\frac{b_x}{R} + \frac{b_6}{2R} + \frac{b_5}{2^2R} + \frac{b_4}{2^3R} + \frac{b_3}{2^4R} + \frac{b_2}{2^5R} + \frac{b_1}{2^6R} + \frac{b_0}{2^7R} \right)$$

dove b_x, b_6, \dots hanno il valore logico 0 o 1 e letti in

questo ordine formano il numero binario in ingresso.

Mettendo in evidenza $\frac{1}{2^x R}$ otteniamo:

$$(2) \text{ (corrente totale) } I = \frac{V_R}{2^x R} (b_x \cdot 2^x + b_6 \cdot 2^6 + b_5 \cdot 2^5 + b_4 \cdot 2^4 + b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2 + b_0)$$

La quantità $\frac{V_R}{2^x R}$ è costante per lo stesso componente, perciò

la corrente in A è proporzionale alla quantità fra paren-

tesi che è appunto il valore numerico della configurazione

di bit in ingresso.

L'operazionale che segue, producendo una tensione

$V_{out} = -\frac{I}{R'}$, aggiunge un $-\frac{1}{R'}$ che non altera la precedente

proporzionalità ma permette di ottenere una tensione, e

non una corrente, in uscita proporzionale al numero in

ingresso. Va aggiunto che il più grave difetto di un DAC

del genere è dato dalla grande influenza sull'uscita che

ha un possibile scarto nei valori delle resistenze, soprat-

tutto in quelle meno elevate che forniscono più corrente.

Per questo tutti i costruttori si servono di una rete di

resistenze costruita diversamente, che usa il doppio di

resistenze ma di valore R e 2R soltanto, detta LADDER e mostrata a Tavola II.

Ciascun nodo a,b,c,... si trova al centro di una rete di resistenze equivalente alla stella mostrata sotto.

Nel nodo h quando l'input è 1000 0000, ci sono $\frac{V_r}{3}$.

Il guadagno dell'operazionale è $-\frac{3R}{2R}$ e la tensione V_{out}

risulta:

$$(3) \quad V_{out} = \frac{V_R}{3} \left(-\frac{3R}{2R} \right) = -\frac{V_R}{2}$$

Considerazioni simili per gli altri nodi forniscono pesi per gli altri bit ognuno la metà del precedente.

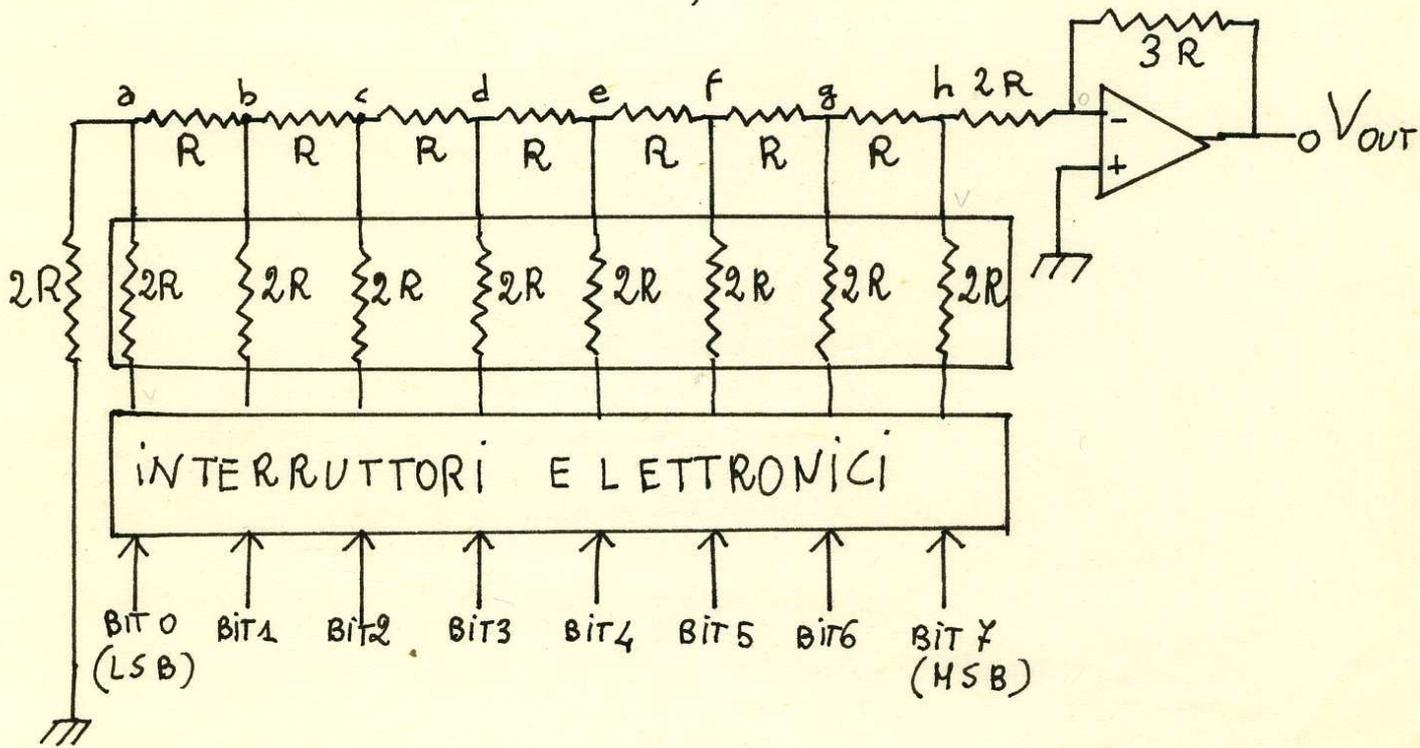
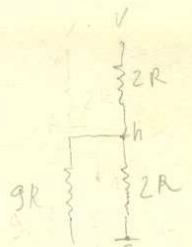
Quando si deve usare un convertitore D-A le caratteristiche più importanti da tenere presenti per l'acquisto presentano un duplice aspetto.

A. Sono da considerare attentamente fattori come:

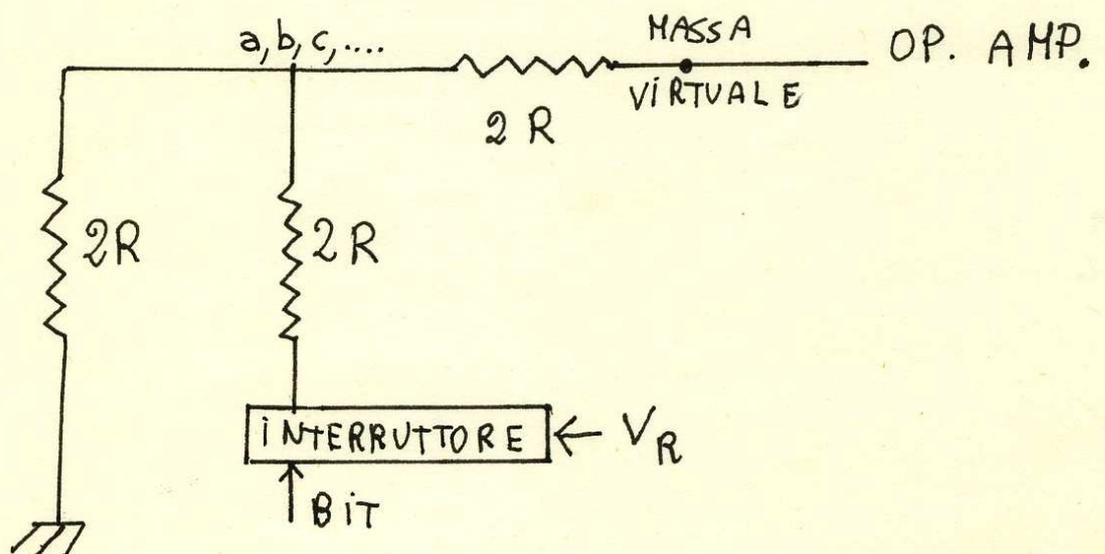
livello del segnale in uscita, precisione fornita, velocità di conversione, alimentazioni richieste per il funzionamento ottimale, fluttuazioni dovute alla temperatura, ingresso TTL compatibile, codice richiesto in input.

B. E' importante una conoscenza sicura di cosa

TAVOLA II
 CONVERTITORE D-A CHE USA UN
 LADDER R, 2R



RETE EQUIVALENTE PER OGNI BIT
 SETTATO



significa per il costruttore l'insieme delle caratteristiche fornite. Non è detto che due costruttori dicano la stessa cosa quando scrivono numeri identici per definire lo stesso parametro. Non esistendo criteri unificati è meglio stare più stretti del necessario nelle richieste di caratteristiche di precisione.

Nel nostro caso i due DAC utilizzati hanno queste caratteristiche principali.

Alimentazione nominale	+15 e -15 V.
Funzionano fino a	+13 e -13 V.
Tempo per passare da 0 V. (dato = 0) alla tensione massima (dato = 11111 11111) (SETTLING TIME)	5 μ sec.
Tensione in uscita	da 0 a -10 V.
Numero di bit	10
Codice	binario puro

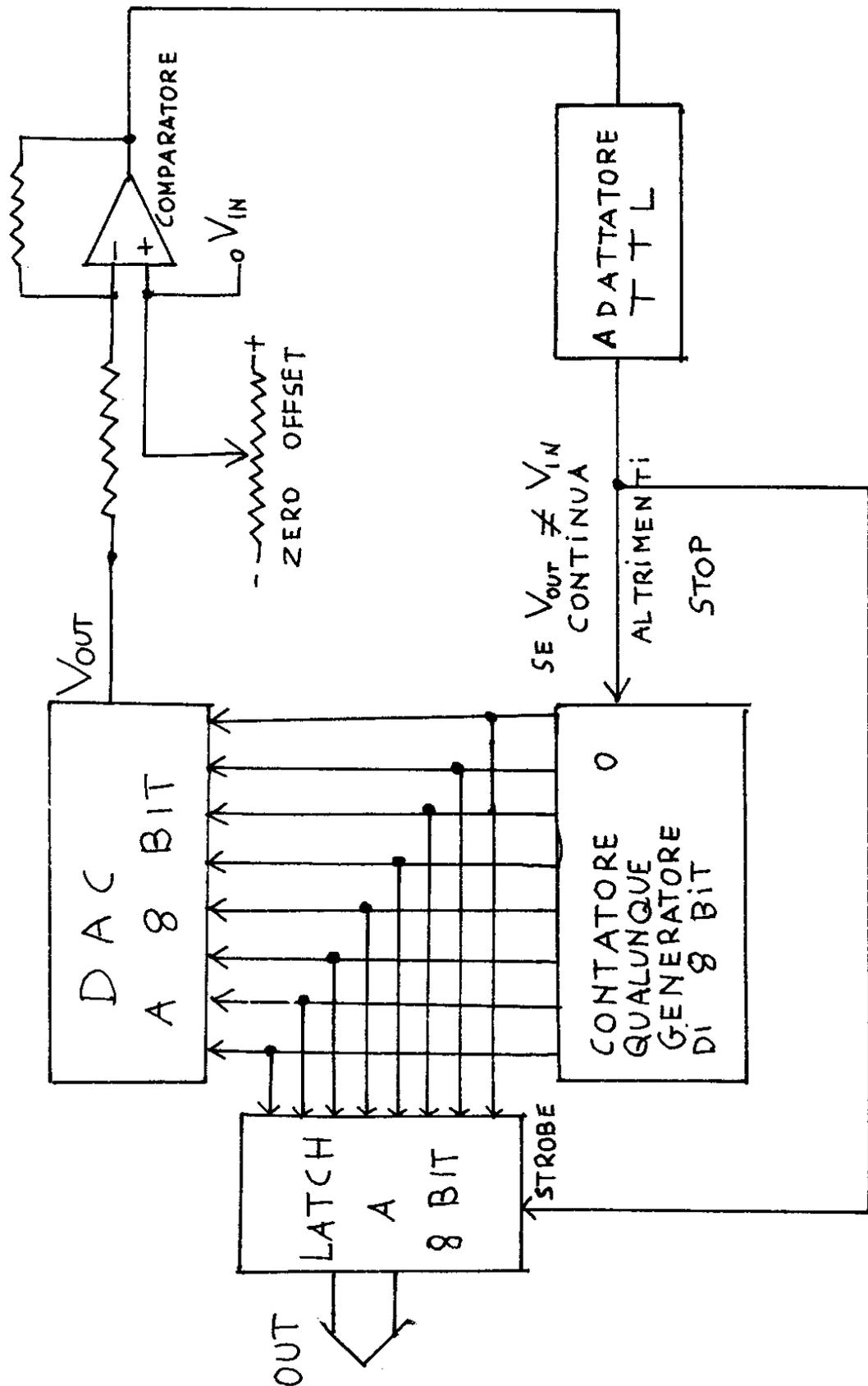
CONVERSIONE ANALOGICO-DIGITALE

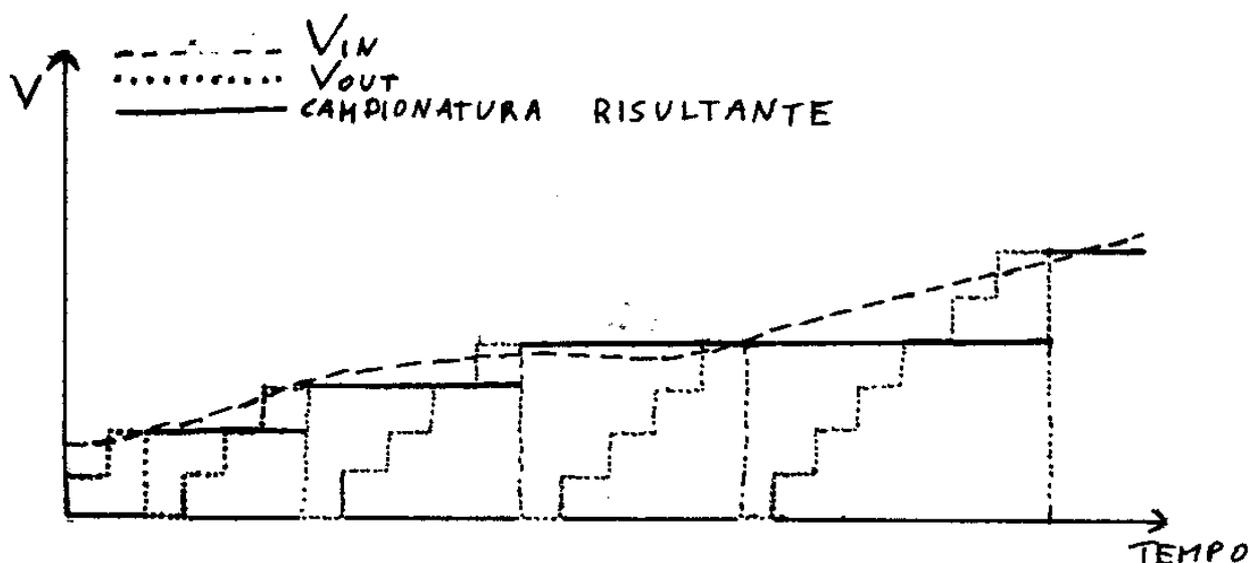
Produrre un sistema che faccia la conversione inversa non è così immediato. Si usa un DAC montato come a Tavola III. Il contatore presenta all'ingresso del DAC un numero binario. Il DAC lo converte in V_{out} . Il comparatore la confronta con V_{in} in ingresso e finchè sono apprezzabilmente diverse invia, attraverso l'adattatore TTL, un opportuno bit che informa il contatore che deve inviare al DAC un nuovo numero da provare. Quando il numero è tale che V_{out} è uguale a V_{in} il bit di risposta cambia, il contatore viene fermato e contemporaneamente il LATCH fornisce, congelata, la configurazione di bit trovata. I metodi più usati per variare i bit di prova all'ingresso del DAC sono i tre seguenti.

a) Tecnica ad inseguimento.

Come generatore di bit si usa un normale contatore inizialmente azzerato. Lo si incrementa di uno ad ogni passo finchè V_{out} raggiunge V_{in} . Si ferma il contatore si preleva il numero dal LATCH e si riparte da zero per il successivo campionamento.

TAVOLA III
 SCHEMA DI UN CONVERTITORE
 ANALOGICO-DIGITALE A 8 BIT

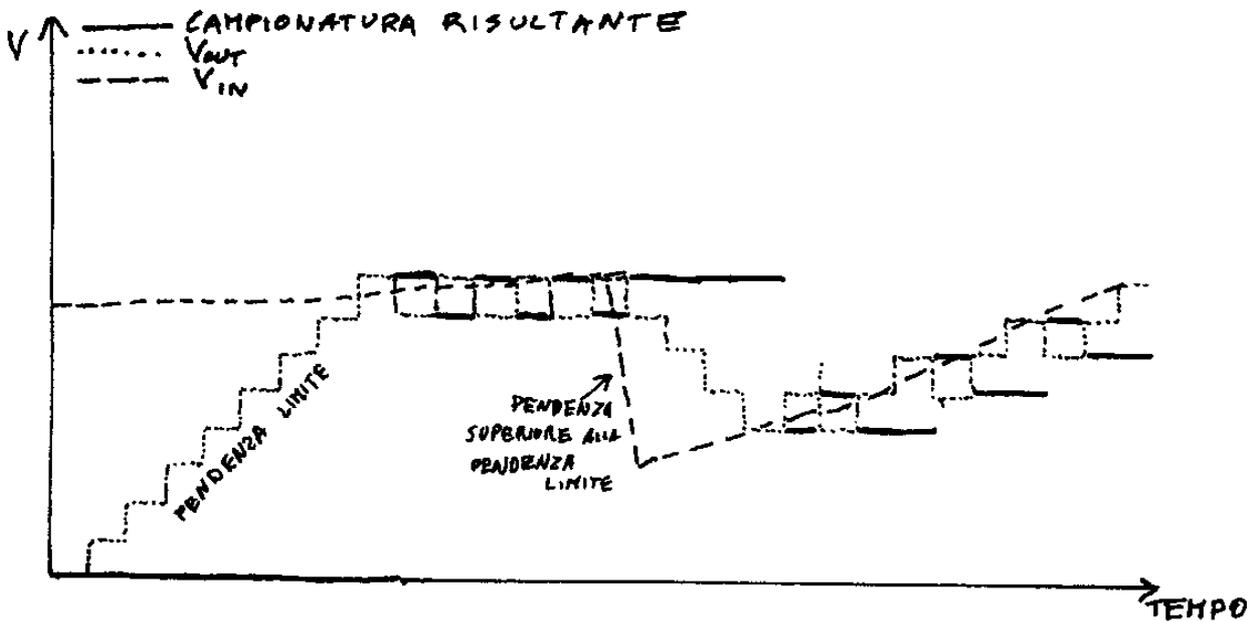




Questo sistema ha un tempo di ricerca che dipende direttamente da V_{in} e dal numero di bit (dalla risoluzione) che si usano.

b) Tecnica ad aggancio.

Per migliorare notevolmente la velocità è sufficiente non ripartire da zero ogni volta. Bisogna però usare un contatore di tipo avanti-indietro. Una volta che V_{out} ha superato V_{in} il comparatore cambia il senso di marcia al contatore e V_{out} diminuisce finchè non scende al di sotto di V_{in} . Il conteggio allora riprende in avanti. Il risultato è che V_{out} resta agganciata un pò' sopra e un pò' sotto a V_{in} e la conversione avviene quasi di continuo.



Si può notare che a causa del ritardo dovuto principalmente al DAC, in misura minore al comparatore e all'adattatore TTL, l'inversione del senso di marcia del contatore avviene dopo un certo tempo. Di qui la caratteristica che ha questo tipo di conversione: non poter seguire una V_{in} che presenta una variazione superiore ad un certo limite.

c) Tecnica ad approssimazioni successive. E' necessario sostituire il contatore con un generatore di bit più versatile e pilotabile a logica (software o hardware).

Dalla figura che segue si vede che V_{in} viene approssimata successivamente **per** V_{out} tramite la somma o la sottrazione di potenze decrescenti di 2.

Nella logica binaria il procedimento che gli equivale è questo.

All'ingresso del DAC, si comincia col porre uguale a 1 (SET) il bit più significativo e i rimanenti uguali a 0 (RESET).

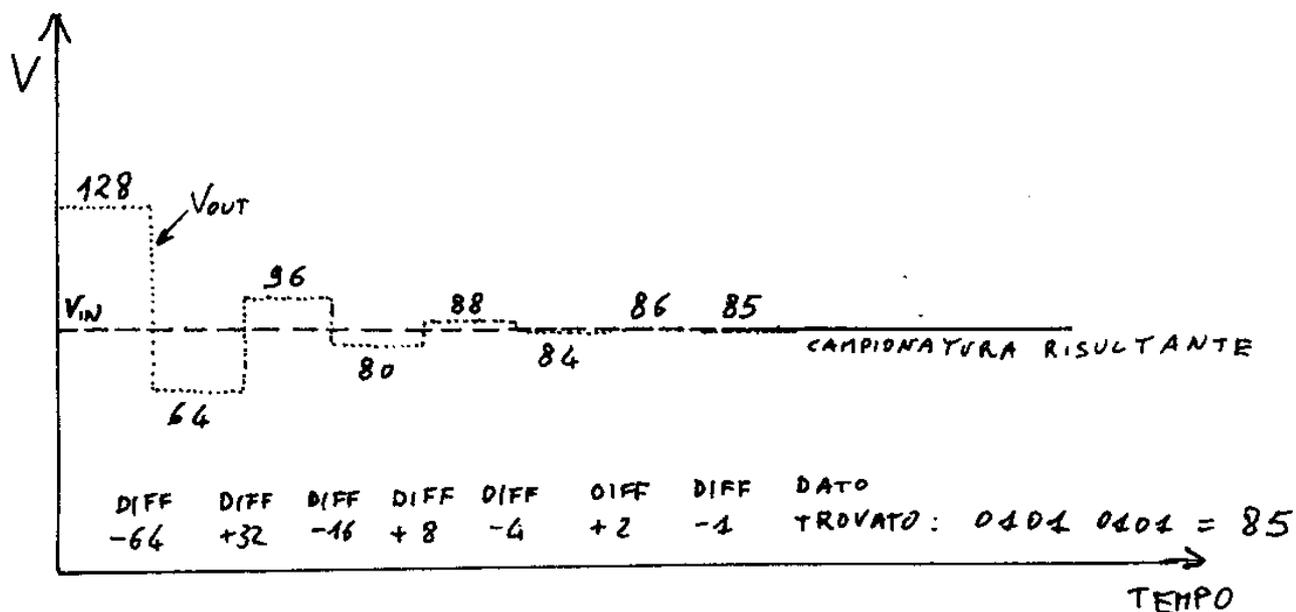
Lo si azzerava subito dopo se V_{out} è già superiore a V_{in} , altrimenti si passa a SETtare il bit successivo, RESETtarlo se

il comparatore lo richiede (V_{out} maggiore di V_{in}) e così fino

al bit meno significativo. Dopo N passi di SET o di SET/RESET

(con un DAC a N bit) la configurazione finale è la misura bi-

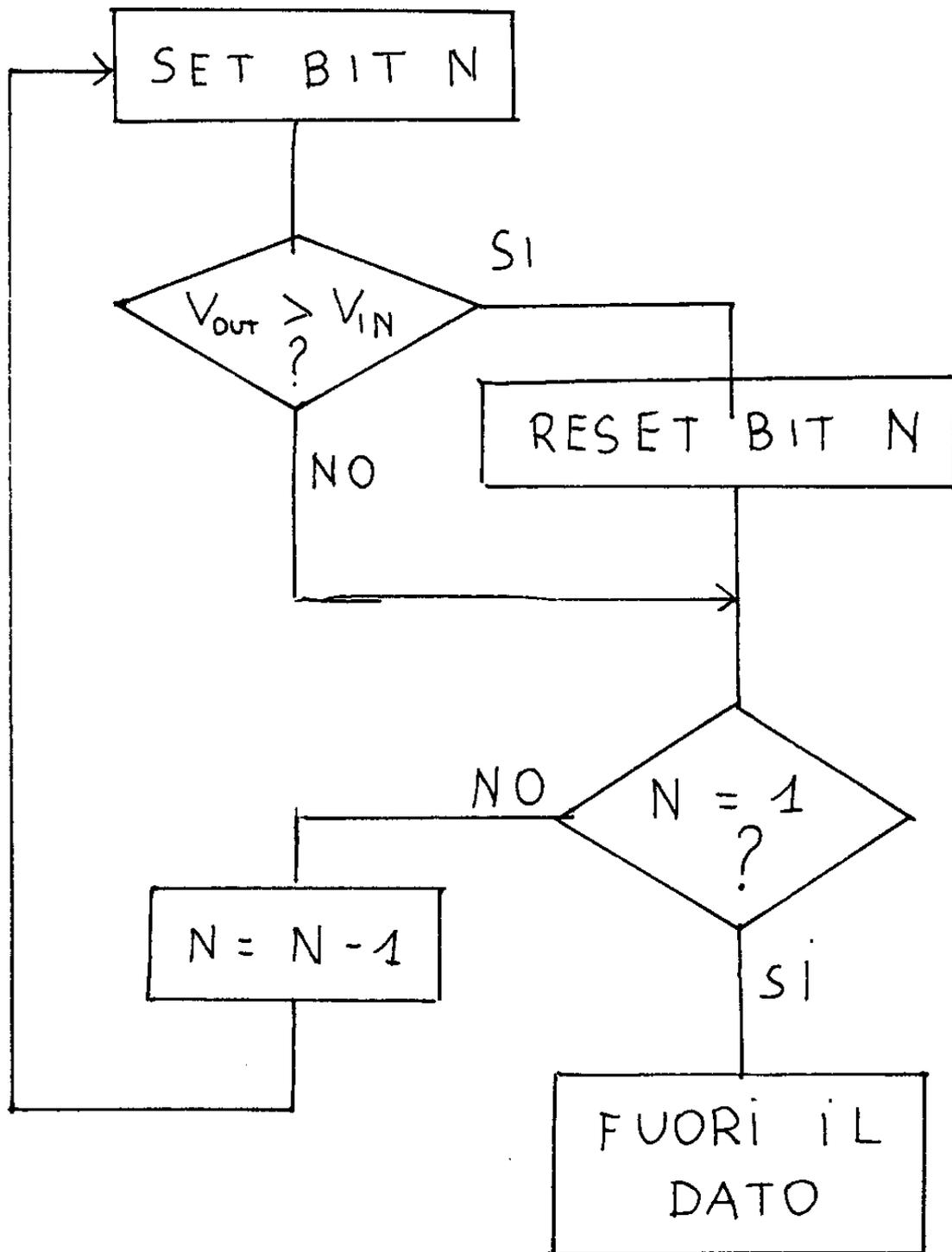
narica di V_{in} .



Questa logica è schematizzata anche a tavola IV.

Il tempo di conversione con questo sistema è proporzionale soltanto al numero N di bit usati e non (come nei precedenti)

TAVOLA IV
LOGICA DI UNA CONVERSIONE A/D
AD APPROSSIMAZIONI SUCCESSIVE



al valore massimo che il DAC può accettare, 2^N .

Abbiamo realizzato a programma queste due ultime tecniche.

Seguono i diagrammi di flusso e il Listing dei due programmi.

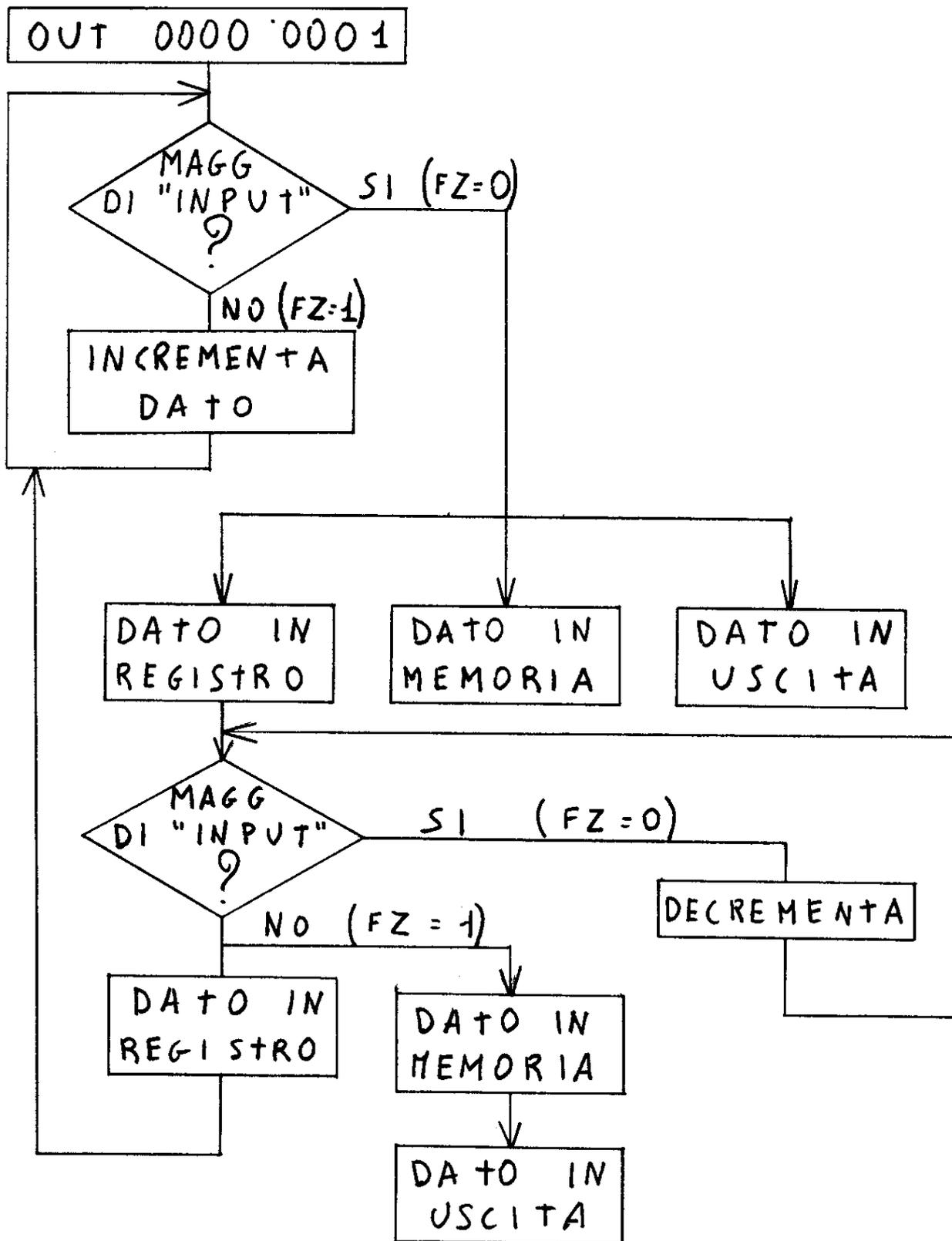
Successivamente abbiamo riportato alcune fotografie fatte all'oscilloscopio per confrontare i risultati pratici della conversione di un segnale sinusoidale a varie frequenze ma di ampiezza costante: 2 V. picco picco.

Per il programma ad aggancio è risultata una pendenza limite di 250 V./sec.

Per il programma ad approssimazioni successive abbiamo rilevato un tempo di campionamento di 350 μ sec.

DIAGRAMMA - DI - FLUSSO - DEL - PROGRAMMA

AD AGGANCIO

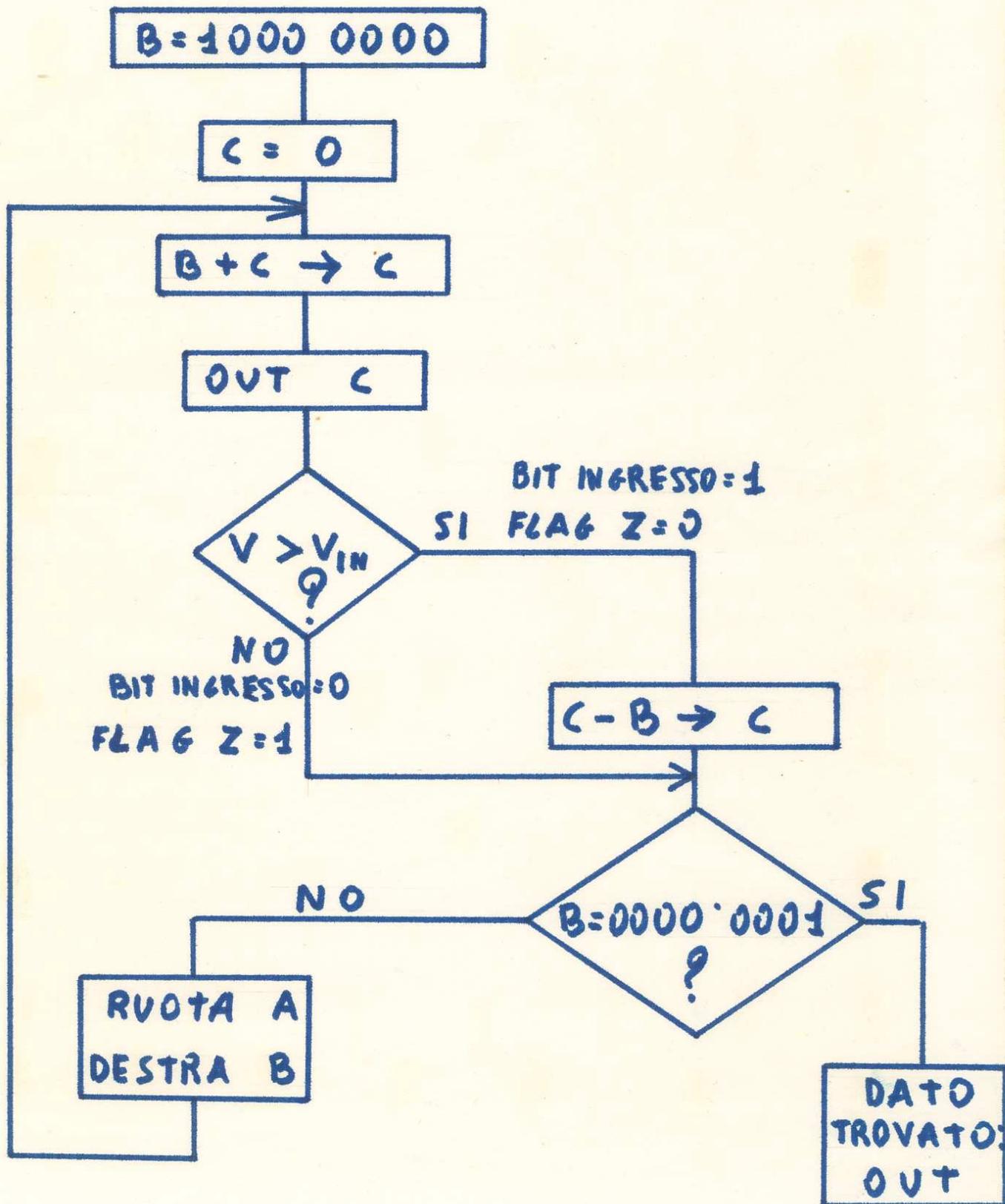


```
0700          ORG      0700H
0089          WORD2   EQU      89H
3C0B          CNTR    EQU      3C0BH
3C08          DAC1    EQU      3C08H
3C09          DAC2    EQU      3C09H
3C0A          COMP    EQU      3C0AH

0700 3E89          MVI      A,WORD2
0702 320B3C        STA      CNTR
0705 0601          MVI      B,01H
0707 78           MOV      A,B
0708 32083C        LOOP1:  STA      DAC1
070B 00           NOP
070C 00           NOP
070D 00           NOP
070E 3A0A3C        LDA      COMP
0711 E601          ANI      01H
0713 C21B07        JNZ      CAT1
0716 04           INR      B
0717 78           MOV      A,B
0718 C30807        JMP      LOOP1
071B 78           CAT1:   MOV      A,B
071C 32093C        STA      DAC2
071F 32083C        LOOP2:  STA      DAC1
0722 00           NOP
0723 00           NOP
0724 00           NOP
0725 3A0A3C        LDA      COMP
0728 E601          ANI      01H
072A CA3207        JZ      CAT2
072D 05           DCR      B
072E 78           MOV      A,B
072F C31F07        JMP      LOOP2
0732 78           CAT2:   MOV      A,B
0733 32093C        STA      DAC2
0736 C30807        JMP      LOOP1
0000          END
```

DIAGRAMMA - DI - FLUSSO - DEL - PROGRAMMA

APPROSSIMAZIONI - SUCCESSIVE

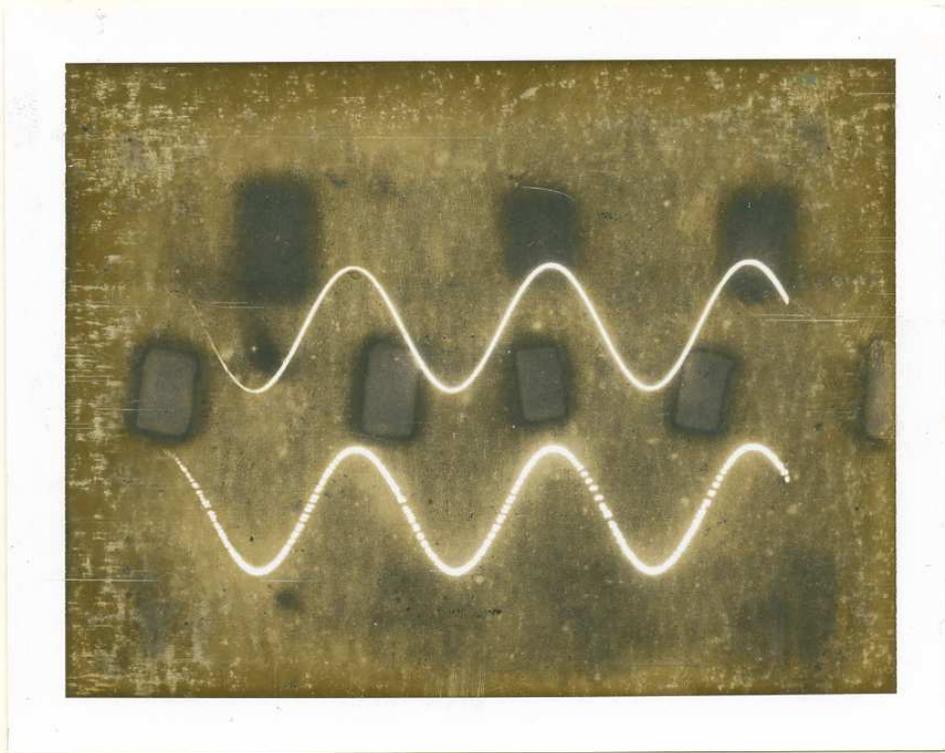


LISTA DEL PROGRAMMA AD APPROSSIMAZIONI SUCCESSIVE

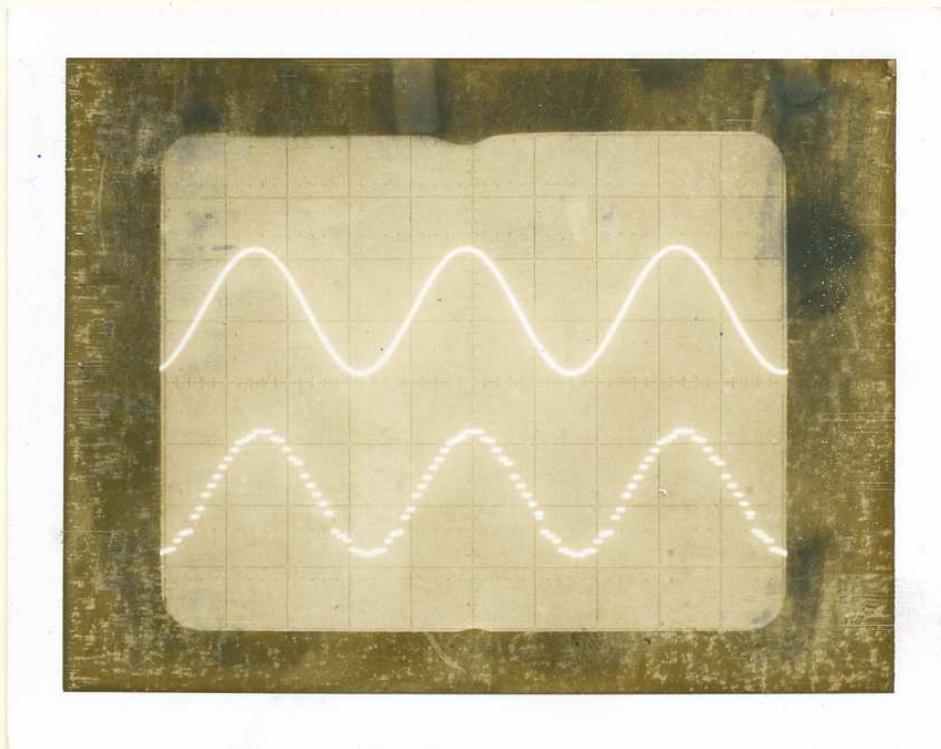
8080 MDS MACRO ASSEMBLER, V2.2

PAGE 1

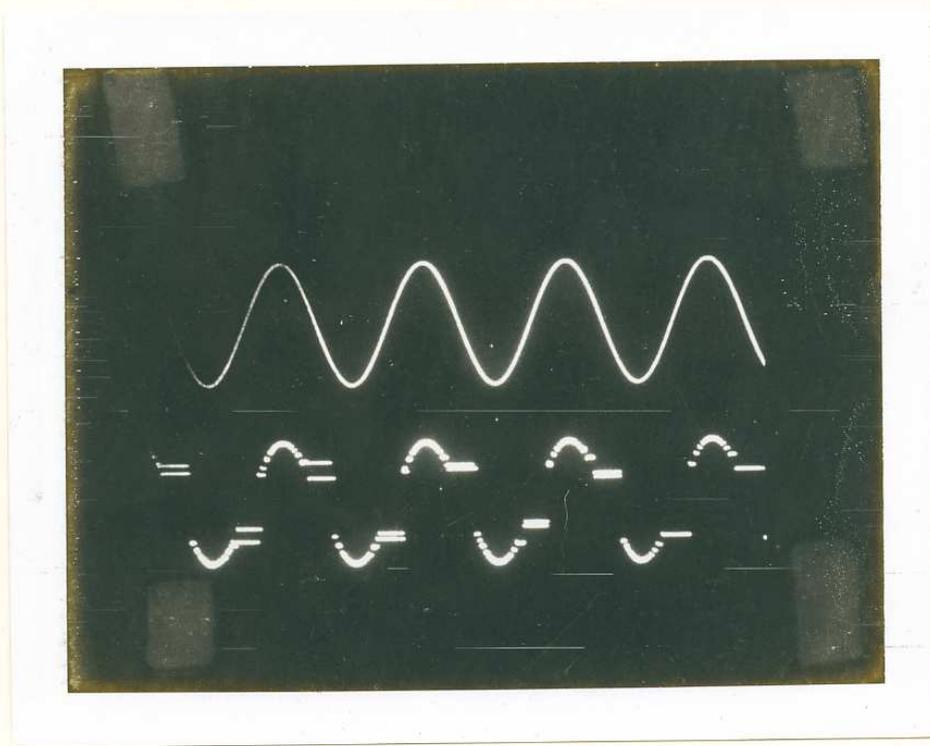
0780		ORG	0780H	
0089		WORD2	EQU	89H
3C08		DAC1	EQU	3C08H
3C09		DAC2	EQU	3C09H
3C0A		COMP	EQU	3C0AH
3C0B		CNTR	EQU	3C0BH
0780	0680	CAMP:	MVI	B,80H
0782	0E00		MVI	C,00
0784	78		MOV	A,B
0785	B1	LOOP:	ORA	C
0786	32083C		STA	DAC1
0789	00		NOP	
078A	00		NOP	
078B	00		NOP	
078C	3A0A3C		LDA	COMP
078F	E601		ANI	01
0791	CA9707		JZ	PROX
0794	78		MOV	A,B
0795	A9		XRA	C
0796	4F		MOV	C,A
0797	78	PROX:	MOV	A,B
0798	0F		RRC	
0799	47		MOV	B,A
079A	D28507		JNC	LOOP
079D	79		MOV	A,C
079E	32093C		STA	DAC2
07A1	C38007		JMP	CAMP
0000			END	



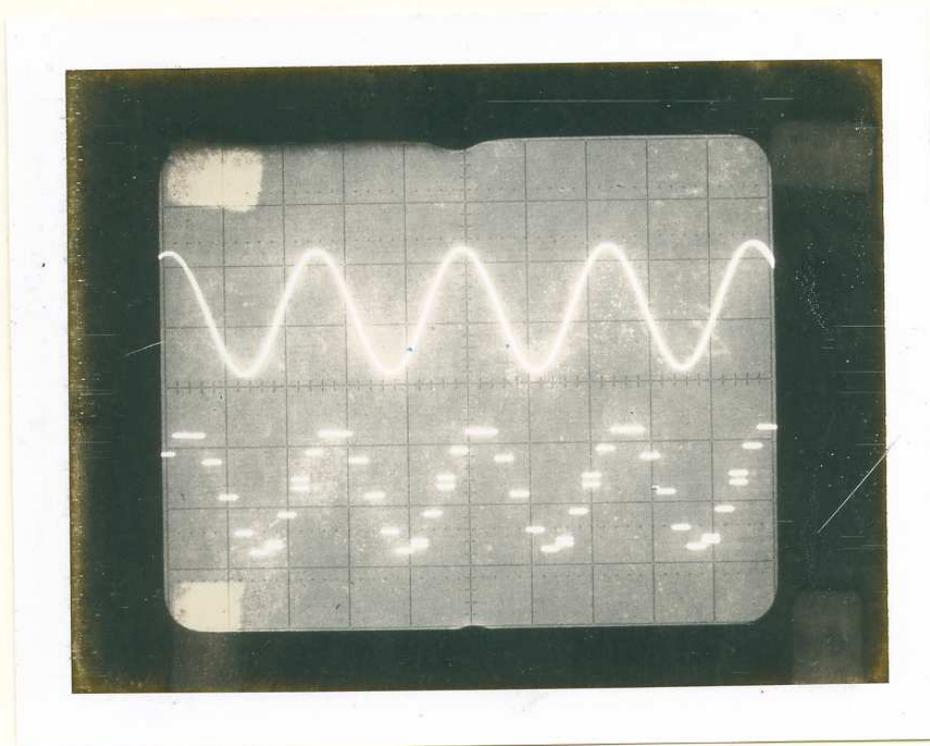
CONVERSIONE AD AGGANCIO 30 Hz 2 V. PICCO 10 millisecc./div.



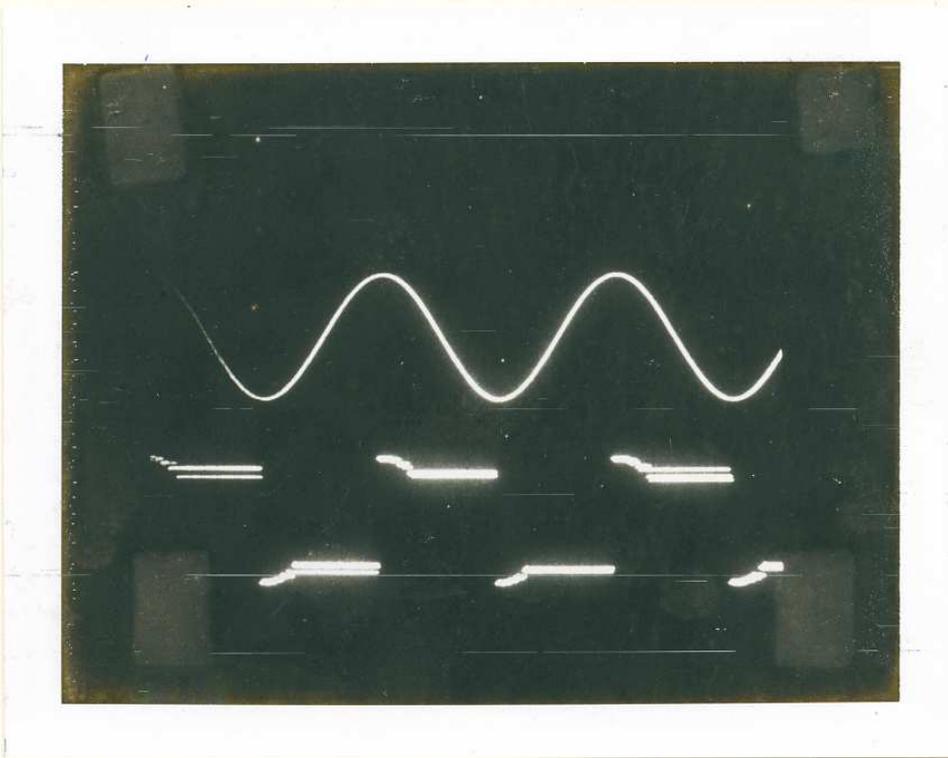
CONVERSIONE AD APPROSSIMAZIONI SUCCESSIVE 80 Hz 5 millisecc./div.



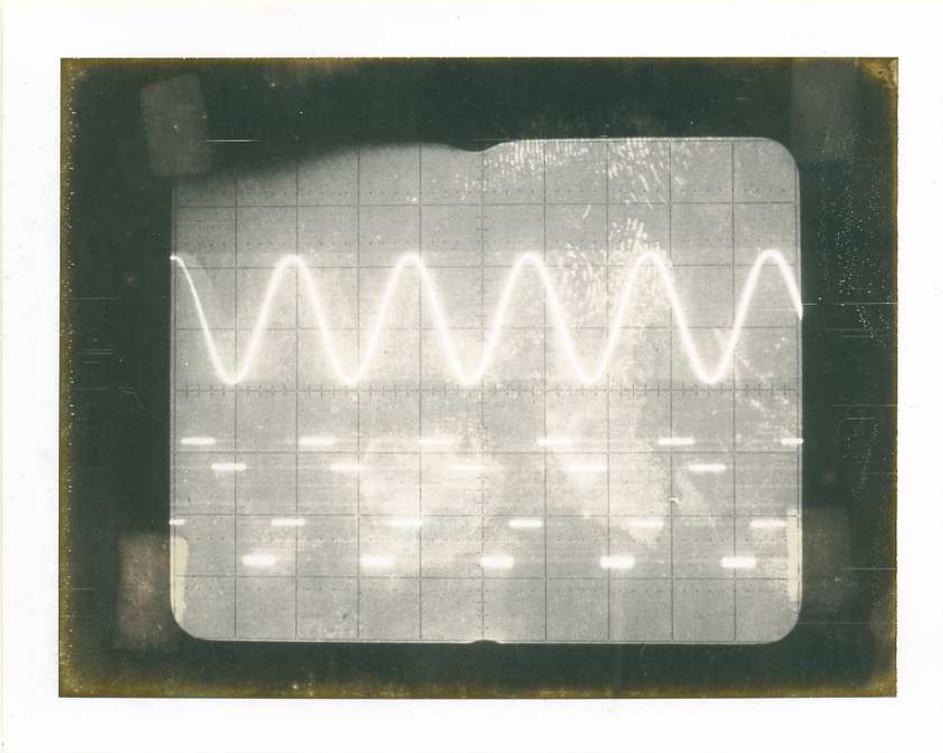
CONVERSIONE AD AGGANCIO 40 Hz 10 millisc./div.



CONVERSIONE AD APPROSSIMAZIONI SUCCESSIVE 200 Hz 2 millisc./div.



CONVERSIONE AD AGGANCIO 50 Hz 5 millisecc./div.



CONVERSIONE AD APPROSSIMAZIONI SUCCESSIVE 500 Hz 1 millisecc./div.

DESCRIZIONE DEL CRAMER KIT E DELLE MODIFICHE APPORTATE

Il microprocessore (μ P) usato è l'8080A della INTEL. Si tratta di un μ P che lavora con dati di 8 bits e indirizzi di 16 bits (tavola V).

Le istruzioni possono occupare 1, 2 o 3 BYTES in memoria.

I cicli della macchina sono pilotati da un CLOCK(8224) alla frequenza di 2 MHz. Un ciclo macchina dura cioè 0.5 sec.

Le istruzioni possono occupare da un minimo di 4 cicli macchina a un massimo di 18 cicli macchina. L'istruzione più breve dura quindi 2μ sec, la più lunga 9μ sec, con una media, fra tutte, di circa 4μ sec.

Si trova montato, a fili, in un kit della Cramer (CRAMER KIT) fornito di 1 K di RAM, 1 K di ROM, due interfacce programmabili (PPI 8255) e vari interruttori, LED, display, secondo lo schema riportato a Tavola VI.

Una configurazione del genere, in cui i dati di INPUT e OUTPUT (attraverso le interfacce 8255) vengono indirizzati come se fossero dati di una memoria, si dice MEMORY MAPPED I/O (Tavola VII).

TAVOLA V

SCHEMA DELLA CPU 8080 INTEL

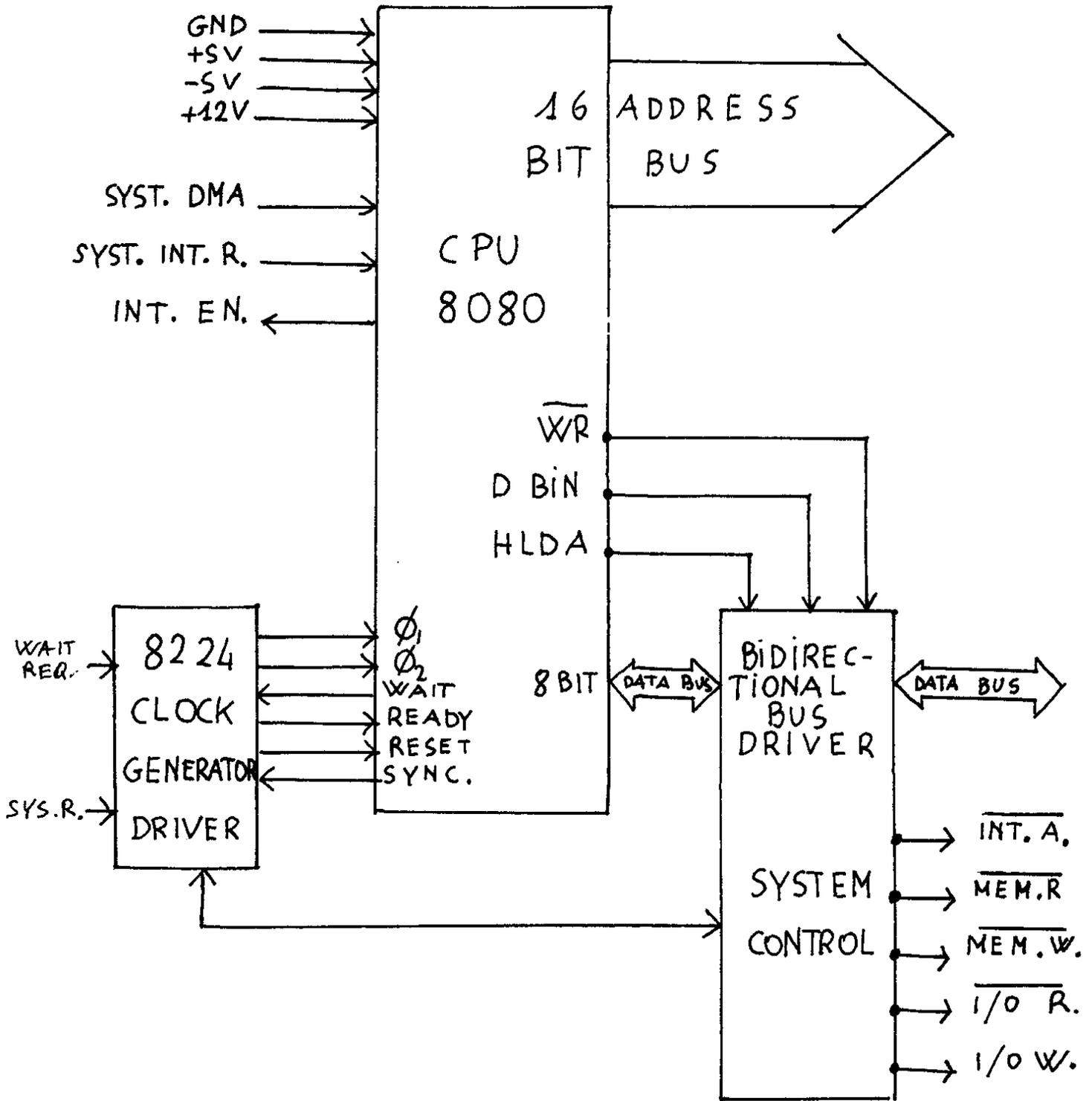


TAVOLA VI
 SCHEMA A BLOCCH DEL CRAMER KIT
 NELLA PRIMA VERSIONE

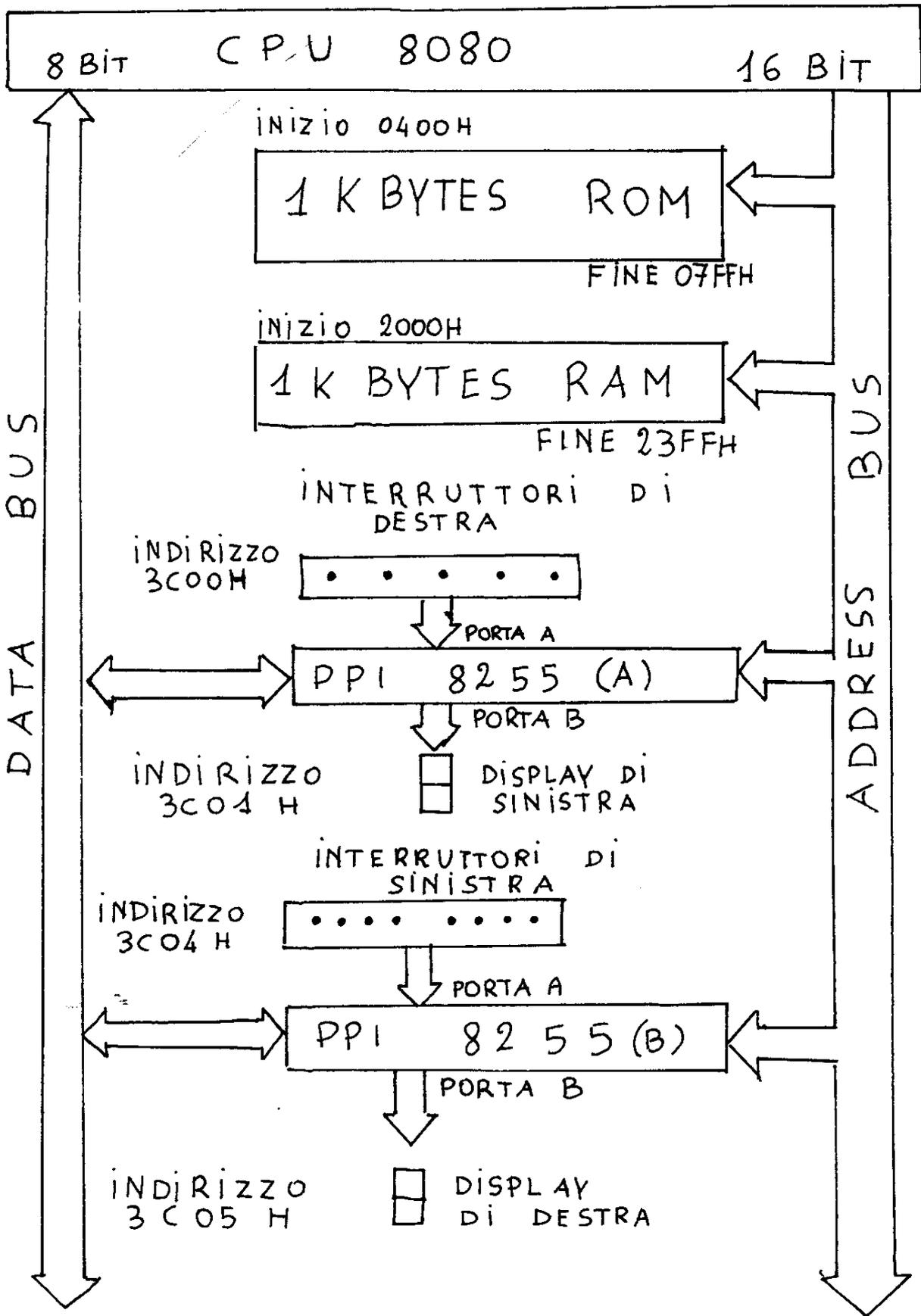
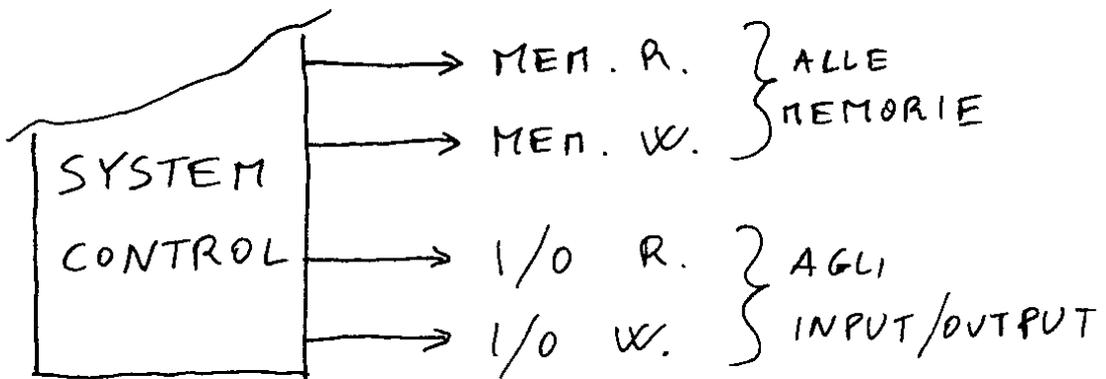
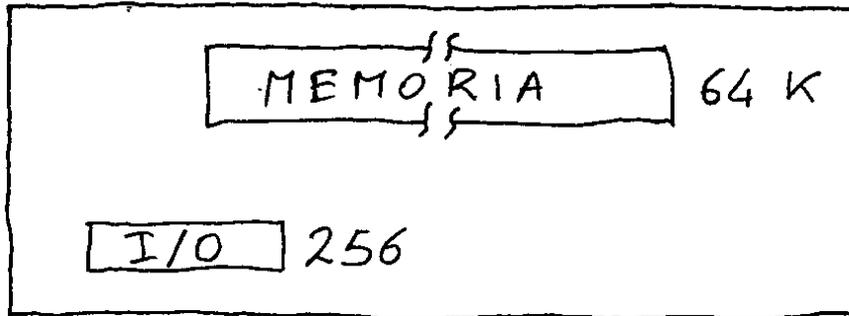
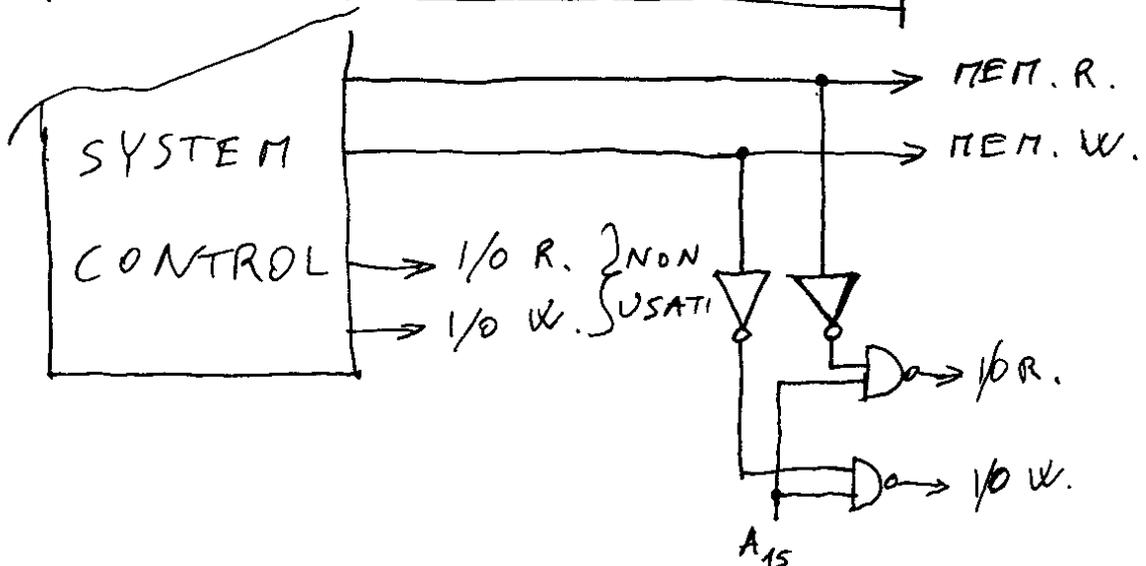
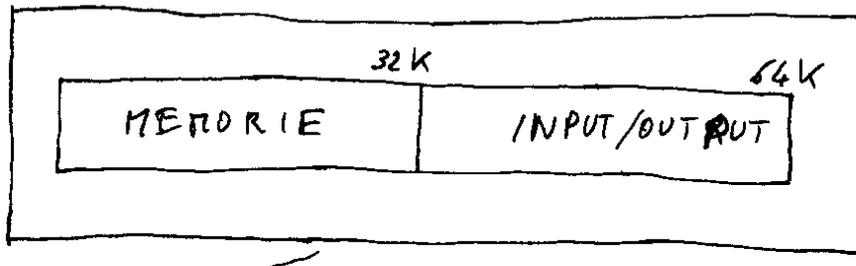


TAVOLA VII

INPUT/OUTPUT ISOLATO



INPUT/OUTPUT MEMORY MAPPED



Da questo tipo di montaggio derivano tre conseguenze importanti:

- le possibili porte di I/O aumentano da 256 (indirizzate con 8 bit) a 32000 (indirizzate assieme alle locazioni di memoria con 16 bits).
- In compenso le locazioni di memoria (64 K) si dimezzano.
- Infine nei programmi non sono più usabili l'istruzione IN (PORTA I/O → REGISTRO 'A') nè l'istruzione OUT (REGISTRO 'A' → PORTA I/O) ma rispettivamente LDA (MEMORIA → REGISTRO 'A') e STA (REGISTRO 'A' → MEMORIA) o istruzioni simili (MOV M,A; MOV A,M; LDAX; STAX).

Comunque quando il programma incontra l'istruzione IN (OUT) viene inviato uno 0 (1) sul filo I/O R. e un 1 (0) sul filo I/O W. Questi fili possono essere quindi usati per l'uscita di bit singoli in sequenza.

Quindi nel CRAMER KIT sono a disposizione:

- 1024 locazioni di memoria ROM, a partire dall'indirizzo 400H (H sta per esadecimale) fino a 7FFH.

Questa ROM contiene il programma MONITOR ma può essere sostit-

tuita con altre che contengono programmi preregistrati dall'utente.

- 1024 locazioni di memoria RAM dalla 2000H fino alla 23FFH

per i dati temporanei;

- 8 interruttori, a sinistra, per caricare manualmente dei

dati tramite la variazione dell'interruttore STROBE INPUT

appartenente al gruppo di

- 5 interruttori, a destra, usati per selezionare varie funzio-

ni sotto il controllo del programma MONITOR

- 2 display a sette segmenti per visualizzare i dati caricati

o qualunque altra cosa possa essere utile durante il funziona-

mento dei programmi;

- un pulsante di reset per riportare a zero il contenuto del

Program Counter, cioè per far ripartire da capo il programma.

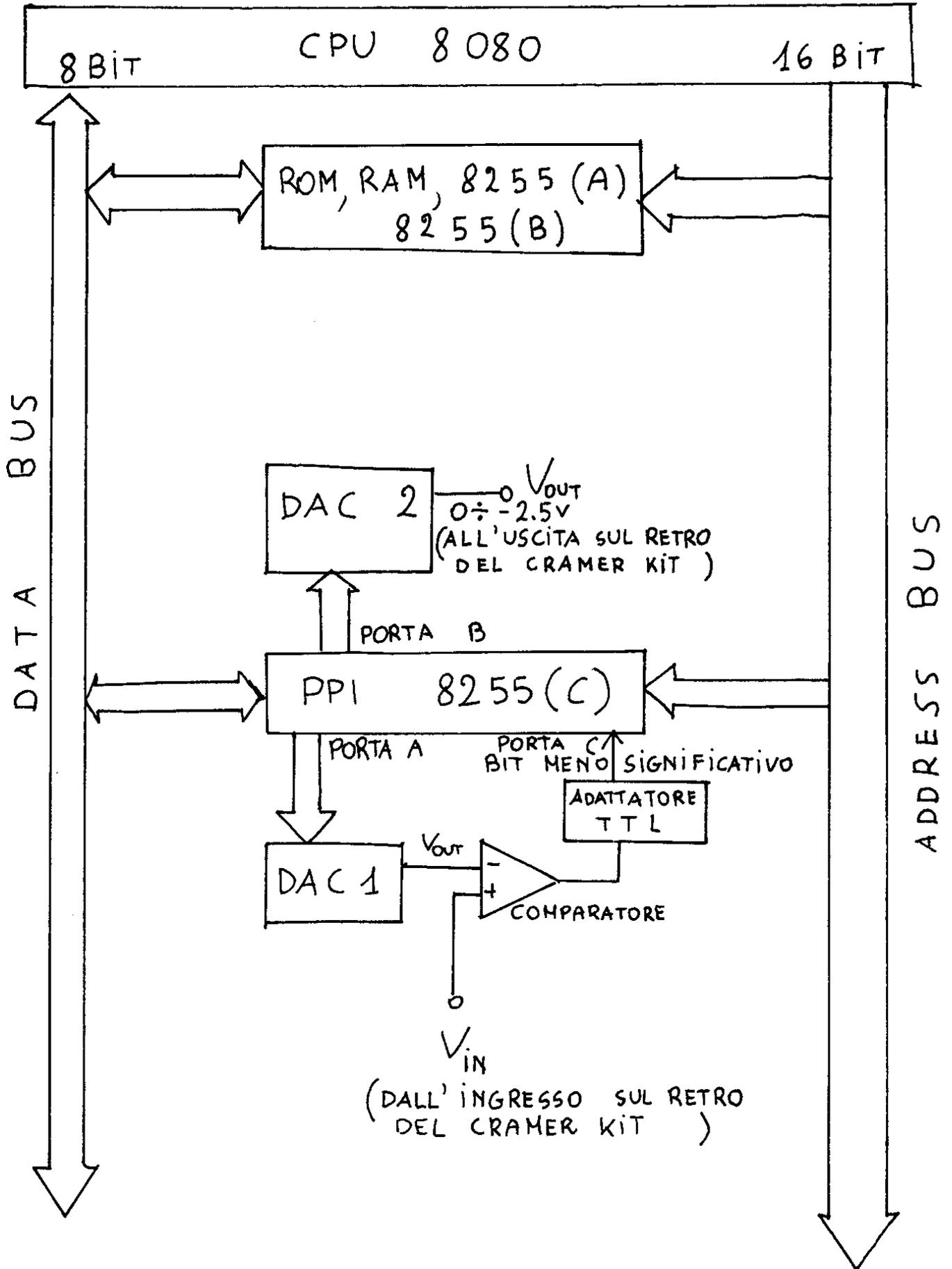
La disposizione di questi comandi è riportata sulla tavola VIII.

Questo elenco è incompleto ma è sufficiente per usare i due programmi aggiuntivi contenuti nel presente lavoro.

La prima modifica effettuata sul CRAMER KIT è stata l'aggiunta di un terzo 8255. E' stato montato assieme a due DAC e un Comparatore secondo lo schema riportato a tavola IX.

TAVOLA IX

SCHEMA DEL CRAMER KIT MODIFICATO



Questa nuova interfaccia ha i seguenti indirizzi:

PORTA A	3C08H	uscita sul primo DAC (DAC1)
PORTA B	3C09H	uscita sul secondo DAC (DAC2)
PORTA C	3COAH	ingresso dal comparatore (primo bit solamente)
CONTROLLO	3COBH	

I dati provenienti dalla CPU, se emessi dalla porta B, escono convertiti dal DAC2 in una tensione compresa tra 0 Volt (dato = 0) e -2.5 Volt (dato = FFH) che si può prelevare all'uscita OUT sul CRAMER KIT. I dati che escono dalla porta A vengono convertiti allo stesso modo da DAC1 ma la tensione in uscita va all'ingresso invertente di un comparatore. All'altro ingresso del comparatore giunge la tensione da misurare (che deve essere ugualmente compresa tra 0 e -2.5 V) proveniente dall'ingresso IN del CRAMER KIT.

Quando la tensione proveniente da DAC1 è minore, in valore assoluto, della tensione V_{in} il comparatore invia alla porta C il bit 0. Fornisce il bit 1 in caso contrario.

E' importante sottolineare che l'uscita del comparatore è solo +15V o -15V (la tensione di alimentazione) e quindi non

compatibile con i componenti TTL che seguono. Abbiamo perciò provveduto a diminuire la tensione con un partitore fino a +1.5V e -1.5V che va alla base di un transistor alimentato con +5V dal cui collettore preleviamo il segnale. Questo passa infine in una porta NOT (7404) che lo presenta alla porta C dell'8255(C) al bit 0 (LSB). Questa configurazione permette di realizzare a programma la conversione ANALOGICO-DIGITALE di cui si è detto sopra.

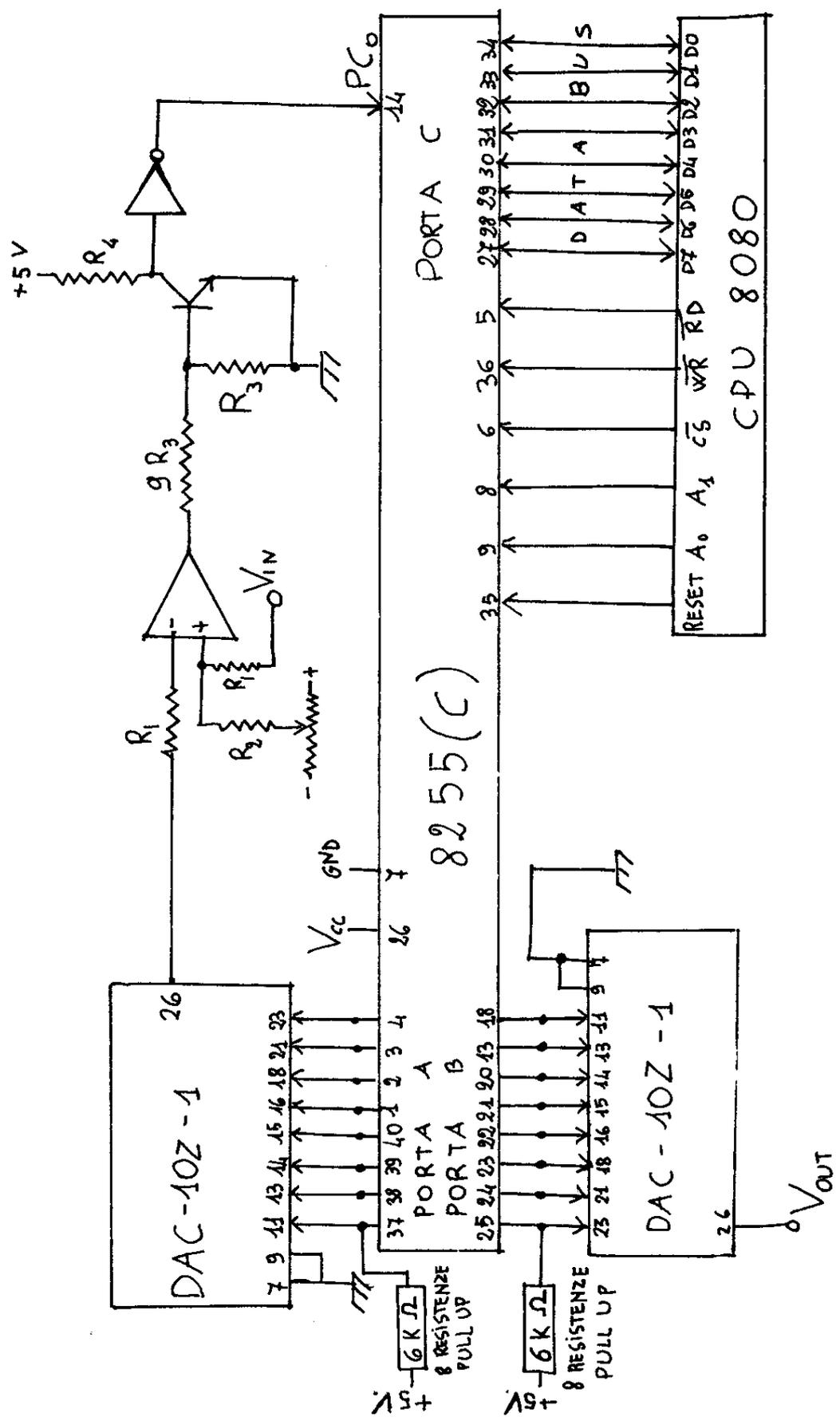
Un inconveniente si è verificato dopo un mese di corretto funzionamento. Alcuni bit dei due CONVERTITORI non riuscivano più ad avere valore 1.

Abbiamo scoperto che l'8255(C) si era rotto perchè non riusciva più a fornire la corrente necessaria nello stato 1 affinché il DAC lo accettasse effettivamente come valore 1. Prima di sostituirlo con il nuovo 8255 abbiamo messo 16 resistenze di PULL UP da $6K \Omega$ collegate da +5V. agli ingressi dei 2 DAC con il compito di fornire il milliAmpère necessario per il corretto funzionamento del DAC quando i bit sono a 1.

Lo schema elettrico è riportato a tavola X.

TAVOLA X

SCHEMA ELETTRICO DELLA BASETTA AGGIUNTA AL CRAMER KIT



RICOSTRUZIONE DI FUNZIONI PERIODICHE SECONDO FOURIER

E' noto che una funzione periodica, di periodo 2 nel nostro caso, sotto certe condizioni, soddisfa l'uguaglianza:

$$(4) \quad F(x) = A_0 + \sum_{n=1}^{\infty} A_n \operatorname{sen} nx + \sum_{n=1}^{\infty} B_n \operatorname{cos} nx$$

I termini al secondo membro costituiscono lo sviluppo in serie di Fourier di $F(x)$.

Esistono tavole che forniscono, per molte $F(x)$ i rispettivi A_0, A_n, B_n .

Il nostro scopo è di utilizzare il CRAMER KIT per visualizzare su un registratore scrivente su carta o su oscilloscopio un certo numero di termini e rendere evidente, a scopo didattico, il diverso modo di combinarsi delle armoniche e in più in dettaglio la velocità di convergenza di diversi sviluppi.

La prima difficoltà incontrata è stata la scarsa capacità di memoria del CRAMER KIT. 1024 BYTE di RAM per immagazzinare i parametri delle armoniche, calcolare i valori e sommarli non sono molti. In più negli altri 1024 BYTES di ROM dovevano trovare posto, oltre al programma, dei dati costanti per ogni

funzione e soprattutto i valori di una sinusoide che servisse da partenza per calcolare i valori delle armoniche successive.

Il problema è stato risolto accettando due compromessi. Per prima cosa ci siamo dovuti limitare a costruire un intero periodo della sinusoide fondamentale con soli 256 numeri. Risulta così formata da 256 gradini che, se non sono visibili su oscilloscopio agli ingrandimenti normali lo diventano aumentando la sensibilità della scala o semplicemente usando un registratore con pennino scrivente su carta.

La seconda necessità è stata quella di rinunciare alla risoluzione dei DAC a nostra disposizione (1 su 2^{10} essendo DAC a 10 bit) e lavorare solo con 8 bit, ponendo a 0 i due più significativi collegandoli a massa. Il μ P, pur essendo a 8 bit di lavoro, avrebbe potuto maneggiare ugualmente dati di 10 bit, complicando leggermente il programma, ma ciascun dato avrebbe occupato 2 BYTES di memoria diminuendo ancor di più il poco spazio disponibile.

In conclusione la ROM e la RAM sono state suddivise in quattro parti di 256 BYTES ciascuna e utilizzate in questo modo:

MAPPA DELLA ROM

0800H	0900H	0A00H	0B00H
PROGRAMMA FOURIER		DECODIFICHE PER DISPLAY SETTE SEGMENTI: ?? F1N1D1 P1 F2...PFF ON0...	VALORI DELLA SINUSOIDE FONDAMEN- TALE
08FFH	09FFH	0AFFH	0BFFH

MAPPA DELLA RAM

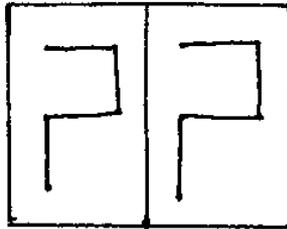
2000H (RAM1)	2100H (RAM2)	2200H (RAM3)	2300H (RAM4)
ZONA DEL CALCOLO DI CIASCUNA ARMONICA E DELLA FUNZIONE RISULTANTE	ZONA DELLA SOMMA PARTE DEI BYTES PIU' SIGNIFICATIVI	PARTE DEI BYTES MENO SIGNIFICATIVI	PRIME LOCALIZIONI PARAMETRI INTRODOTTI: LORO NUMERO, FREQUENZA, AMPIEZZA SFASAMENTO ULTIME LOCALIZIONI: STACK
20FFH	21FFH	22FFH	23FFH

Con questa tecnica un numero di 16 bits viene suddiviso in due parti: la parte sinistra (BYTE PIU' SIGNIFICATIVO) va in RAM2, la parte destra (BYTE MENO SIGNIFICATIVO) va in RAM3. Ad esempio il settimo numero ha questi indirizzi: 2106H BYTE più significativo; 2206H BYTE meno significativo.

STRUTTURA DEL PROGRAMMA FOURIER

Il programma per prima cosa trasferisce dalla ROM i valori della sinusoide fondamentale in RAM3. POI azzerata tutta RAM2. Ora è pronto per ricevere i dati dagli interruttori.

Visualizza sui display la prima richiesta:



Il dato da caricare a questo punto è il numero totale di parametri che definiscono le armoniche da sommare alla fondamentale. Questo numero è $4 \times$ Numero di armoniche. Cioè ogni armonica richiede per essere definita questi quattro numeri:

-FREQUENZA in multipli della frequenza fondamentale. Corrisponde a n nella (4)

-NUMERATORE e

-DENOMINATORE della frazione che misura l'ampiezza della armonica in unità di ampiezza della fondamentale. Se in (4)

$A_1 = 1$ allora gli A_n e i B_n sono proprio la frazione da

caricare

-SFASAMENTO. Numero compreso tra 0 e 255. Esprime la differenza di fase (in 256esimi di 2π). Serve per esprimere coseno, -coseno, -seno presenti nella (4) tutti in funzione di seno.

Infatti:

$$\cos x = \sin (x + \underline{64} \times 2\pi/256);$$

$$-\sin x = \sin (x + \underline{128} \times 2\pi/256);$$

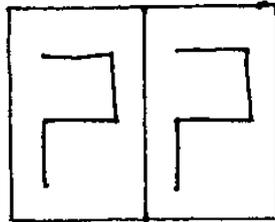
$$-\cos x = \sin (x + \underline{192} \times 2\pi/256).$$

Il numero sottolineato in ogni identità è lo sfasamento da impostare. I dati vanno caricati in questo ordine per ogni armonica. C'è da notare che A_0 viene tralasciato e così pure qualunque costante moltiplicativa comune a tutti i termini della serie perchè il programma comprime i valori finali tra 0 e 255.

Finito il caricamento dei dati il programma comincia a calcolare i valori di ciascuna armonica in RAM1 e a sommarli di volta in volta in RAM3 tenendo conto dei riporti nella locazione corrispondente di RAM2. Fatta la somma di tutte le armoniche cerca il massimo fra i numeri di RAM2.

Col numero trovato normalizza tutti i numeri, che in genera-

le sono a 16 bits, in numeri ad 8 bits. Il maggiore dei numeri a 16 bits diventa così un numero il più vicino possibile a 256. Si sfrutta in questo modo tutta l'informazione disponibile con 8 bits. Il risultato di queste 256 divisioni viene posto in RAM1 in ordine. Prima di far uscire i dati definitivi sui display appare ancora:



La richiesta corrispondente è: i dati devono uscire su oscilloscopio o su registratore scrivente su carta?

Se si usa il registratore bisogna caricare il numero 0; se si usa l'oscilloscopio va bene un qualunque numero diverso da 0. A questo punto i display sono tutti accesi. I dati cominciano a uscire in ordine e ciclicamente. Tra un dato e l'altro c'è una attesa di circa mezzo secondo se è in funzione il registratore su carta; non c'è nessun ritardo se si usa l'oscilloscopio. A questa velocità i 256 dati vengono emessi in circa 7 millisecondi; a una frequenza cioè di circa 140Hz. Nel caso di un registratore con velocità della carta di 5 cm/min

un periodo viene tracciato in poco più di 10 cm.

NOTE SU ALCUNE SUBROUTINES DEL PROGRAMMA

DISPL: legge il dato indirizzato da H,L; lo invia al display di sinistra; legge il successivo e lo invia al display di destra. Incrementa di 2 H,L.

STRB: legge la configurazione degli interruttori di destra, prende in considerazione il primo (STROBE INPUT).
Attende 25 millisecc prima di un'altra lettura. Confronta la nuova posizione dell'interruttore con la precedente. Quando c'è una variazione carica in 'A' il dato impostato sugli interruttori di sinistra.

DIV: un numero di 8 bits ('A') intero positivo viene diviso per un altro numero di 8 bits ('B') intero positivo.
Il risultato va in 'A'. Viene tralasciato il resto.

'B' contiene il denominatore della frazione $\frac{\text{ampiezza armonica}}{\text{ampiezza fondamentale}}$

- MULTI:** il risultato di DIV viene moltiplicato per 'B', che contiene il numeratore della precedente frazione. Essendo questa frazione minore di 1 il risultato non supera mai 255.
- SOMMA:** un numero di RAM1 viene sommato al corrispondente di RAM3. Il risultato va in RAM3. Se c'è riporto viene aumentato di 1 il corrispondente di RAM2.
- DDIVI:** un numero di 16 bits (H,L) formato da un BYTE di RAM2 ('H') e dal BYTE corrispondente di RAM3 ('L') viene diviso per il massimo + 1 fra i valori contenuti in RAM2: in questo modo il risultato ('D') sarà sicuramente inferiore a 256 (8 bits). Così per tutti i valori accoppiati di RAM2 e RAM3. Il risultato va in RAM1.
- DIVID:** questa routine che esegue la divisione usa la tecnica più lenta ma in questo programma un secondo o due di calcoli non sono quasi avvertiti.

ISTRUZIONI PER L'USO DEL PROGRAMMA FOURIER

-Connettere l'oscilloscopio o il registratore scrivente su carta alla presa OUT sul retro del CRAMER KIT. Escono tensioni comprese tra 0 e -2.5 V.

-Alimentare i due DAC con +15 V. e -15 V.

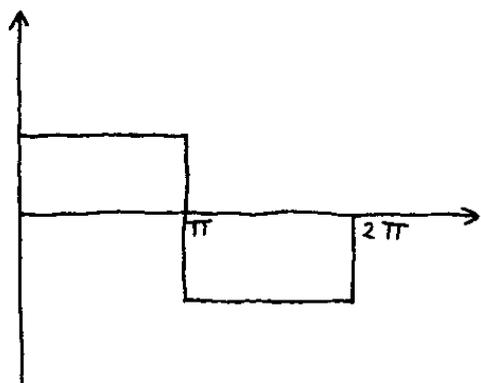
-All'accensione il programma ha il controllo e risponde sui display con '? ?' due punti interrogativi.

Il programma ora prevede una successione di operazioni per introdurre i parametri delle armoniche.

Queste operazioni sono descritte nella prossima pagina.

ESEMPI DI FUNZIONI, TABELLE DEI PARAMETRI E GRAFICI RISULTANTI

ONDA QUADRA



$$f(x) = \frac{4}{\pi} \sum_{n=1,3,5,\dots} \frac{1}{n} \sin nx$$

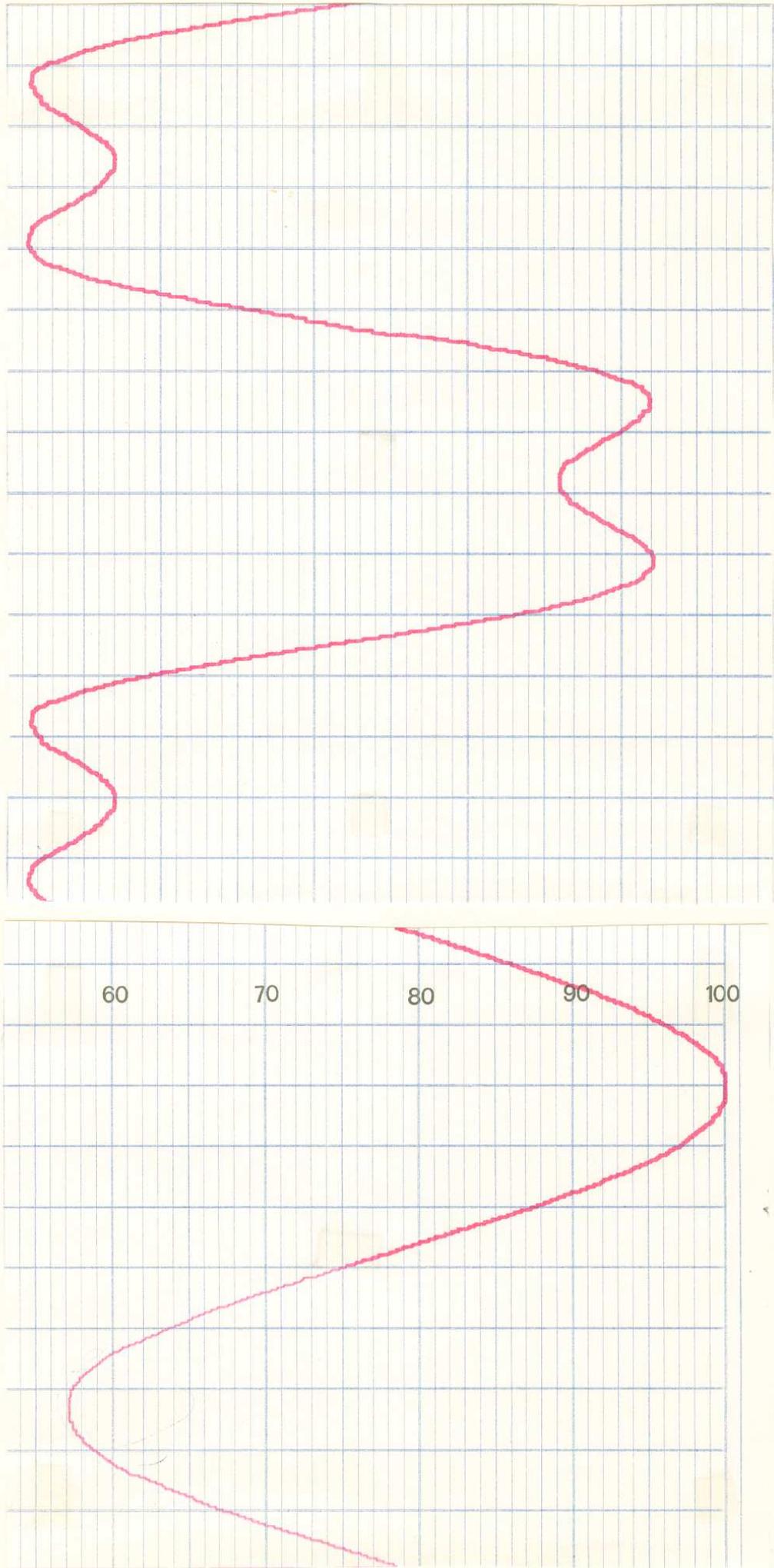
ARMONICA	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1
F	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35
N	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

N.B. Eventuali costanti moltiplicative o additive INDIPENDENTI

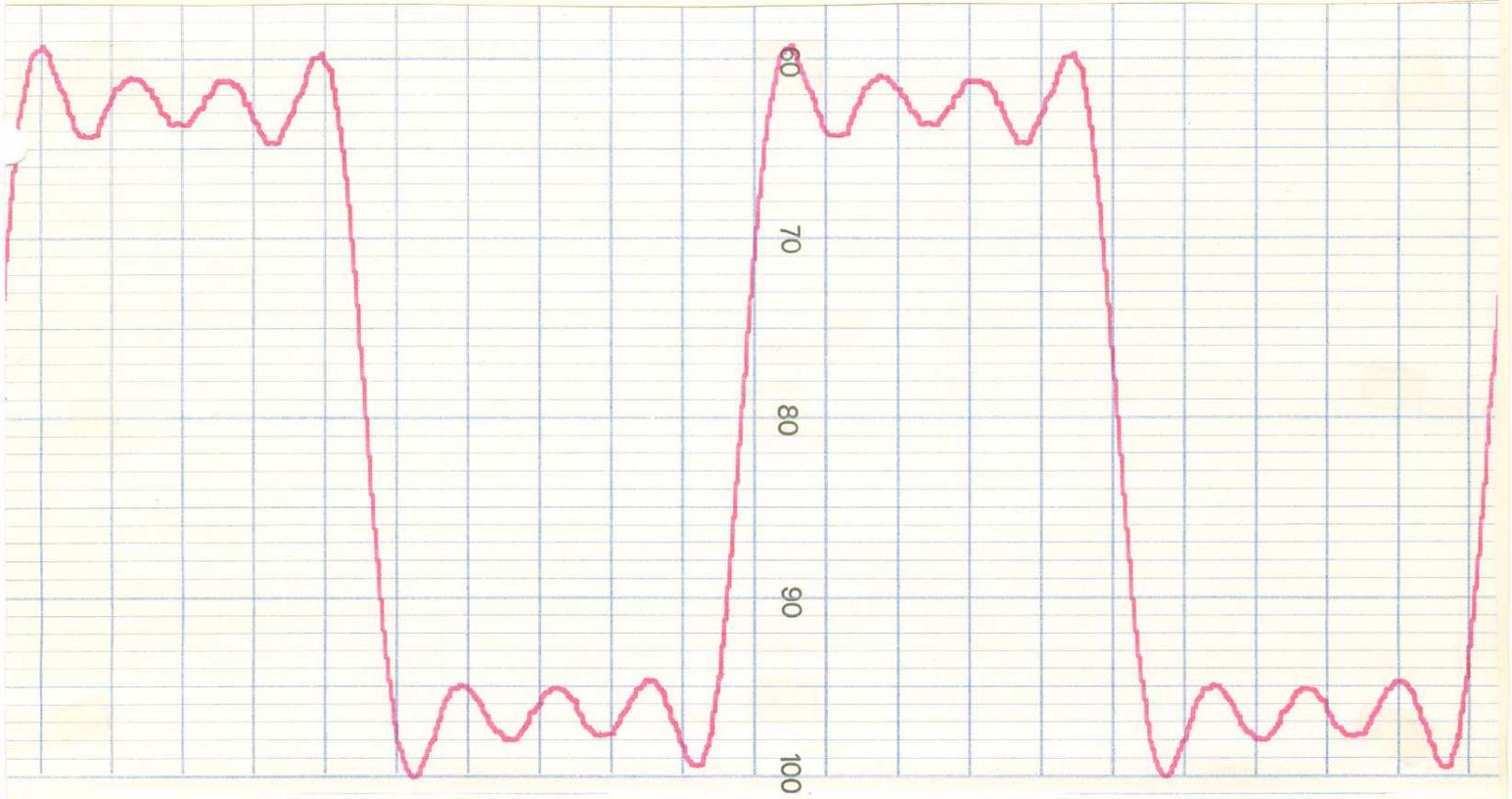
da n sono tralasciate perchè il programma comprime comunque

i dati finali tra 0 e 256.

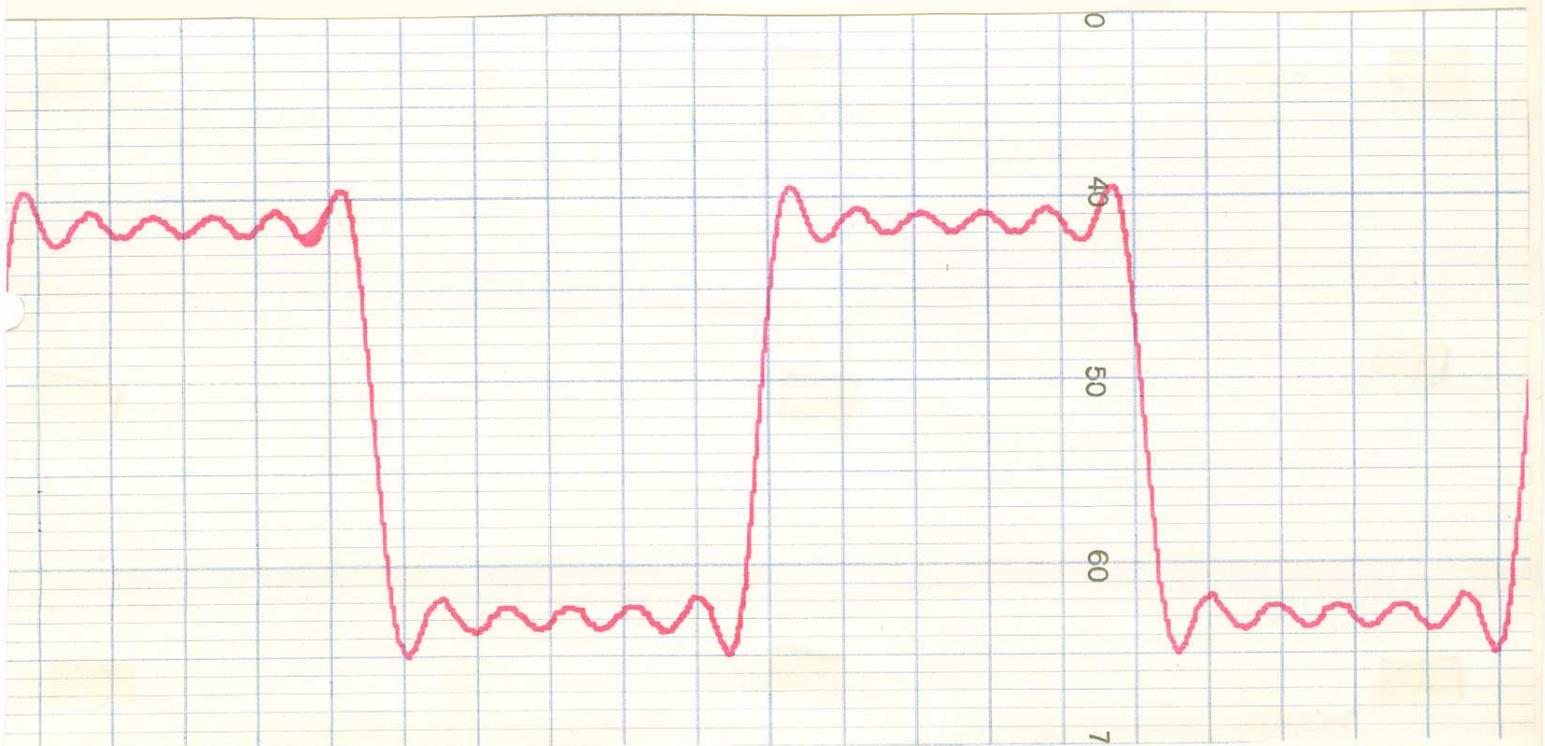
GRAFICI DELLA SINUSOIDE FONDAMENTALE E DELLA PRIMA APPROSSIMAZIONE DELL'ONDA QUADRA



ONDA QUADRA: FONDAMENTALE + 3 ARMONICHE



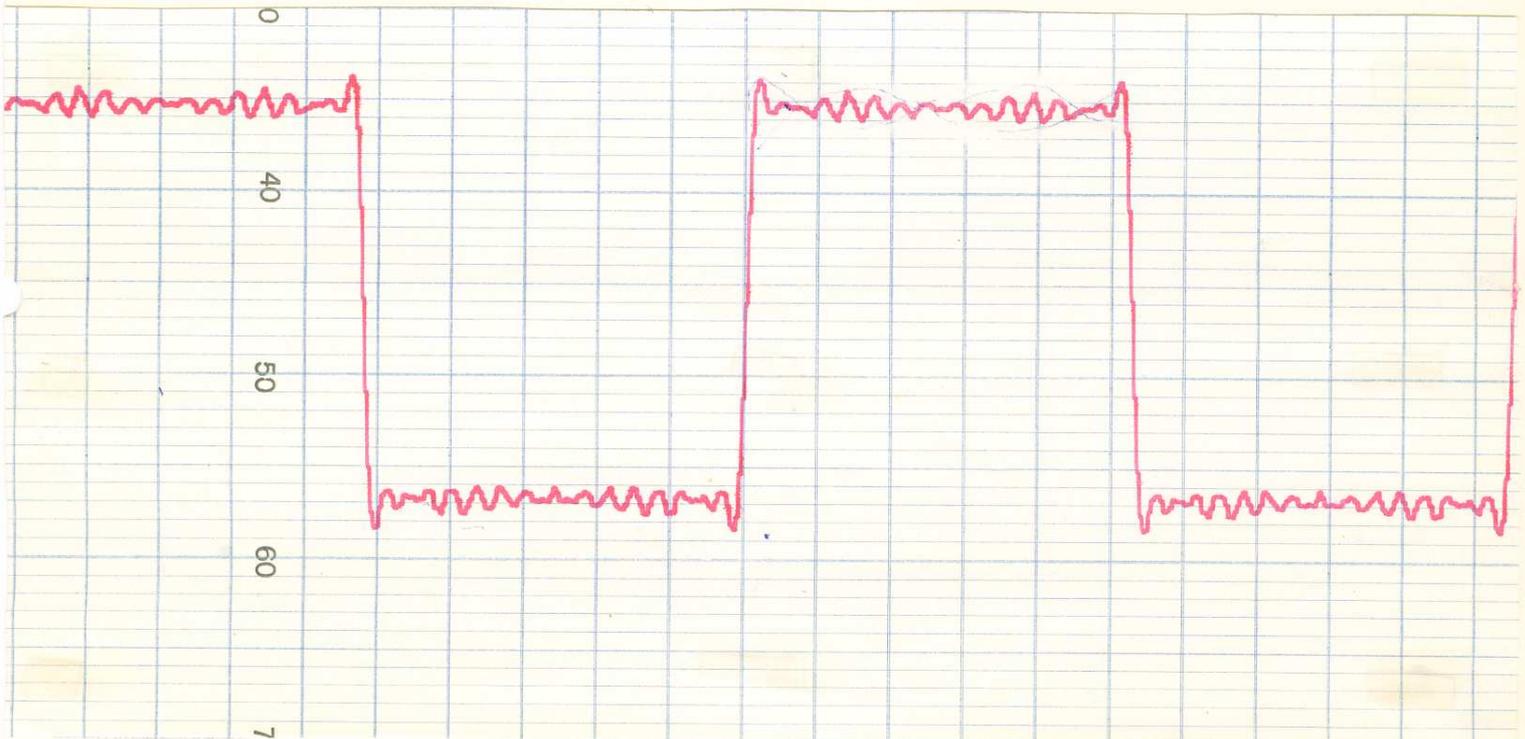
ONDA QUADRA: FONDAMENTALE + 5 ARMONICHE

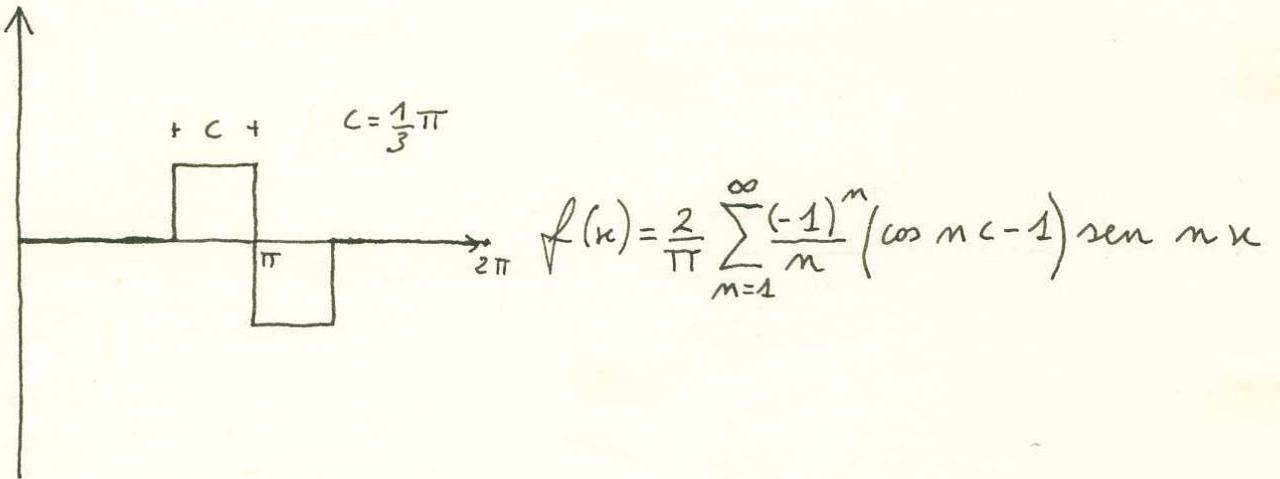


ONDA QUADRA: FONDAMENTALE + 8 ARMONICHE



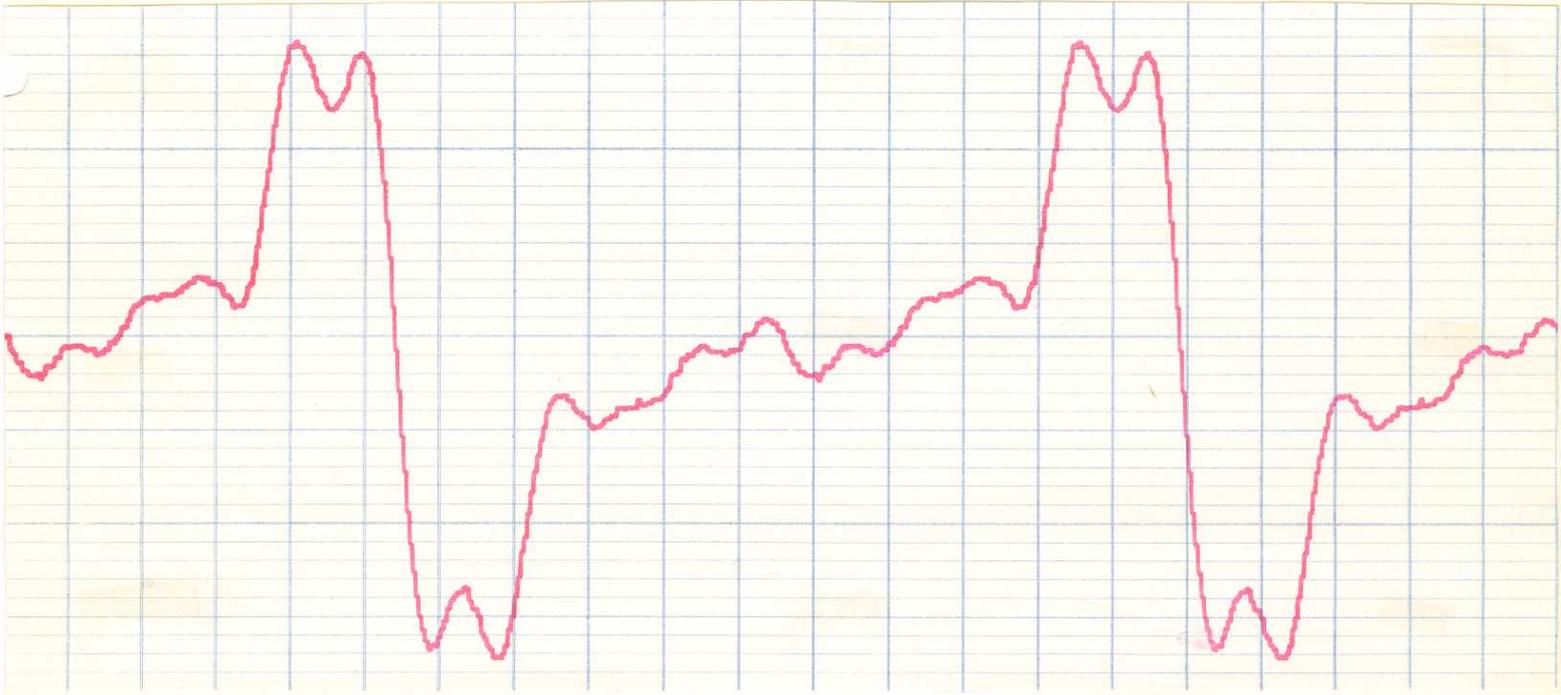
ONDA QUADRA: FONDAMENTALE + 15 ARMONICHE. NOTARE I PICCHI DI GIBBS E IL RIPPLE DEL TETTO



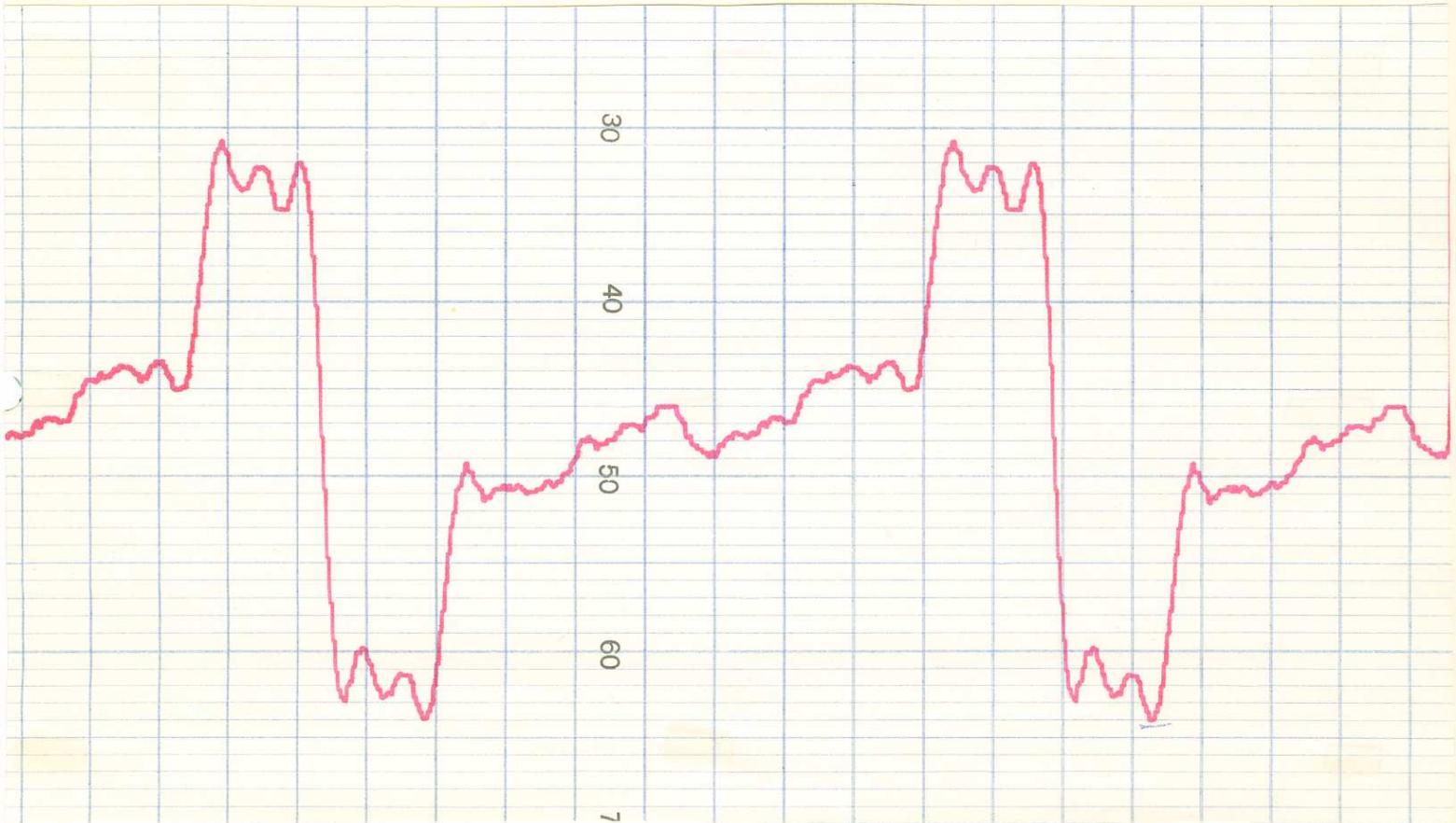


ARMONICA	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1
F	2	3	4	5	7	8	9	10	11	13	14	15	16	17	19	20	21
N	3	2	3	1	1	3	2	3	1	1	3	2	3	1	1	3	2
D	4	3	8	10	14	16	9	20	22	26	28	15	32	34	38	40	21
P	128	0	128	0	0	128	0	128	0	0	128	0	128	0	0	128	0

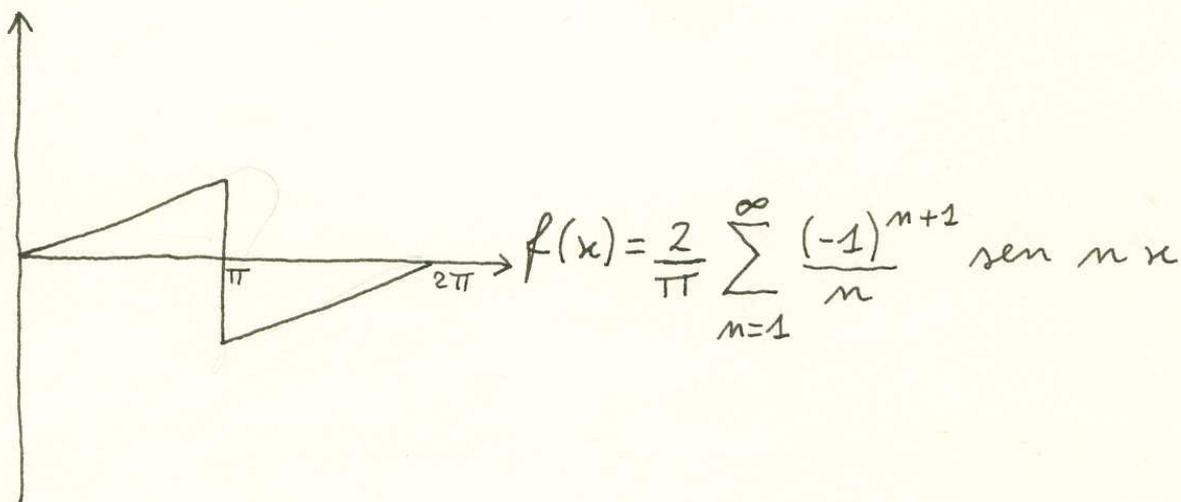
FONDAMENTALE + 8 ARMONICHE. LENTISSIMA LA CONVERGENZA DI
QUESTO SVILUPPO IN SERIE



FONDAMENTALE + 15 ARMONICHE



RAMPA DI SALITA

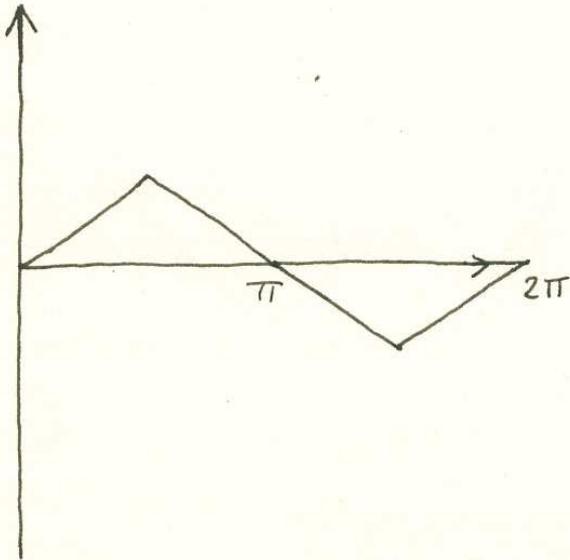


ARMONICA	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1
F	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
N	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
D	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
P	128	0	128	0	128	0	128	0	120	0	128	0	128	0	128	0	128

RAMPA DI SALITA: FONDAMENTALE + 19 ARMONICHE



ONDA TRIANGOLARE



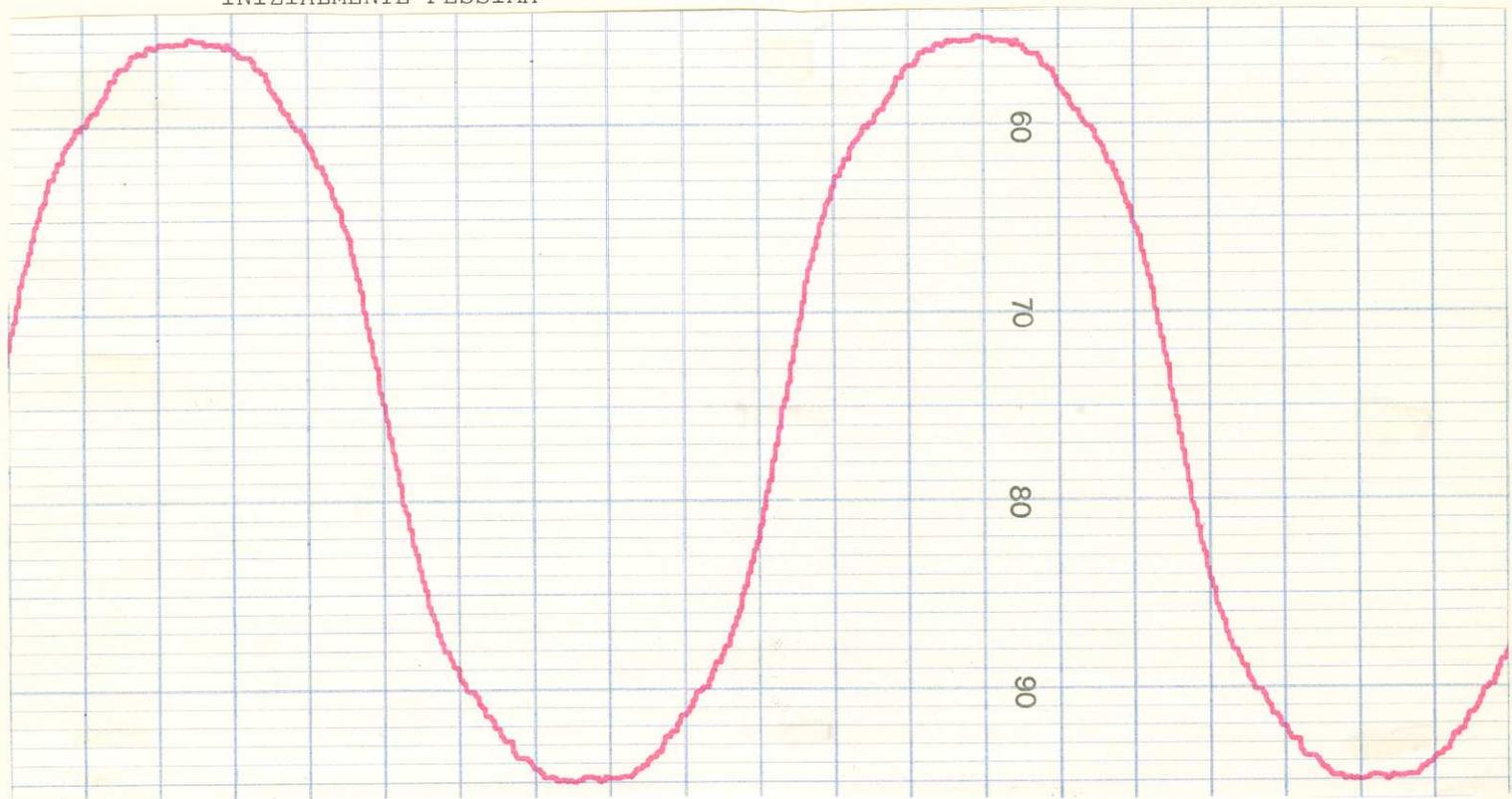
$$f(x) = \frac{8}{\pi^2} \sum_{m=1,3,5,\dots} \frac{(-1)^{\frac{m-1}{2}}}{m^2} \text{sen } mx$$

ARMONICA	1	2	3	4	5	6	7
F	3	5	7	9	11	13	15
N	1	1	1	1	1	1	1
D	9	25	49	81	121	169	225
P	128	0	128	0	128	0	128

NOTA: Non avrebbe senso impostare le successive armoniche

(nè si potrebbe) perchè D è maggiore di 255. Cioè l'ampiezza è già inferiore a 1/255 della fondamentale e quindi non più registrabile in Bytes di 8 bits.

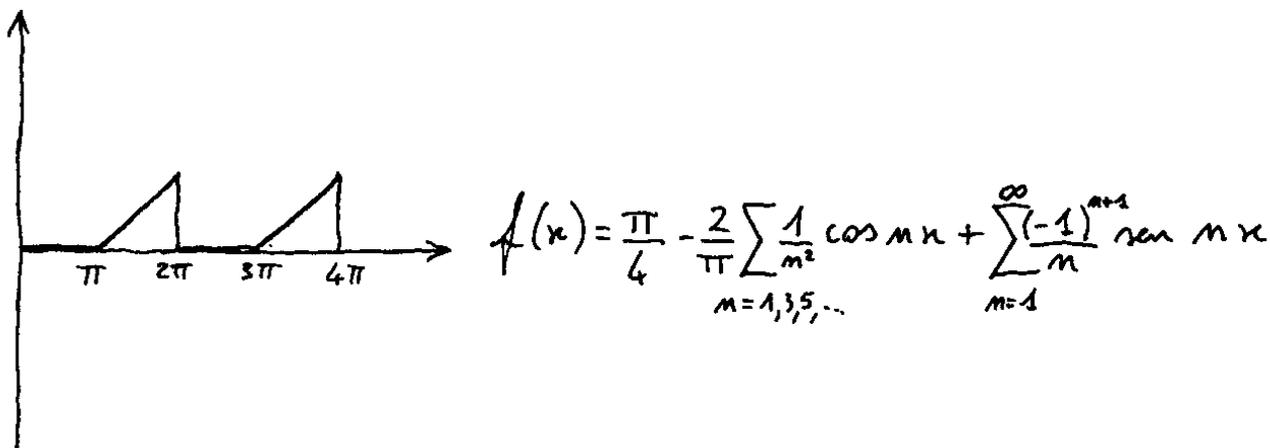
ONDA TRIANGOLARE: FONDAMENTALE + 3 ARMONICHE. APPROSSIMAZIONE
INIZIALMENTE PESSIMA



FONDAMENTALE + 7 ARMONICHE. LE 4 ARMONICHE AGGIUNTE MIGLIORANO
VELOCEMENTE L' APPROSSIMAZIONE



RAMPA DI SALITA CON PIANEROTTOLO

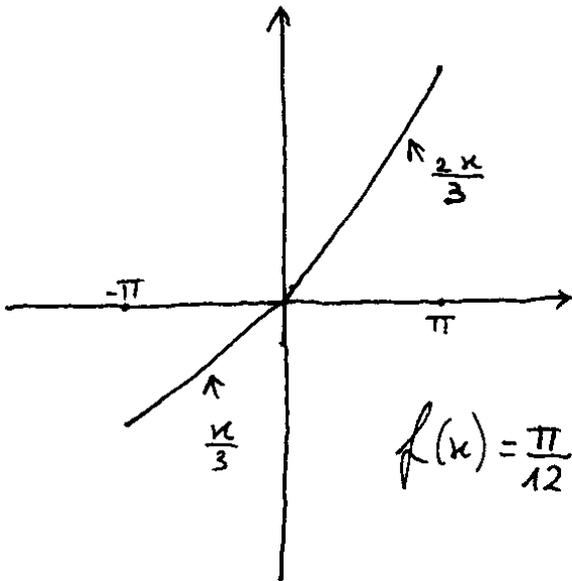


ARMONICA	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0
F	1	2	3	3	4	5	5	6	7	7	8	9	9	10	11	11
N	2*	1	1	1*	1	1	1*	1	1	1*	1	1	1*	1	1	1*
D	3*	2	3	14*	4	5	39*	6	7	77*	8	9	127*	10	11	190*
P	192	128	0	192	128	0	192	128	0	192	128	0	192	128	0	192

ARMONICA	1	2	3	4	5	6	7	8	9	A	B	C	D
F	12	13	14	15	16	17	18	19	20	21	22	23	24
N	1	1	1	1	1	1	1	1	1	1	1	1	1
D	12	13	14	15	16	17	18	19	20	21	22	23	24
P	128	0	128	0	128	0	128	0	128	0	128	0	128

I numeri segnati con * sono stati arrotondati perchè provenienti da frazioni che contengono π.

RAMPA DI SALITA CON DUE PENDENZE



$$f(x) = \frac{\pi}{12} - \frac{2}{3\pi} \sum_{m=1,3,5,\dots} \frac{1}{m^2} \cos mx + \sum_{m=1}^{\infty} \frac{(-1)^{m+1}}{m} \sin mx$$

ARMONICA	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1
F	1	2	3	3	4	5	5	6	7	7	8	9	10	11	12	13	14
N	1*	1	1	1*	1	1	1*	1	1	1*	1	1	1	1	1	1	1
D	5*	2	3	42*	4	5	118*	6	7	231*	8	9	10	11	12	13	14
P	192	128	0	192	128	0	192	128	0	192	128	0	128	0	128	0	128

I numeri segnati con * sono arrotondati da frazioni che contengono π.

MEDIA DI IMPULSI QUASI PERIODICI DI TIPO BATTITO CARDIACO

I tracciati dell'elettrocardiogramma, del fonocardiogramma, e dell'ecocardiogramma sono argomento di studio da parte di coloro che si occupano di misure in cardiologia per rilevarne caratteristiche diagnostiche. Una parte di queste misure consiste nel determinare il periodo degli impulsi e di calcolare il periodo medio su un certo numero di essi, dato che la frequenza del battito non è mai esattamente costante nemmeno in persone sane. Per di più la forma stessa degli impulsi e in particolare il tempo di salita (sistole) cambia leggermente e di continuo. Altre misure sono quindi volte a determinare la durata della sistole e delle sue variazioni. Tutte queste misure e queste medie sono eseguite manualmente sui grafici che le varie apparecchiature di rilevazione e registrazione (normalmente dei poligrafi galvanometrici) forniscono.

Abbiamo preso lo spunto per la presente applicazione dei μ^P da queste necessità, cercando non tanto di risolvere in modo completo tutto il problema (questo avrebbe richiesto

la disponibilità di apparecchiature specifiche della cardiologia) quanto di dare una dimostrazione della capacità dei μP di risolvere problemi del genere, che d'altra parte rappresentano il campo tipico di utilizzo degli stessi.

Cioè in generale vengono usati per l'elaborazione in tempo reale di alcuni dati o segnali prelevati da appositi sensori.

Il lavoro è quindi iniziato con la costruzione di un generatore di segnali analoghi al battito cardiaco.

Per analogo non si intende della stessa forma ma con gli stessi parametri critici del battito cardiaco. Quindi rimangono circa uguali la frequenza, la sua variabilità, la larghezza dell'impulso, la ripidità del fronte di salita e di discesa.

Ne segue che il problema da noi affrontato e risolto abbastanza semplicemente ha il vantaggio di poter essere allargato anche ad altri campi di indagine scientifica in cui si abbia a che fare con segnali di forma e frequenza imprevedibili.

Altro fatto importante: il metodo da noi usato è proprio quello che di solito si usa in campo industriale

nell'impiego dei μP per automatizzare un processo fisico.

Un esempio chiarirà meglio questo metodo di lavoro.

Supponiamo che una industria di ceramiche voglia controllare il processo di cottura e i vari stadi attraverso cui passano le piastrelle, utilizzando un μP proprio per il motivo che quando dovessero in seguito cambiare alcune caratteristiche delle piastrelle (tipo di terre, spessore, tipo di disegno, tipo di vernici) risulti più semplice cambiare il programma nella ROM del μP che rifare l'intero progetto hardware in un sistema tradizionale a logica cablata.

A questo punto il progettista deve avere ben definiti tutti i parametri di ingresso e le specifiche richieste in uscita; sulla base di questi dati egli costruisce un hardware-emulator che riproduce in laboratorio le caratteristiche dell'impianto che il μP dovrà controllare.

Ora, con questo hardware-emulator a disposizione, è possibile sviluppare il sistema in modo autonomo in laboratorio fino a quando il μP svolge esattamente le

funzioni richieste. E' probabile comunque che la versione così approntata non sia quella definitiva poichè in fase di installazione possono verificarsi delle condizioni non previste che richiederanno ulteriori messa a punto. Ma il grosso del lavoro è già stato fatto più facilmente in laboratorio.

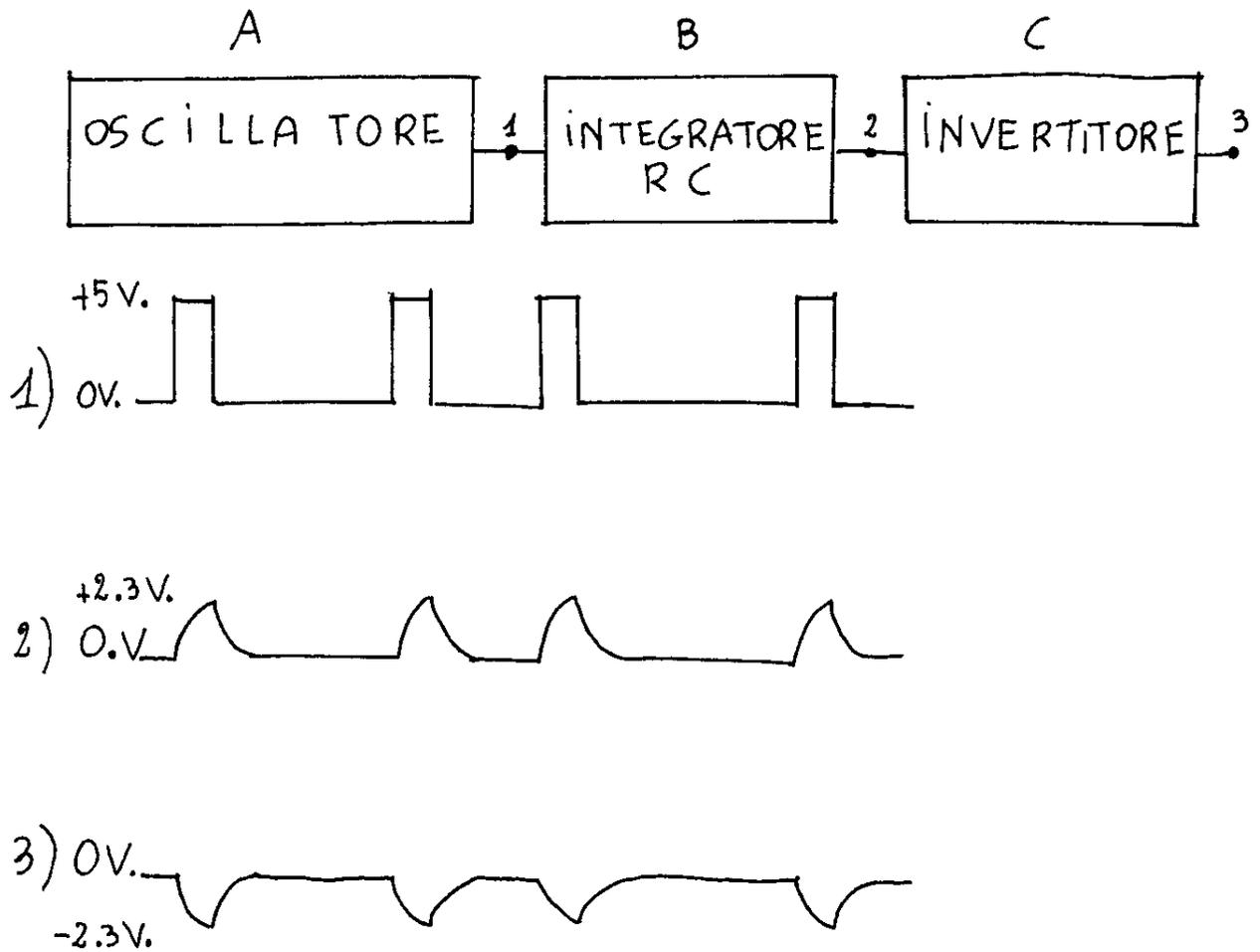
Ritornando al nostro particolare lavoro, riassumiamo i parametri che abbiamo preso in considerazione per costruire il simulatore di battito cardiaco:

FREQUENZA	1 Hz con possibilità di variazione manuale da 0.5 Hz a -2 Hz
LARGHEZZA DELL'IMPULSO	1/3 circa dell'intero periodo

L'ampiezza in tensione non è un parametro critico perchè con opportuni amplificatori può essere facilmente portata nell'intervallo di valori validi per l'input analogico del CRAMER KIT, cioè fra 0 e -2.5 V.

Questo dispositivo è particolarmente semplice: lo schema di principio è riportato a tavola XI e lo schema elettrico alla tavola XII

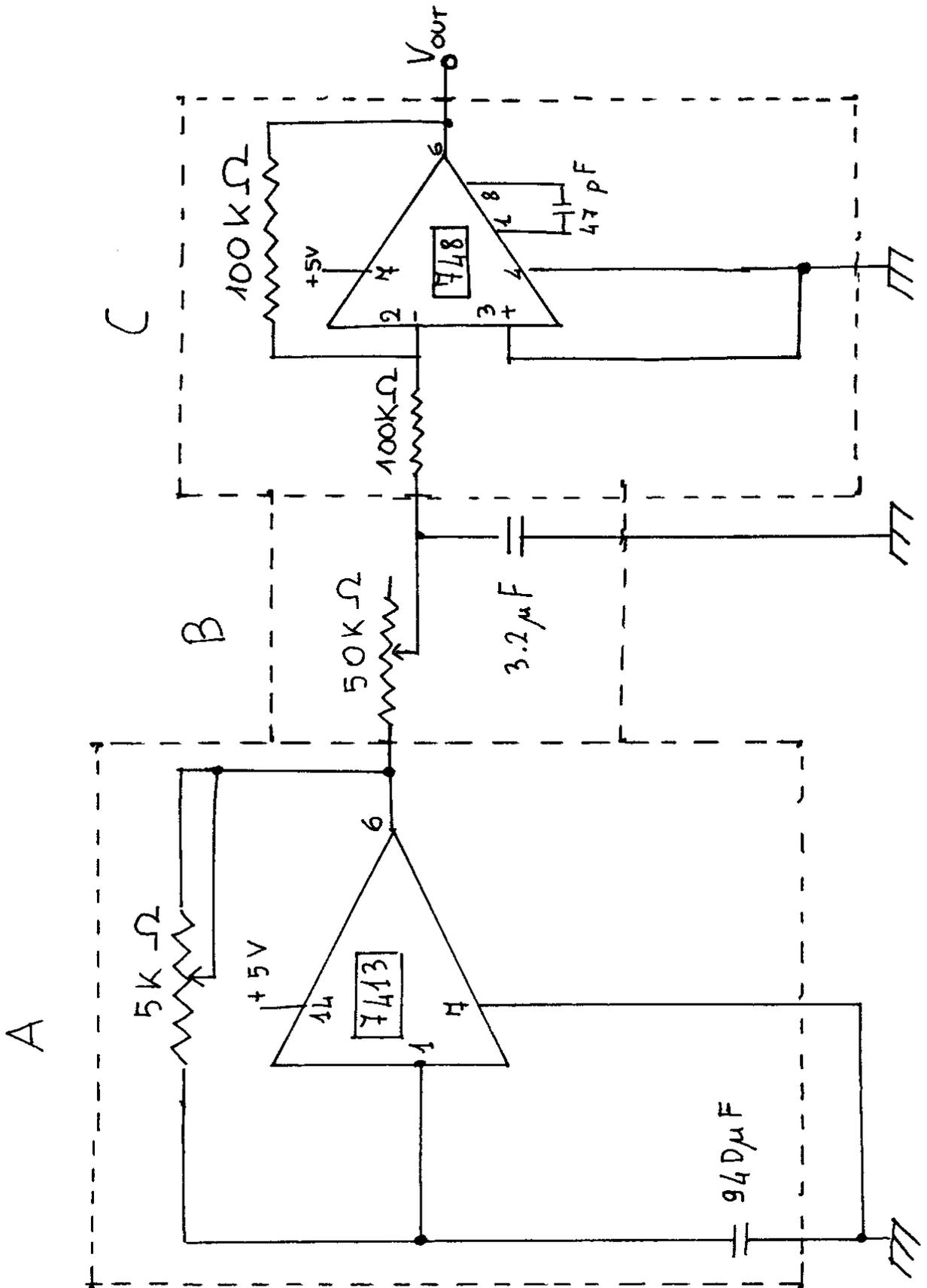
TAVOLA XI
SCHEMA A BLOCCHI DEL SIMULATORE
DI BATTITO CARDIACO



FORME POSSIBILI DEI SEGNALI
GENERATI NEI TRE PUNTI DELLA
CATENA

TAVOLA XII

SCHEMA ELETTRICO DEL SIMULATORE DI BATTITO CARDIACO



L'oscillatore A genera un impulso ogni secondo circa. I fronti di salita sono molto ripidi, la tensione massima è + 5 V. Con un potenziometro si può variare la frequenza notevolmente. La larghezza dell'impulso varia allo stesso modo ma percentualmente molto meno. Un normale filtro passa basso provvede a diminuire parecchio la pendenza dei fronti di salita fino al punto che la tensione massima raggiunta è inferiore a + 2.5 V. Dopo di che comincia subito la discesa, anche questa smorzata. Infine un operazionale con guadagno 1 inverte il segnale che è così adattato per l'ingresso analogico del CRAMER KIT.

Una volta costruito il simulatore si comincia con l'impostare l'organizzazione logica del programma.

Si tenga presente che nel nostro caso abbiamo a disposizione solo 1 K Bytes di ROM e 1K Bytes di RAM. Questo fatto presenta due aspetti interessanti.

Da un lato costituisce un grosso inconveniente in quanto limita le possibilità operative del sistema e nel

nostro caso è non troppo buona la risoluzione dei dati finali che costituiscono la media.(*). D'altra parte essere costretti a lavorare ugualmente con poca memoria a disposizione è molto istruttivo perchè è una situazione che si presenta spesso nella realtà.

La prima cosa da chiarire sulle possibilità del programma è che i dati da immagazzinare sono solo quelli relativi all'impulso, tralasciando il pianerottolo tra l'uno e l'altro. Ovviamente questa è una scelta di lavoro perchè volendo registrare l'intero segnale l'impostazione

(*) Va precisato che la carenza di ROM e di RAM non è implicita nelle capacità del CRAMER KIT che potenzialmente ha la possibilità di arrivare a 32 K Bytes suddivisibili in RAM e in ROM a piacere; ma non ci è sembrato necessario predisporre le basette per l'effettivo utilizzo di ulteriori parti di memoria solo per questa applicazione.

del programma sarebbe stata diversa.

Ci è sembrato inoltre necessario lasciare all'utente la possibilità di scegliere una tensione-soglia al di sopra della quale il segnale va considerato già valido e da considerare come uno degli impulsi da studiare.

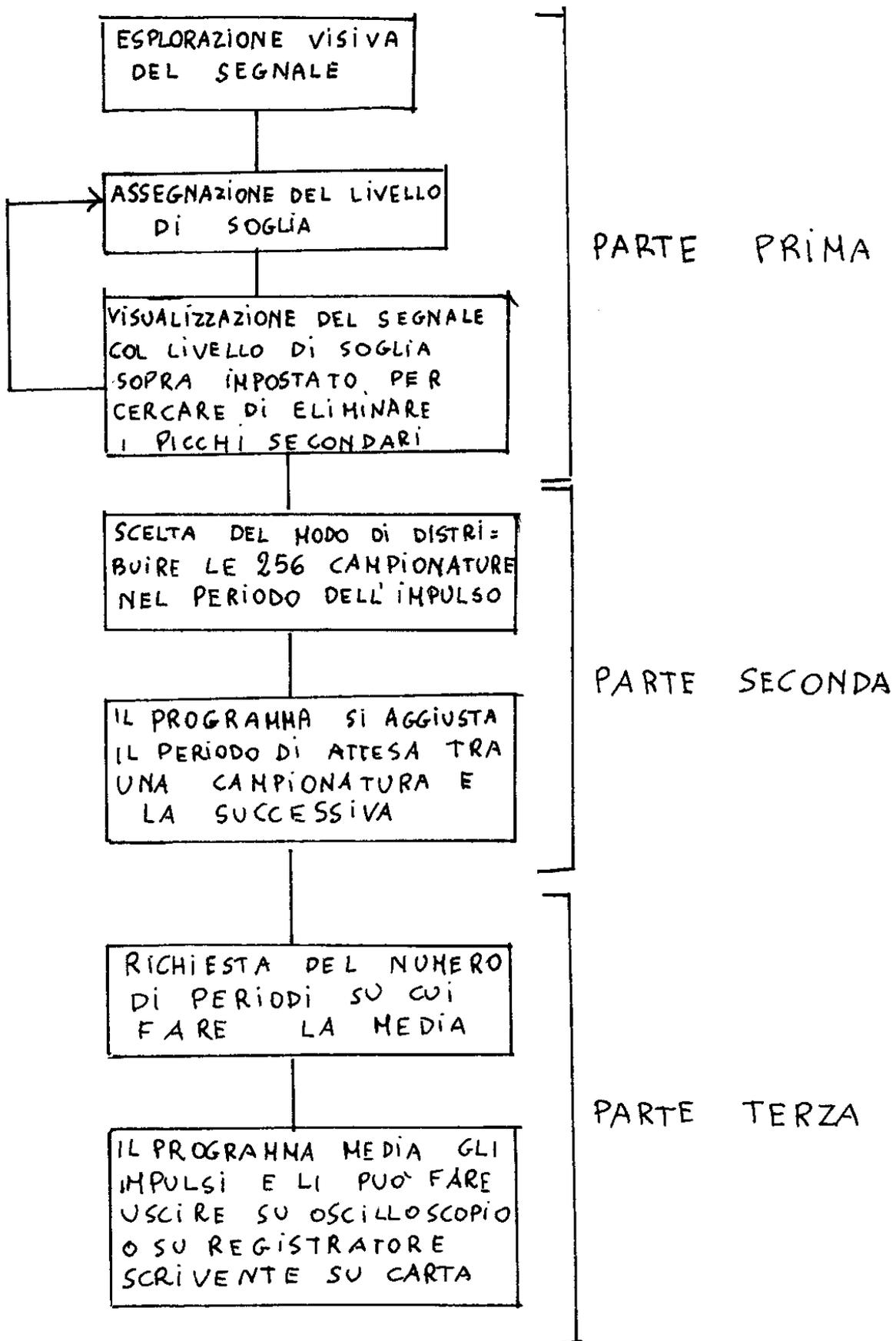
E' chiaro che anche una parte del segnale che precede il momento in cui la tensione supera la soglia e una parte che segue il momento in cui ridiscende al di sotto, possono essere utili da esaminare, ed effettivamente vengono prese in considerazione. Vediamo in che modo.

Per quanto detto prima sulla quantità di memoria disponibile, vengono fatte 256 campionature in tutto per ogni impulso. Per cui il programma prevede di prelevare un certo numero di queste durante il periodo di sovra-soglia e le rimanenti vengono distribuite a metà prima e dopo questo periodo. Questo numero viene richiesto dal programma all'inizio della seconda parte.

Nella Tavola XIII è riportato lo schema a grandi blocchi dell'intero programma.

TAVOLA XIII

PROGRAMMA MEDIA



Il problema della frequenza variabile non ha creato molte difficoltà per quanto riguarda la campionatura dei dati utili all'arrivo di ogni impulso. Infatti il programma, dopo aver memorizzato i dati relativi a un impulso secondo le specifiche stabilite in precedenza, si mette in un LOOP di attesa di lunghezza variabile fino all'arrivo del successivo. In questo stato di attesa il μP campiona in continuazione e confronta ogni dato in ingresso col livello di soglia stabilito. Decide quindi di riconoscere il nuovo impulso quando la tensione di ingresso supera di nuovo la soglia. In questa logica non sono stati previsti dei picchi anomali di ampiezza circa uguale a quella degli impulsi da studiare; ma c'è la possibilità di ignorare tutti i picchi di ampiezza inferiore agli impulsi principali. Infatti è sufficiente alzare il livello di soglia al di sopra di questi picchi. Di conseguenza è possibile che si restringa molto il periodo di sovrasoglia ma basta richiedere che molte campionature avvengano prima e dopo questo periodo cosicchè l'intero impulso risulta ugualmen-

te registrato.

La terza parte ha presentato difficoltà maggiori a causa di questo fatto.

Nel momento in cui il segnale supera la soglia devono essere già state fatte un certo numero di campionature valide per la media finale. Inoltre non è possibile prevedere quando giunge l'impulso nè quanto durerà il pianerottolo. Per questo siamo ricorsi alla tecnica del BUFFER ROTANTE. Lo spazio di memoria a disposizione dei dati (256Bytes nel nostro caso) non viene considerato come una stringa aperta che comincia dal primo Byte e finisce con l'ultimo ma come un 'anello chiuso' di memoria in cui l'ultima locazione precede immediatamente la prima. Questo anello non inizia più dall'indirizzo più basso ma da un punto qualunque in base a un indice che ruota su questo anello.

In pratica avviene questo: all'inizio le 256 locazioni di memoria sono vuote. Si comincia a riempirle come se si trattasse di dati validi, cioè effettuando campionature

intervallate dal tempo di attesa calcolato nella seconda parte (Tavola XIII). Finchè ci si trova nella zona del pianerottolo può accadere che si riempiano più volte le 256 locazioni. Ogni volta che si giunge alla 256esima si ricomincia dalla prima come se questa fosse la successiva. Contemporaneamente all'immissione di ogni dato un indice scatta in avanti di un posto ad indicare che di lì in poi comincia la zona di memoria disponibile. Anche l'indice, giunto al 256esimo posto si posiziona all'inizio. Quando giunge l'impulso, quando cioè la tensione supera la soglia, sono ancora disponibili i 256 dati precedentemente prelevati ad intervalli regolari e tutti buoni per la media, se necessario.

Vengono ora effettuate un certo numero di campionature sull'impulso secondo la precedente richiesta dell'utente e alla fine il Buffer Rotante conterrà ancora 256 dati validi per la media sempre centrati attorno alla parte significativa del segnale.

A questo punto si 'riapre' l'anello nel punto indicato

dall'indice e i dati si trovano automaticamente ordinati.

Vanno poi sommati agli altri dei precedenti impulsi in modo che i massimi di tutti gli impulsi coincidano.

Questo particolare modo di mettere a registro gli impulsi è una scelta arbitraria ma suggerita dal fatto che talvolta anche l'operatore umano, quando effettua delle misure sui tracciati reali, prende i picchi come punto di riferimento perchè spesso sono ben definibili all'interno del periodo.

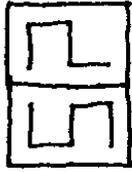
Il programma termina con la registrazione dell'impulso medio su oscilloscopio o, a scelta, su registratore scrivente su carta.

USO PRATICO DEL PROGRAMMA MEDIA

- Connettere l'oscilloscopio o il registratore scrivente su carta alla presa OUT sul retro del CRAMER KIT. Escono tensioni comprese tra 0 e -2.5 V.
- Alimentare i due DAC con +15 V. e -15 V.
- Il segnale da studiare deve essere compreso tra 0 e -2.5 V. e entra nella presa IN sul retro del CRAMER KIT.
- All'accensione i Display contengono: S ?

Le fasi successive sono illustrate a partire dalla pagina seguente.

DISPLAY

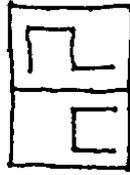


SIGNIFICATO IMPOSTARE E CARICARE CON STROBE INPUT:

S? Livello di soglia
Un numero inferiore a 256 per definire il livello di tensione al di sopra della quale il segnale è considerato valido
es.: numero = 16 soglia = 0.16 V. circa
numero = 256 soglia = 2.5 V.

PER VARIARE LA SOGLIA SPINGERE RESET E IMPOSTARE UN VALORE DIVERSO

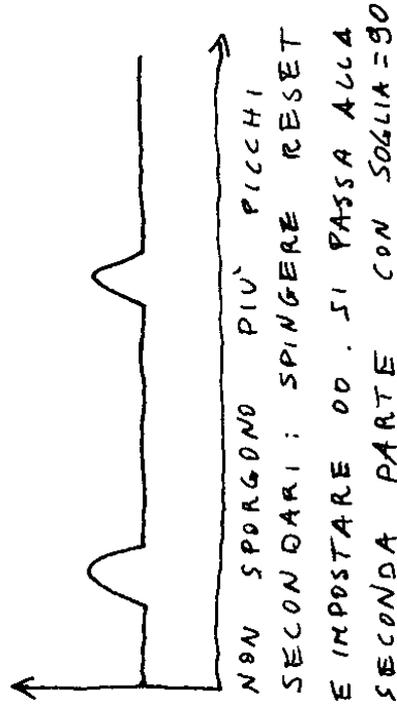
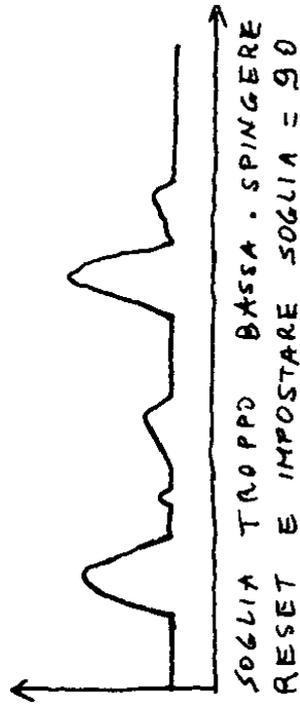
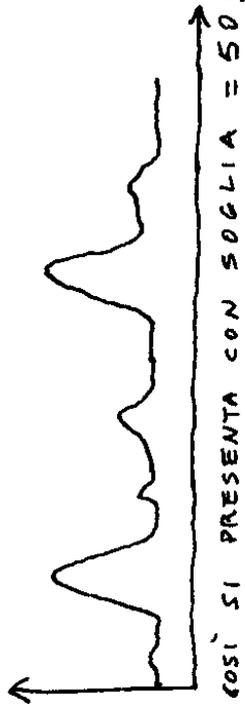
SE LA SOGLIA E' GIUSTA SPINGERE RESET MA IMPOSTARE 00 che viene considerato come un comando di passare alla seconda parte:



N? Numero di campionature
Un numero inferiore a 227 che definisce quante campionature devono essere effettuate nel periodo di sovrasoglia.
Le rimanenti vengono distribuite prima e dopo detto periodo. Se l'impulso è quasi tutto sporgente dalla soglia e le code sono brevi impostare un numero

ESEMPIO QUALITATIVO:

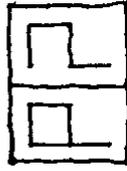
DATO UN SEGNALE IN INGRESSO:



DISPLAY SIGNIFICATO IMPOSTARE E CARICARE CON STROBE INPUT

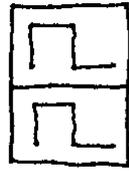
alto. Se l'impulso che sporge è stretto perchè la soglia è alta e le code sono importanti e di durata notevole impostare un numero basso.

Dopo l'introduzione di N il display di sinistra inizia un conto alla rovescia scattando ad ogni impulso. Dopo 8 periodi il programma si è già aggiustato al meglio il tempo che dovrà attendere fra una campionatura e l'altra perchè si verifichino le condizioni imposte dall'utente. Dopo di che:



P? Periodi su Il limite al numero di impulsi che si vogliono mediare è 255:
cui fare la la somma dei dati potrebbe superare la capacità della memoria.
media

Trascorsi gli impulsi richiesti appare:

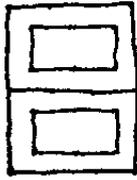


?? Oscilloscopio Se si usa il registratore scrivente su carta impostare 00
o registratore altrimenti impostare un altro numero. I dati sono 256 ed escono scrivente? in continuazione. Nel loro insieme disegnano l'impulso medio senza il pianerottolo che li separa.

CASI DI ERRORE

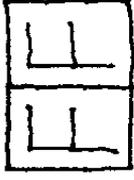
DISPLAY

SIGNIFICATO:



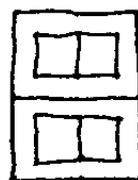
lampeg-
giante

Il tempo di attesa fra le campionature dovrebbe essere troppo breve per soddisfare le condizioni imposte. Ad esempio quando con un impulso troppo stretto si richiedono troppe campionature all'interno. Spingere RESET e impostare valori diversi.



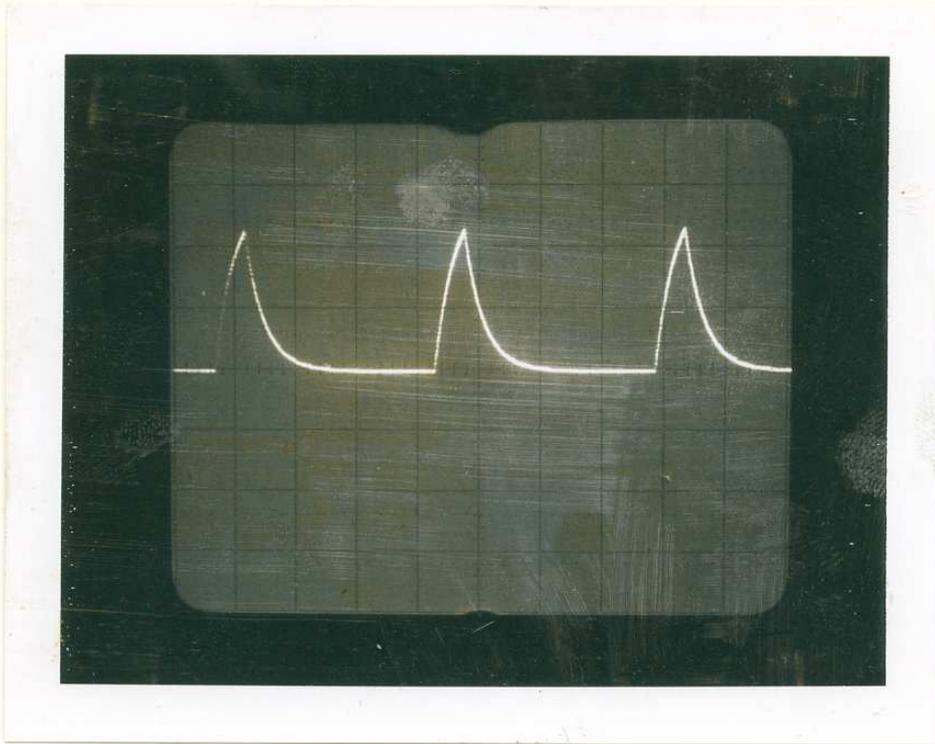
lampeg-
giante

Il tempo di attesa tra le campionature dovrebbe essere troppo lungo per soddisfare le condizioni imposte. Su un impulso molto largo si richiedono poche campionature. Spingere RESET e impostare valori diversi.

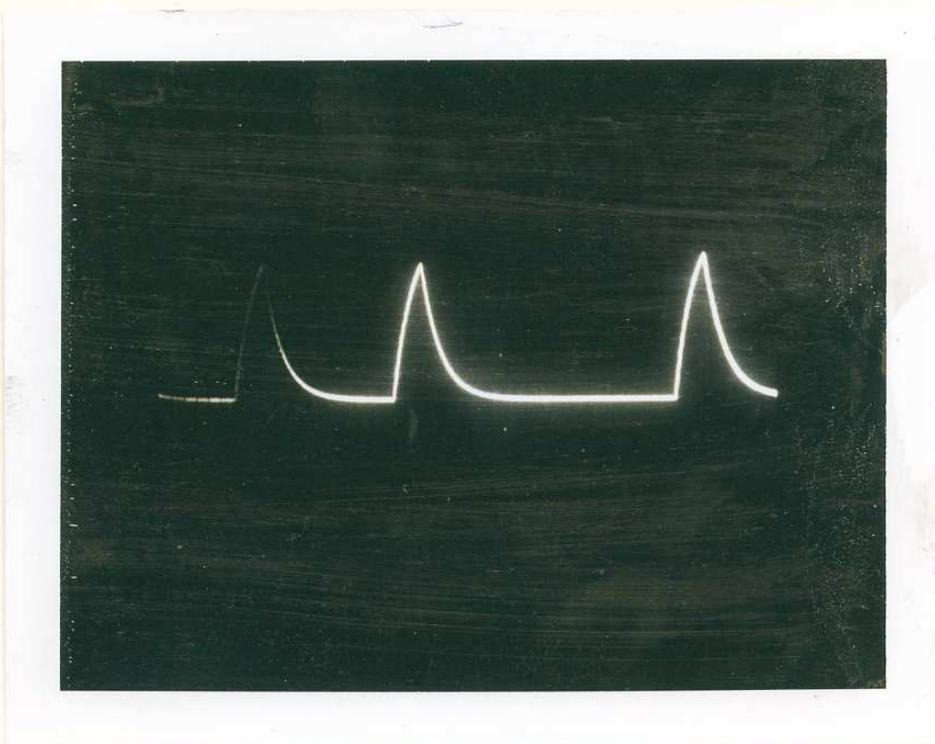


lampeg-
giante

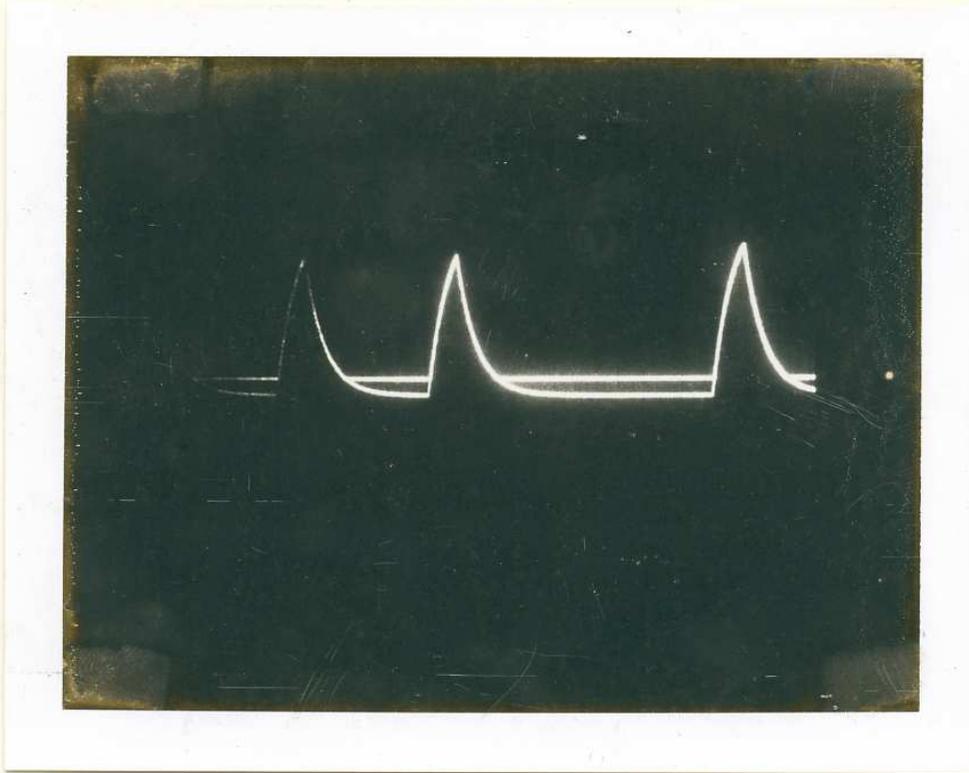
E' stato richiesto un numero di campionature in sovrasoglia maggiore di 227 non c'è spazio per le code. RESET e rifare con N minore o uguale di 227.



SEGNALE DEL SIMULATORE E, SOVRAPPOSTO, IL SEGNALE IN USCITA CON
SOGLIA = 1 0.5 sec./div. 1 V./div.

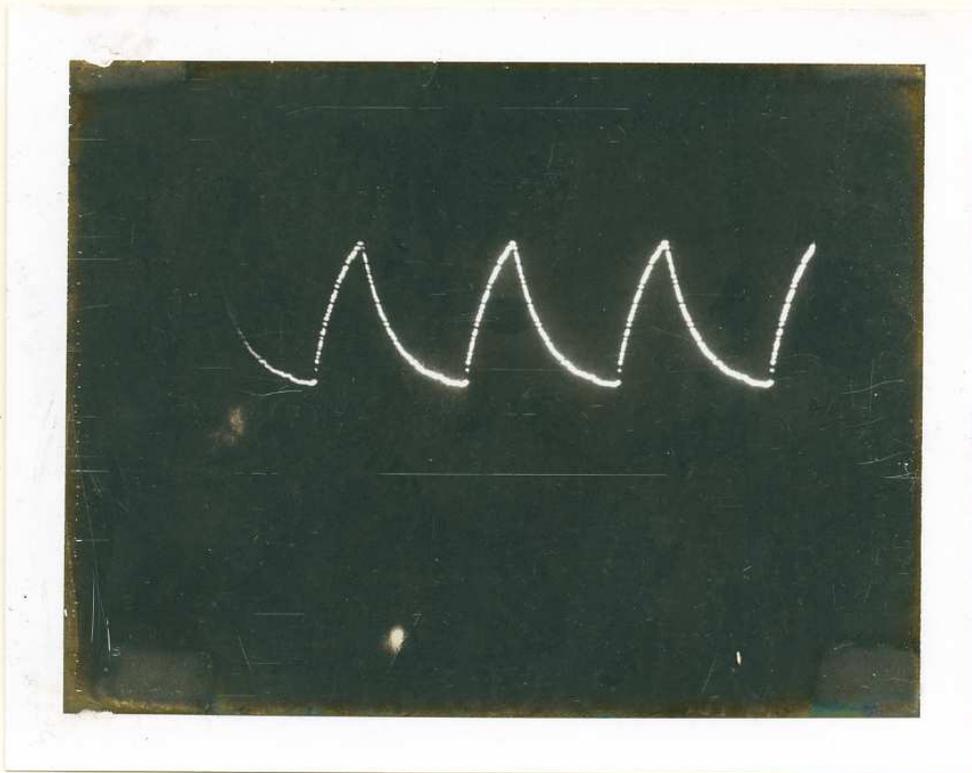


SOLO SEGNALE DEL SIMULATORE, VARIATO IN FREQUENZA
0.5 sec./div 1 V./div.



SEGNALE PRECEDENTE PIU' SEGNALE CAMPIONATO CON SOGLIA = 32

0.5 sec./div 1 V./div.



MEDIA SU SETTE IMPULSI 5millisec./div. 1 V./div.

APPENDICE

LISTA DEL PROGRAMMA FOURIER

DIAGRAMMA DI FLUSSO DEL PROGRAMMA MEDIA

LISTA DEL PROGRAMMA MEDIA

;PROGRAMMA FOURIER RICOMPOSTO . GIRA SUL KRAMER KIT
;SOMMA UNA SINUSOIDE CON UN NUMERO QUALUNQUE DI
;ARMONICHE . I DATI SI IMPOSTANO NORMALMENTE SUL
;KRAMER KIT .
;CARICARE IN ORDINE QUESTI DATI:
;NUMERO TOTALE DEI PARAMETRI (QUATTRO PER OGNI ARMONICA)
;FREQUENZA (IN UNITA' DELLA FONDAMENTALE)
;AMPIEZZA (NUM/DEN DELLA FONDAMENTALE)
;SFASAMENTO (IN 255ESIMI DELLA FONDAMENTALE)
;COMUNQUE SUL DISPLAY APPARE LA RICHIESTA DEL
;DATO DA CARICARE AL MOMENTO

0400	ORG	0400H	
2400	STACK	EQU	2400H
0098	WORD1	EQU	98H
0089	WORD2	EQU	89H
3C03	CNTR1	EQU	3C03H
3C07	CNTR2	EQU	3C07H
3C0B	CNTR3	EQU	3C0BH
3C09	DAC2	EQU	3C09H
3C01	MSD	EQU	3C01H
3C05	LSD	EQU	3C05H
3C00	SWCR	EQU	3C00H
3C04	SWCL	EQU	3C04H
0700	FOND	EQU	0700H
2000	RAM1	EQU	2000H
2100	RAM2	EQU	2100H
2200	RAM3	EQU	2200H
2300	PARAM	EQU	2300H
0600	LISTA	EQU	0600H
0400	ERROR	EQU	0400H
0080	TARAT	EQU	80H
0400	310024	LXI	SP,STACK
0403	3E98	MVI	A,WORD1
0405	32033C	STA	CNTR1
0408	32073C	STA	CNTR2
040B	3E89	MVI	A,WORD2
040D	320B3C	STA	CNTR3
0410	3E80	MVI	A,TARAT
0412	32093C	STA	DAC2

;PORTE CONTROLLO 8255

;INDIRIZZO DEI DISPLAY

; = DEGLI INTERRUTTORI

;FONDAMENTALE

;PRIMO QUARTO DI RAM

;SECONDO = =

;TERZO = =

;INDIRIZZO DEI PARAMETRI

;LISTA DELLE LETTERE DA VISUALIZ

;IN CASO DI ERRORE RICOMINCIA

;ESCONO -1,75 VOLT

;CONTROLLO INTERRUTTORI

;E DISPLAY

;CONTROLLO CONVERTITORE

;D'USCITA

;TARATURA PER IL CENTROBANDA

;TRASFERISCE LA FONDAMENTALE IN RAM
;D,E = FOND H,L = RAM3

0415	110007	LXI	D,FOND
0418	210022	LXI	H,RAM3

```
041B 1A      LSBYT: LDAX   D      ;BYTE MENO SIGNIFICATIVO
041C 77              MOV   M,A
041D 1C              INR   E
041E 2C              INR   L
041F C21B04      JNZ   LSBYT
0422 210021      LXI   H,RAM2 ;ZONA DEI BYTES PIU' SIGNIFICATI
0425 AF              XRA   A      ;AZZERA A
0426 77      MSBYT: MOV   M,A      ;AZZERA TUTTI I BYTES
0427 2C              INR   L      ;PIU' SIGNIFICATIVI
0428 C22604      JNZ   MSBYT
```

```
;CARICA I PARAMETRI :
;NUMERO TOTALE PARAMETRI,
;FREQUENZA , AMPIEZZA (NUMERATORE)
;AMPIEZZA (DENOMINATORE) , SFASAMENTO
```

```
042B 210006      LXI   H,LISTA
042E CD4B04      CALL  DISPL  ;??
0431 CD5604      CALL  STRB
0434 110023      LXI   D,PARAM
0437 12              STAX  D      ;SALVA IN RAM DUE VOLTE
0438 13              INX   D      ;PER GLI USI SUCCESSIVI
0439 12              STAX  D
```

```
;COMINCIA IL CARICAMENTO PARAMETRI
```

```
043A CD4B04      ARMON: CALL  DISPL
043D CD5604      CALL  STRB
0440 13              INX   D
0441 12              STAX  D
0442 CD7B04      CALL  TEST1 ;DECREMENTA IL NUMERO PARAMETRI
0445 C23A04      JNZ   ARMON ;FINITI I PARAMETRI?
0448 C38304      JMP   MEMOR
```

```
;VISUALIZZA SUI DISPLAY H,L PUNTATORI IN ROM
```

```
044B 7E      DISPL: MOV   A,M
044C 32013C      STA   MSD
044F 23              INX   H
0450 7E              MOV   A,M
0451 32053C      STA   LSD
0454 23              INX   H
0455 C9              RET
```

```
;CARICA I DATI DAGLI INTERRUTTORI IN A
```

```
0456 3A003C      STRB: LDA   SWCR
```

```
0459 E602      ANI      02H
045B 47        MOV      B,A
045C D5        PUSH     D
045D CD6E04    CALL    WAIT1
0460 D1        POP      D
0461 3A003C    LDA      SWCR
0464 E602      ANI      02H
0466 B8        CMP      B
0467 CA5604    JZ       STRB
046A 3A043C    LDA      SWCL
046D C9        RET
```

```
;ATTESA DI 25 MILLI SEC. ANTIRIMBALZO
;USA D,E COME CONTATORI
```

```
046E 1619    WAIT1:  MVI      D,25
0470 1E64    RIT1:   MVI      E,100
0472 1D      RIT2:   DCR      E
0473 C27204  JNZ     RIT2
0476 15      DCR      D
0477 C27004  JNZ     RIT1
047A C9      RET
```

```
;CONTROLLA SE SONO FINITI I PARAMETRI DA
;CARICARE
```

```
047B 3A0023  TEST1:  LDA      PARAM
047E 3D      DCR      A
047F 320023  STA     PARAM
0482 C9      RET
```

```
;SISTEMA LE MEMORIE A 16 BIT CON LA SOMMA DI
;TUTTE LE ARMONICHE
;H,L = ROM FONDAMENTALE
;D,E = RAM1
;B,C = PARAMETRI
```

```
0483 3A0123  MEMOR:  LDA      PARAM+1 ;RISISTEMA IL
0486 320023  STA     PARAM ;CONTATORE
0489 110020  LXI     D,RAM1
048C 010223  LXI     B,PARAM+2
048F 210007  FREQ:   LXI     H,FOND
0492 03      INX     B ;TIENE CONTO DELLO
0493 03      INX     B ;SFASAMENTO
0494 03      INX     B ;=
0495 0A      LDAX   B ;=
0496 6F      MOV     L,A
```

```

0497 0B          DCX      B          ;CALCOLA L'AMPIEZZA
0498 0A          AMP:    LDAX     B          ;=
0499 05          PUSH    B          ;=
049A 47          MOV     B,A        ;=
049B 7E          MOV     A,M        ;=
049C CDC804      CALL    DIV        ;=
049F 01          POP     B          ;=
04A0 12          STAX    D          ;PROVVISORIAMENTE IN RAM1
04A1 0B          DCX      B          ;ANCORA AMPIEZZA
04A2 0A          LDAX     B          ;=
04A3 05          PUSH    B          ;=
04A4 47          MOV     B,A        ;=
04A5 1A          LDAX     D          ;=
04A6 CDBF04      CALL    MULTI       ;=
04A9 01          POP     B          ;=
04AA 12          STAX    D          ;RISULTATO FINALE IN RAM1
04AB 1C          INR     E          ;=
04AC F5          PUSH    PSW         ;=
04AD E5          PUSH    H          ;=
04AE CCDB04      CZ       SOMMA      ;FINITO IL CALCOLO DELLA
                                ;ARMONICA LA SOMMA ALLE
                                ;PRECEDENTI IN RAM2,3 E
                                ;LIBERA RAM1 PER LA PROSSIMA

04B1 E1          POP     H          ;=
04B2 F1          POP     PSW         ;=
04B3 CAEF04      JZ      TEST2       ;=
04B6 0B          DCX      B          ;TIENE CONTO DELLA FREQUENZA
04B7 0A          LDAX     B          ;=
04B8 85          ADD     L          ;=
04B9 6F          MOV     L,A        ;=
04BA 03          INX     B          ;SI POSIZIONA SULL'AMPIEZZA
04BB 03          INX     B          ;=
04BC C39804      JMP     AMP         ;=

```

```

;MULTIPLICAZIONE
;NON C'E' MAI OVERFLOW

```

```

04BF 05          MULTI:  DCR      B
04C0 CAC704      JZ      EXITM
04C3 87          ADD     A
04C4 C3BF04      JMP     MULTI
04C7 09          EXITM:  RET

```

```

;DIVISIONE: A/B---->C---->A

```

```

04C8 4F          DIV:    MOV     C,A
04C9 78          MOV     A,B
04CA 87          ORA     A          ;DIVISORE = 0?

```

```
04CB CA0004      JZ      ERROR      ;DA CAPO IL PROGRAMMA
04CE 79          MOV      A,C
04CF 0E00        MVI      C,00
04D1 90          DIVI:   SUB      B
04D2 DAD904      JC      EXITD
04D5 0C          INR      C
04D6 C3D104      JMP      DIVI
04D9 79          EXITD:  MOV      A,C
04DA C9          RET
```

```
;SOMMA RAM1 IN RAM3 TENENDO CONTO DEL
;RIPORTO IN RAM2
```

```
04DB 210020      SOMMA:  LXI      H, RAM1
04DE 7E          SOMM1:  MOV      A,M
04DF 24          INR      H          ;
04E0 24          INR      H          ;H,L = RAM3
04E1 86          ADD      M
04E2 77          MOV      M,A
04E3 D2E804      JNC      PROX      ;IN CASO DI NON OVERFLOW
04E6 25          DCR      H          ;H.L = RAM2
04E7 34          INR      M
04E8 2620        PROX:   MVI      H,20H
04EA 2C          INR      L
04EB C2DE04      JNZ      SOMM1
04EE C9          RET
```

```
;FINITE LE ARMONICHE DA SOMMARE?
```

```
04EF 3A0023      TEST2:  LDA      PARAM
04F2 3D          DCR      A          ;CONSUMA QUATTRO PARAMETRI
04F3 3D          DCR      A          ;OGNI ARMONICA
04F4 3D          DCR      A
04F5 3D          DCR      A
04F6 320023      STA      PARAM
04F9 CA0205      JZ      MAX
04FC 03          INX      B          ;SI POSIZIONA ALL'INIZIO
04FD 03          INX      B          ;DEI SUCCESSIVI PARAMETRI
04FE 03          INX      B
04FF C38F04      JMP      FREQ
```

```
;PER L'USCITA I DATI DA 16 BIT DEVE DIVIDERLI
;E RIDURLI A 8 BIT . PER ORA CERCA IL MASSIMO
;TRA I BYTES PIU' SIGNIFICATIVI
;ALLA FINE E INDIRIZZA IL MASSIMO
```

```
0502 210021      MAX:   LXI      H, RAM2
0505 5E          MOV      E,M
```

```
0506 2C      RINR:  INR      L
0507 CA1305      JZ      DDIVI
050A 7E      MOV      A,M
050B 8B      CMP      E
050C DA0605      JC      RINR
050F 5F      MOV      E,A
0510 C30605      JMP      RINR
```

```
;DIVISIONE
;H,L = DIVIDENDO 16 BIT
;E = DIVISORE 8 BIT
;D = RISULTATO
```

```
0513 1C      DDIVI:  INR      E      ;DIVIDE PER MAX+1
0514 CA0004      JZ      ERROR  ;DA CAPO SE =0
0517 010021      LXI      B, RAM2
051A 0A      PREPD:  LDAX     B      ;BYTE PIU' SIGNIFICATIVO
051B 67      MOV      H,A
051C 04      INR      B      ;BYTE MENO SIGNIFICATIVO
051D 0A      LDAX     B      :
051E 6F      MOV      L,A      :
051F C5      PUSH     B
0520 CD3005      CALL    DIVID
0523 7A      MOV      A,D
0524 C1      POP      B
0525 05      DCR      B
0526 05      DCR      B      ;PUNTA SU RAM1
0527 02      STAX     B      ;COLLOCA IL RISULTATO
0528 0C      INR      C      ;UN PASSO AVANTI
0529 CA4E05      JZ      REQST
052C 04      INR      B      ;PUNTA SU RAM2
052D C31A05      JMP      PREPD  ;ALTRA DIVISIONE
```

```
;H,L DIVIDENDO
;E DIVISORE
;D RISULTATO
;B,C DATI PROVVISORI
```

```
0530 010000      DIVID:  LXI      B,00
0533 1600      MVI      D,00
0535 7B      SCATT:  MOV      A,E
0536 81      ADD      C
0537 4F      MOV      C,A
0538 D23C05      JNC     CONFR
053B 04      INR      B
053C 7C      CONFR:  MOV      A,H
053D 8B      CMP      B
053E DA4D05      JC      EXIT
```

```
0541 CA4805      JZ      SECND
0544 14          ALTRO:  INR      D
0545 C33505      JMP      SCATT
0548 7D          SECND:  MOV      A,L
0549 B9          CMP      C
054A D24405      JNC      ALTRO
054D C9          EXIT:   RET
```

```
;DATI IN USCITA SULL'OSCILLOSCOPIO O SULLA
;SCRIVENTE?
;IMPOSTARE 00 SE SCRIVENTE
;ALTRO NUMERO SE OSCILLOSCOPIO
```

```
054E 3E65      REQST:  MVI      A,65H    ;DISPLAY: ??
0550 32053C     STA      LSD
0553 32013C     STA      MSD
0556 CD5604     CALL     STRB
0559 4F         MOV      C,A
055A 3E7F      MVI      A,7FH    ;DISPLAY: 88
055C 32053C     STA      LSD
055F 32013C     STA      MSD
0562 C36505     JMP      FOUR
```

```
;VISUALIZZA LA FORMA D'ONDA RISULTANTE
```

```
0565 210020     FOUR:   LXI      H,RAM1
0568 7E         FOUR1:  MOV      A,M
0569 32093C     STA      DAC2
056C AF         XRA      A        ;AZZERA A
056D B9         CMP      C
056E CC7505     CZ      WAIT2
0571 2C         INR      L
0572 C36805     JMP      FOUR1
```

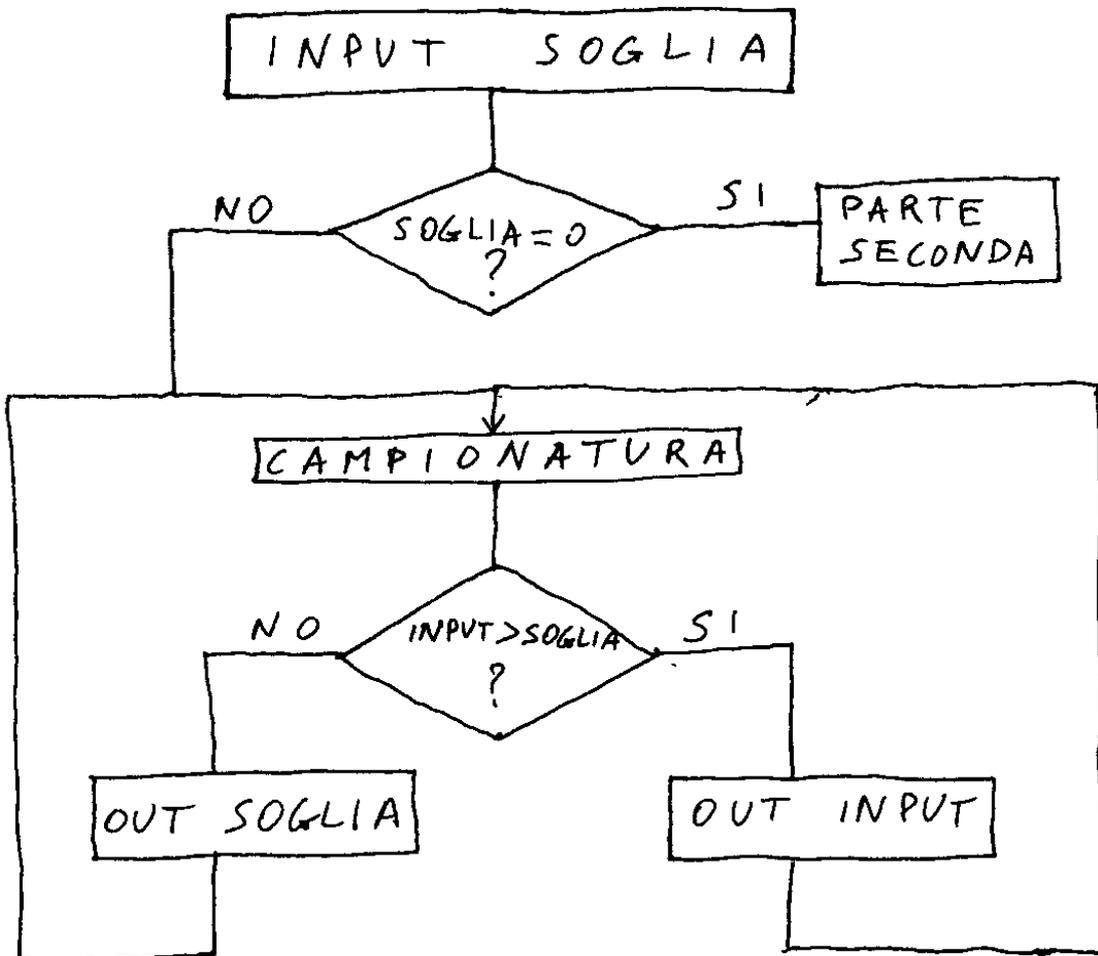
```
;LUNGA ATTESA PER ADATTARE L'USCITA
;ALLA SCRIVENTE: 0.4 SEC. PER DATO
```

```
0575 16FF      WAIT2:  MVI      D,0FFH
0577 1EFF      RIT3:   MVI      E,0FFH
0579 1D        RIT4:   DCR      E
057A C27905     JNZ      RIT4
057D 15        DCR      D
057E C27705     JNZ      RIT3
0581 C9        RET
0000          END
```

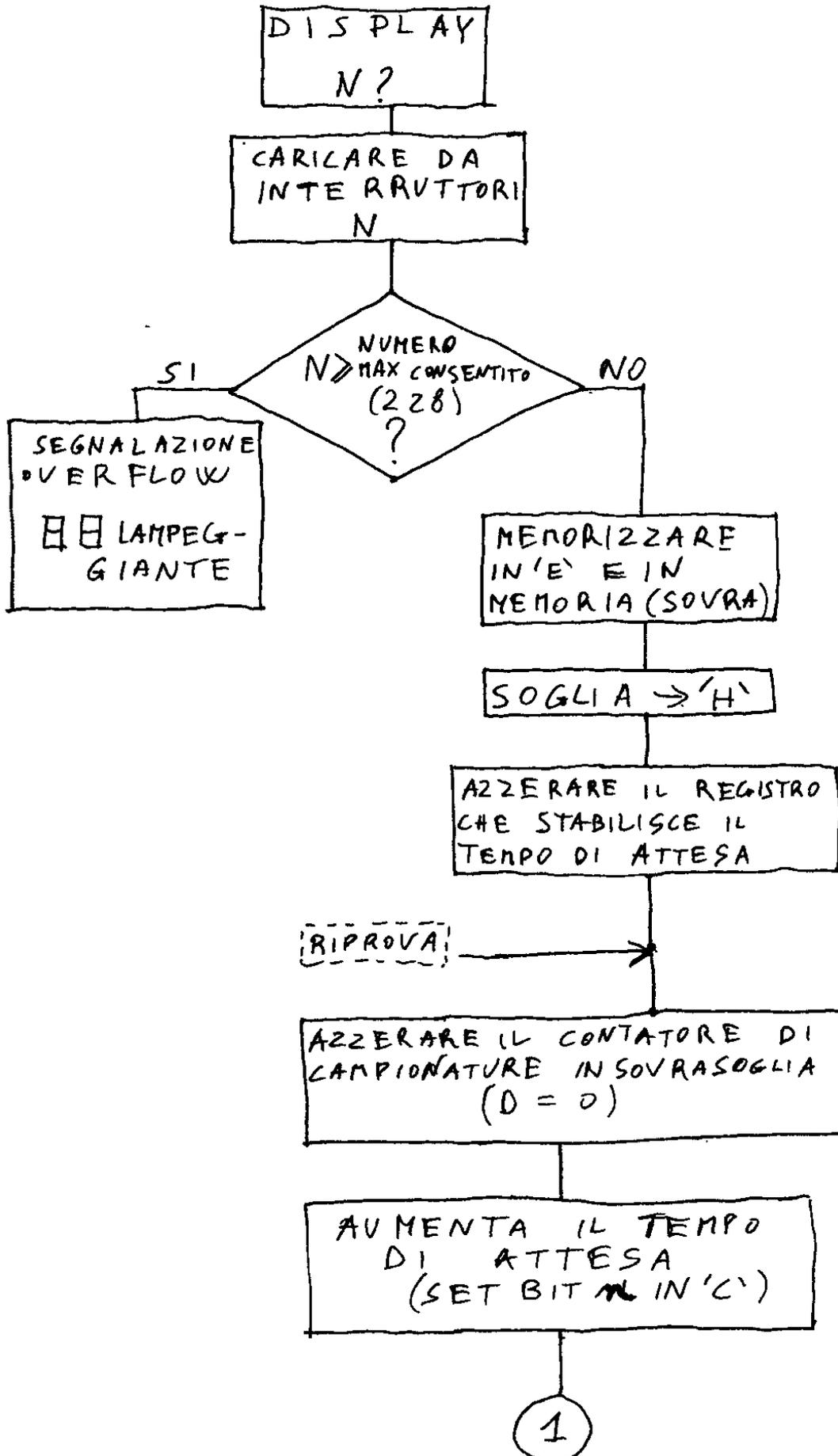
ALTR0 0544	AMP 0498	ARMON 043A	CNTR1 3C03
CNTR2 3C07	CNTR3 3C0B	CONFR 053C	DAC2 3C09
DDIVI 0513	DISPL 044B	DIV 04C8	DIVI 04D1
DIVID 0530	ERROR 0400	EXIT 054D	EXITD 04D9
EXITM 04C7	FOND 0700	FOUR 0565	FOUR1 0568
FREQ 048F	LISTA 0600	LSBYT 041B	LSD 3C05
MAX 0502	MEMOR 0483	MSBYT 0426	MSD 3C01
MULTI 04BF	PARAM 2300	PREPD 051A	PROX 04E8
RAM1 2000	RAM2 2100	RAM3 2200	REQST 054E
RINR 0506	RIT1 0470	RIT2 0472	RIT3 0577
RIT4 0579	SCATT 0535	SECNO 0548	SOMM1 04DE
SOMMA 04DB	STACK 2400	STRB 0456	SWCL 3C04
SWCR 3C00	TARAT 0080	TEST1 047B	TEST2 04EF
WAIT1 046E	WAIT2 0575	WORD1 0098	WORD2 0089

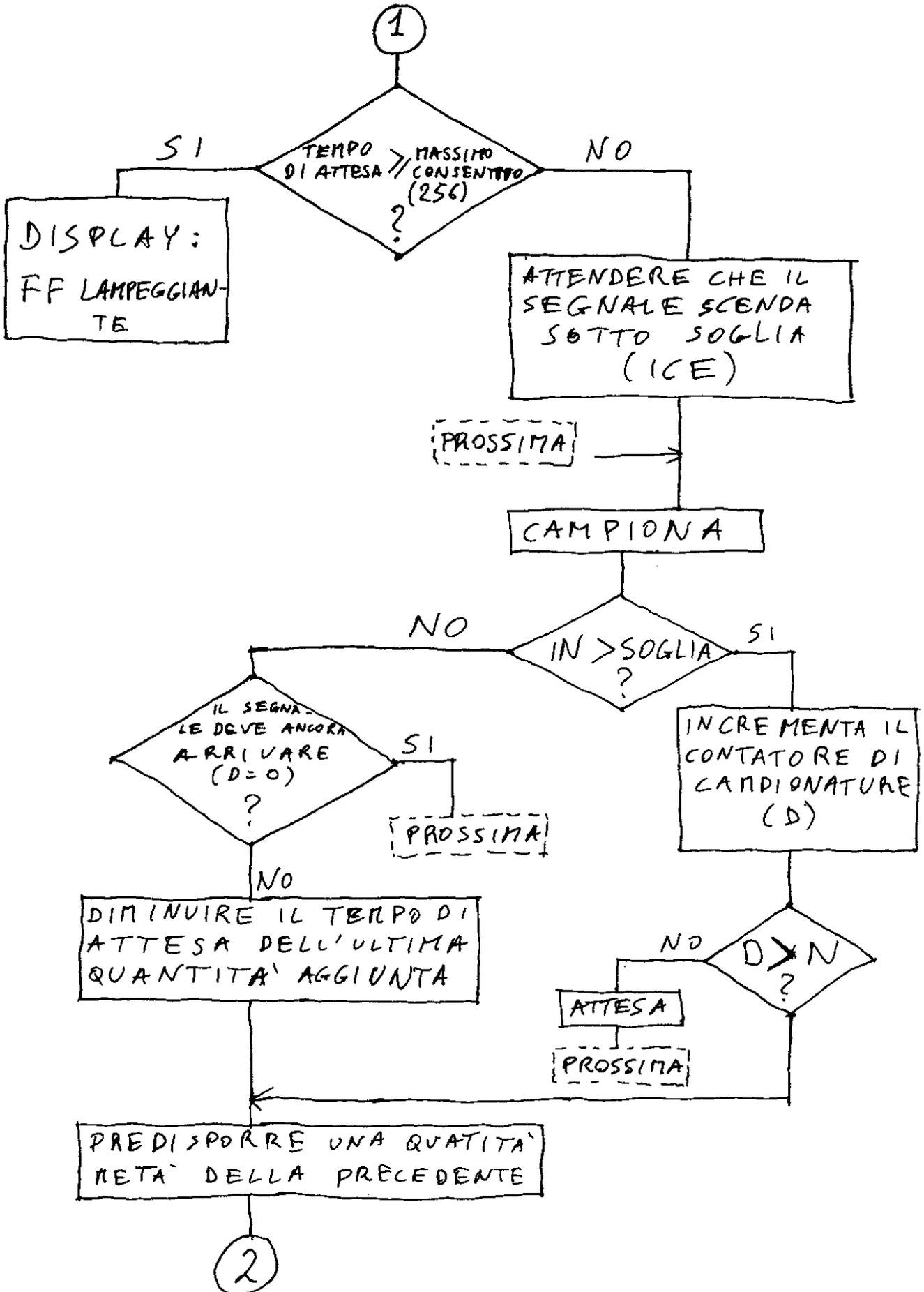
DIAGRAMMA DI FLUSSO DEL PROGRAMMA MEDI A

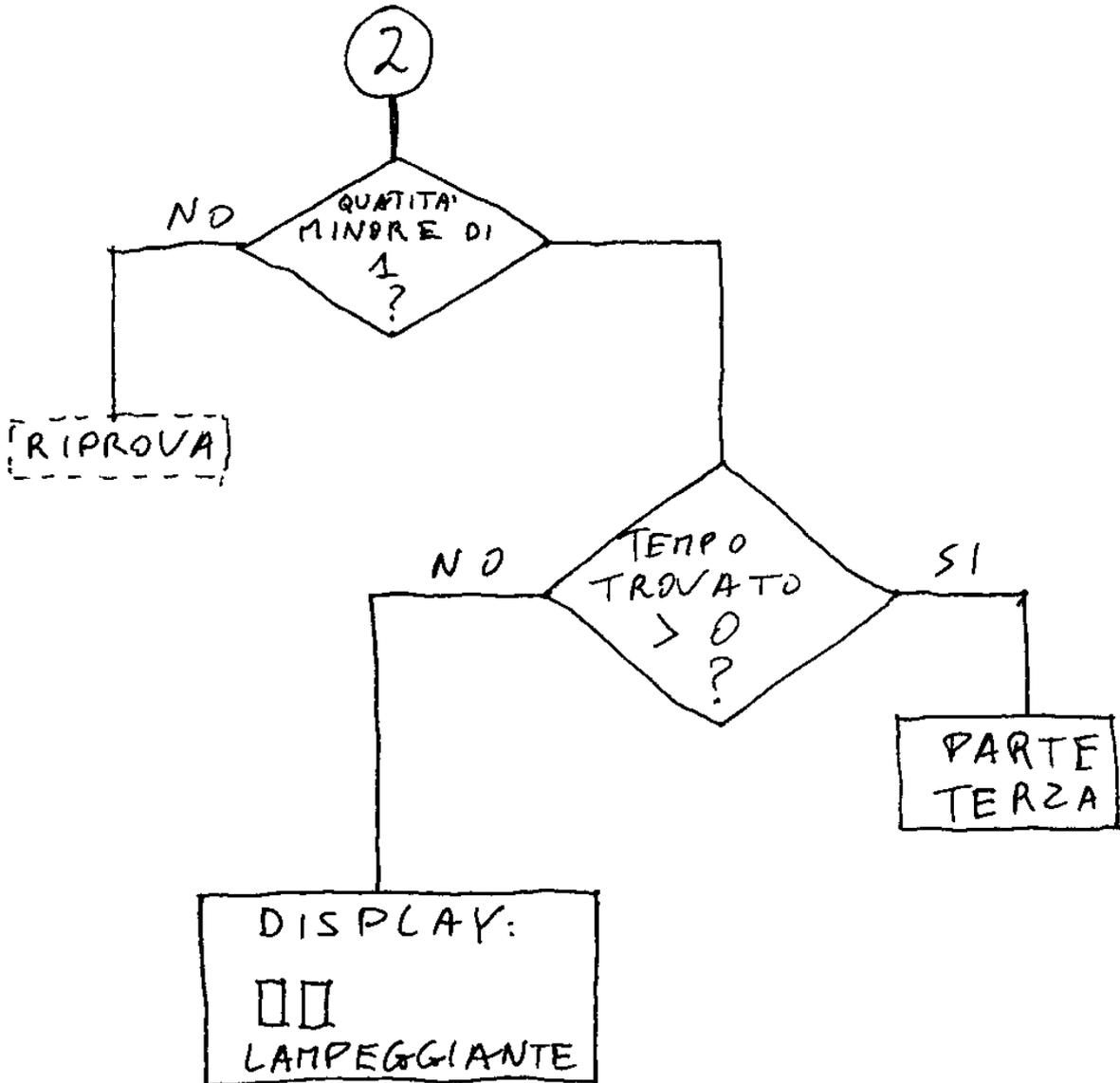
PARTE PRIMA DETERMINARE LA SOGLIA



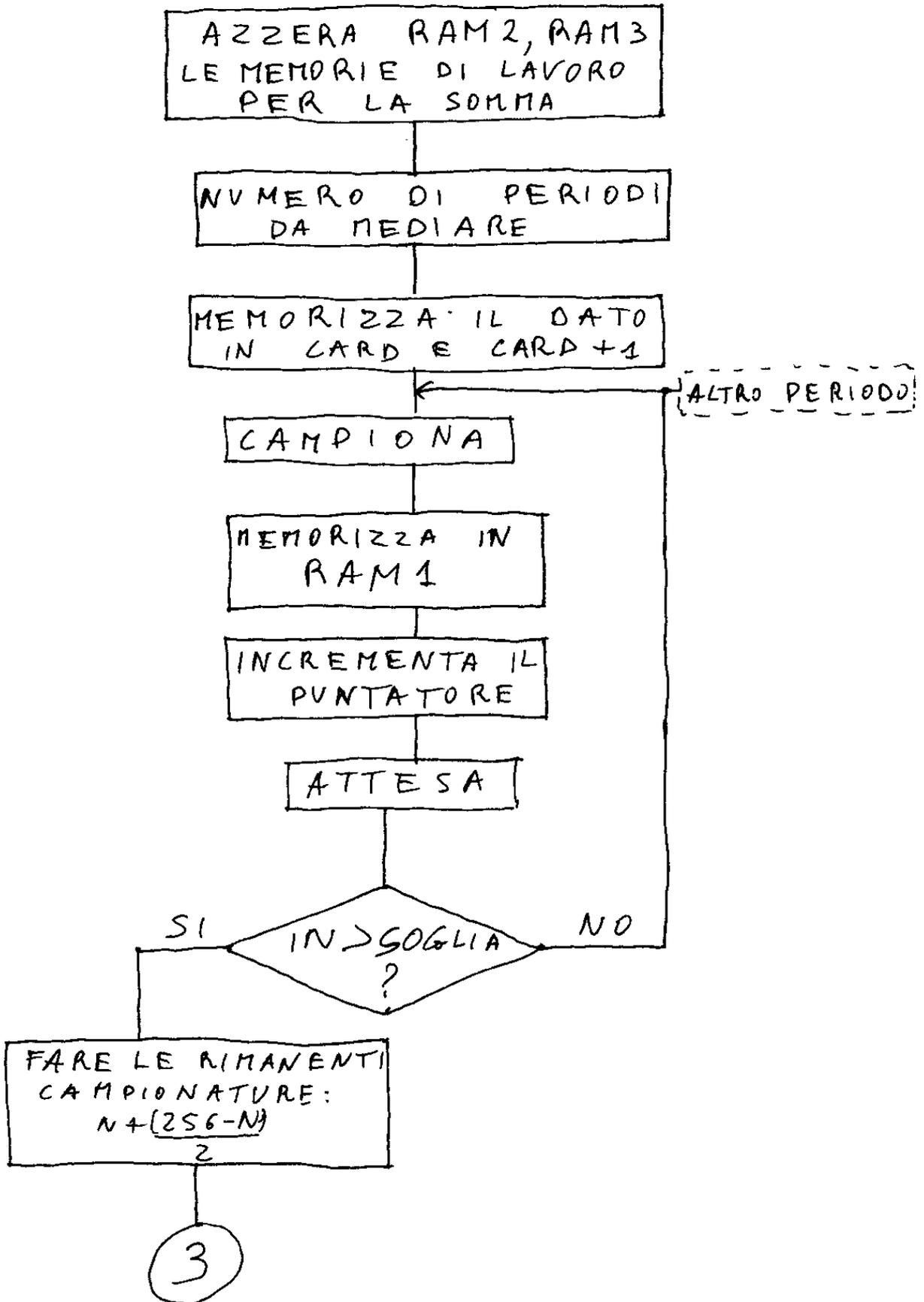
PARTE SECONDA

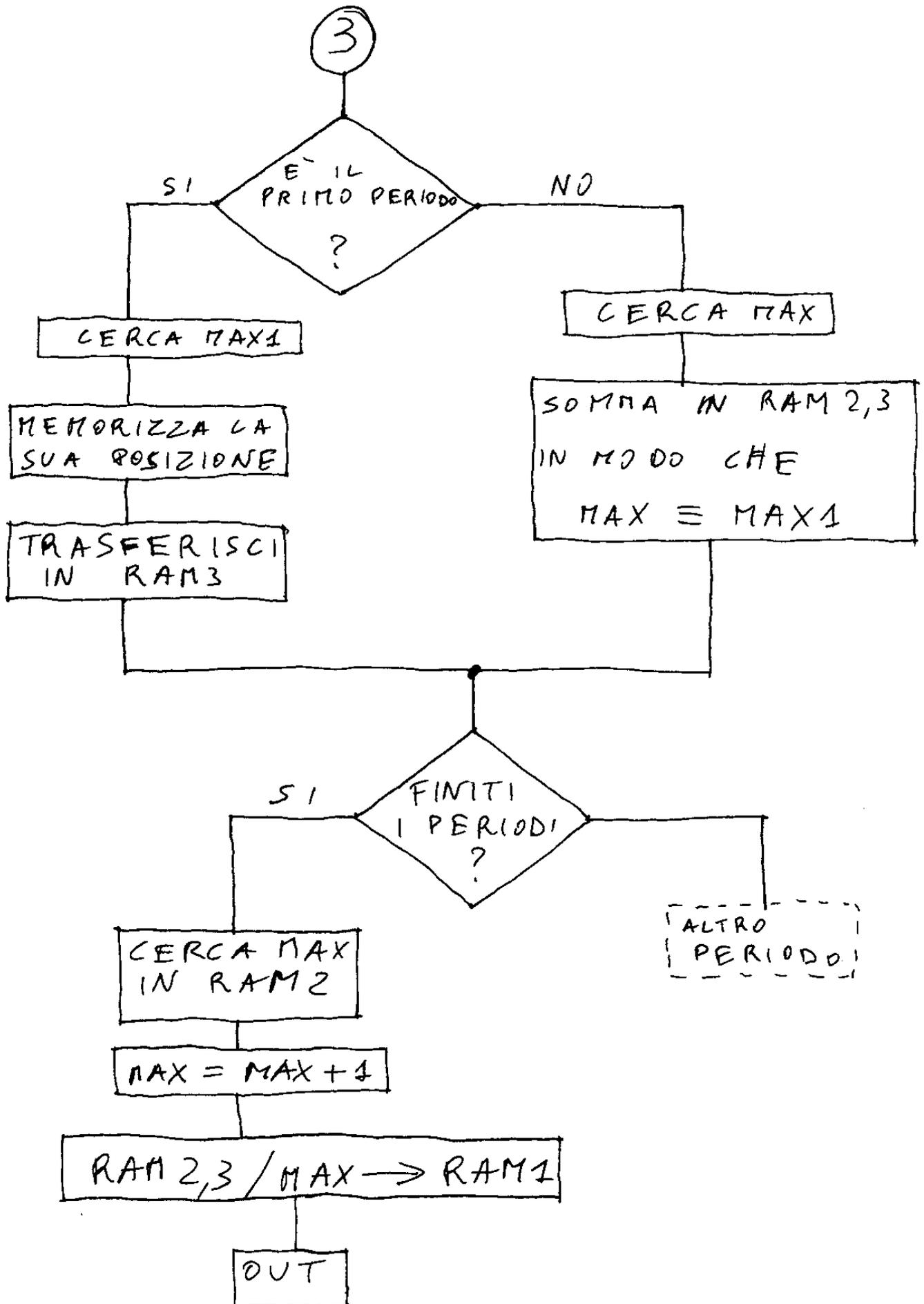






PARTE TERZA: MEDIA





;PROGRAMMA MEDIA. GIRA SUL KRAMER KIT
;DATO UN SEGNALE DI CIRCA UN HERTZ
;NON ESATTAMENTE PERIODICO FA LA MEDIA
;SU UN NUMERO QUALUNQUE DI PERIODI DI
;UNA CERTA PARTE DEL SEGNALE
;
;PARTE PRIMA : DETERMINAZIONE DELLA SOGLIA
;OLTRE LA QUALE IL SEGNALE E' CONSIDERATO
;SIGNIFICATIVO

0400	ORG	0400H		
2400	STACK	EQU	2400H	
0098	WORD1	EQU	98H	
0089	WORD2	EQU	89H	;PAROLE DI CONTROLLO 8255
3C03	CNTR1	EQU	3C03H	
3C07	CNTR2	EQU	3C07H	
3C0B	CNTR3	EQU	3C0BH	;PORTE CONTROLLO 8255
3C08	DAC1	EQU	3C08H	
3C09	DAC2	EQU	3C09H	;INDIRIZZO CONVERTITORI D/A
3C0A	COMP	EQU	3C0AH	;INDIRIZZO COMPARATORE (INPT)
3C01	MSD	EQU	3C01H	
3C05	LSD	EQU	3C05H	;INDIRIZZO DISPLAY
3C00	SWCR	EQU	3C00H	
3C04	SWCL	EQU	3C04H	;INDIRIZZO INTERRUITORI
2000	RAM1	EQU	2000H	;ZONA DI CARICAMENTO
2100	RAM2	EQU	2100H	;ZONA DEI BYTES PIU' SIGNIFICATI
2200	RAM3	EQU	2200H	;ZONA DEI BYTES MENO SIGNIFICATI
0080	TARAT	EQU	80H	;TARATURA: -1,25 VOLT
				;DECODIFICHE PER DISPLAY SETTE SEGMENTI
0065	INTER	EQU	65H	; ?
005B	ESSE	EQU	5BH	; S
0015	ENNE	EQU	15H	; N
0067	PI	EQU	67H	; P
0047	EFFE	EQU	47H	; F
007E	ZERO	EQU	7EH	; 0
2300	LEVEL	EQU	2300H	;INDIRIZZO DELLA SOGLIA
2301	SOVRA	EQU	2301H	; = DEL NUMERO DI CAMPIONA
2302	ATTES	EQU	2302H	; = DEL 'TEMPO' DI ATTESA
2303	CARD	EQU	2303H	; = DEL NUMERO DI PERIODI
				;DA CAMPIONARE
2305	MMASS	EQU	2305H	;INDIRIZZO DELLA POSIZIONE DEL
				;MASSIMO DEL PRIMO PERIODO
0400	310024	LXI	SP,STACK	
0403	3E98	MVI	A,WORD1	

```
0405 32033C      STA      CNTR1
0408 32073C      STA      CNTR2
040B 3E89        MVI      A,WORD2
040D 320B3C      STA      CNTR3
0410 3E80        MVI      A,TARAT
0412 32093C      STA      DAC2
0415 3E65        MVI      A,INTER ;DISPLAY: S ?
0417 32053C      STA      LSD      ;
041A 3E5B        MVI      A,ESSE   ;
041C 32013C      STA      MSD      ;
041F CD4104      CALL     STRB     ;CARICA IL VALORE SOGLIA
0422 B7          ORA      A      ;TEST SE=0
0423 CA8B04      JZ       SCND    ;SOGLIA=0 SIGNIFICA PASSARE
                        ;ALLA SECONDA PARTE CON IL
                        ;VALORE DI SOGLIA PRECEDEMENT
                        ;MEMORIZZATO
```

```
0426 320023      STA      LEVEL   ;SE O.K. LO MEMORIZZA
0429 CD6604      INPUT:  CALL     CAMP    ;CAMPIONA L'INPUT
042C 47          MOV      B,A
042D 3A0023      LDA      LEVEL
0430 B8          CMP      B      ;CONFR. INPUT CON SOGLIA
0431 DA3A04      JC       FUORI   ;SE INPUT<SOGLIA
0434 32093C      STA      DAC2    ;ALTRIMENTI FUORI INPUT
0437 C32904      JMP      INPUT
043A 78          FUORI:  MOV      A,B
043B 32093C      STA      DAC2    ;FUORI SOGLIA
043E C32904      JMP      INPUT
```

;CARICA I DATI DAGLI INTERRUATTORI IN A

```
0441 3A003C      STRB:  LDA      SWCR
0444 E602        ANI      02H
0446 47          MOV      B,A
0447 D5          PUSH     D
0448 CD5904      CALL     WAIT1
044B D1          POP      D
044C 3A003C      LDA      SWCR
044F E602        ANI      02H
0451 B8          CMP      B
0452 CA4104      JZ       STRB
0455 3A043C      LDA      SWCL
0458 C9          RET
```

;ATTESA DI 25 MILLISEC. ANTIRIMBALZO

```
0459 1619      WAIT1:  MVI      D,25
045B 1E64      RIT1:   MVI      E,100
045D 1D          RIT2:   DCR      E
```

```
045E C25D04      JNZ      RIT2
0461 15          DCR      D
0462 C25B04      JNZ      RIT1
0465 C9          RET
```

```
;CONVERSIONE ANALOGICO---->DIGITALE
;DATO DI PROVA IN C
;COSTANTE IN B
```

```
0466 C5          CAMP:   PUSH    B
0467 0680        MVI     B,80H   ;SET BIT 7
0469 0E00        MVI     C,00H   ;TOTALIZZATORE
046B 78          MOV     A,B
046C B1          LOOP:   ORA     C           ;B+C-->A
046D 32083C      STA     DAC1    ;OUT DI PROVA
0470 4F          MOV     C,A
0471 00          NOP
0472 00          NOP
0473 00          NOP
0474 00          NOP
0475 00          NOP
0476 00          NOP      ;ATTESA PER DAC E CO.
0477 3A0A3C      LDA     COMP
047A E601        ANI     01
047C CA8204      JZ      PROX
047F 78          MOV     A,B
0480 A9          XRA     C           ;RESET BIT 'N'
0481 4F          MOV     C,A
0482 78          PROX:   MOV     A,B           ;
0483 0F          RRC     ;N=N-1
0484 47          MOV     B,A           ;
0485 D26C04      JNC     LOOP    ;SE N=0 RET
0488 79          MOV     A,C
0489 C1          POP     B
048A C9          RET
```

```
;PARTE SECONDA: AGGIUSTAMENTO TEMPO D'ATTESA
;B : BIT N SETTATO
;C : 'TEMPO' D'ATTESA
;D : CONTATORE DI CAMPIONATURE
;E : NUMERO DI CAMPIONATURE
;H : LIVELLO DI SOGLIA
```

```
048B 3E15        SCND:   MVI     A,ENNE ;DISPLAY : N?
048D 32013C      STA     MSD      ;
0490 3E65        MVI     A,INTER  ;
0492 32053C      STA     LSD      ;
0495 CD4104      CALL    STRB    ;NUM. DI CAMPIONATURE
```

```
0498 FEE4          CPI      0E4H
049A D22005       JNC      OVERF    ;N SEMPRE < 228
049D 320123       STA      SOVRA    ;
04A0 5F           MOV      E,A      ;
04A1 3A0023       LDA      LEVEL    ;SOGLIA
04A4 67           MOV      H,A      ;
04A5 0680         MVI      B,80H    ;'TEMPO' DI PROVA
04A7 0E00         MVI      C,00H    ;
04A9 78           MOV      A,B
04AA 1600         RIPR:   MVI      D,00    ;CONTATORE =0
04AC B1           ORA      C      ;SET BIT N (B+C-->C)
04AD 4F           MOV      C,A
04AE 0C           INR      C
04AF CA2505       JZ       FULL     ;C=FFH?
04B2 0D           DCR      C
04B3 CDEE04       CALL    ICE      ;ATTENDE L'IMPULSO
04B6 CD6604       INGR:   CALL    CAMP
04B9 BC           CMP      H      ;INPUT>SOGLIA?
04BA D2C804       JNC      AUM     ;SE INPUT>SOGLIA
04BD AF           XRA      A
04BE B2           ORA      D      ;D=0?
04BF CAB604       JZ       INGR
04C2 78           MOV      A,B     ;ALTRIMENTI
04C3 A9           XRA      C      ;DIMINUISCE IL TEMPO DI
04C4 4F           MOV      C,A     ;ATTESA
04C5 C3CE04       JMP      ALTRA
04C8 14           AUM:    INR      D      ;CONTATORE+1
04C9 7A           MOV      A,D
04CA BB           CMP      E      ;CONTATORE < N ?
04CB DAE604       ALTRA:  MOV      A,C
04CE 79           RRC
04CF 0F           STA      LSD
04D0 32053C       MOV      A,B     ;BIT N-1 SETTATO
04D3 78           RRC
04D4 0F           STA      MSD
04D5 32013C       MOV      B,A
04D8 47           JNC      RIPR
04D9 D2AA04       XRA      A
04DC AF           ORA      C
04DD B1           JNZ     MEDIA    ;TERZA PARTE
04DE C22A05       MVI      A,ZERO
04E1 3E7E         JMP      LAMP
04E3 C30705       PROSS:  PUSH     B
04E6 C5           CALL    WAIT
04E7 CDFA04       POP      B
04EA C1           JMP      INGR
04EB C3B604
```

;ROUTINE ICE ATTENDE CHE

;FINISCA L'IMPULSO

04EE 3A0023	ICE:	LDA	LEVEL
04F1 67		MOV	H,A
04F2 CD6604		CALL	CAMP
04F5 BC		CMP	H
04F6 D2EE04		JNC	ICE
04F9 C9		RET	

;ATTESA TRA UNA CAMPIONATURA E L'ALTRA
;RICHIEDE CHE C CONTENGA GIA' IL GIUSTO VALORE

04FA C5	WAIT:	PUSH	B
04FB 060F		MVI	B,0FH
04FD 05	RIT:	DCR	B
04FE C2FD04		JNZ	RIT
0501 0D		DCR	C
0502 C2FB04		JNZ	WAIT+1
0505 C1		POP	B
0506 C9		RET	

;LAMPEGGIA SUI DISPLAY IL VALORE DI A

0507 47	LAMP:	MOV	B,A
0508 32053C		STA	LSD
050B 32013C		STA	MSD
050E CD5206		CALL	WAIT2
0511 3E00		MVI	A,00
0513 32053C		STA	LSD
0516 32013C		STA	MSD
0519 CD5206		CALL	WAIT2
051C 78		MOV	A,B
051D C30805		JMP	LAMP+1

;DISPLAY: 88 LAMPEGGIANTE

0520 3E7F	OVERF:	MVI	A,7FH
0522 C30705		JMP	LAMP

;DISPLAY: FF LAMPEGGIANTE

0525 3E47	FULL:	MVI	A,EFFE
0527 C30705		JMP	LAMP

;PARTE TERZA : MEDIA

;DATI GIA' ACQUISITI:
;LEVEL : SOGLIA
;ATTES : TEMPO DI ATTESA ---> C
;SOVRA : NUM. DI CAMPIONATURE
;DATO DA DEFINIRE DALL'UTENTE:
;CARD E CARD+1 : NUM. DI PERIODI DA
;REGISTRARE

```
052A 79      MEDIA:  MOV    A,C
052B 320223  STA    ATTES ;TEMPO TROVATO IN RAM
052E 3E67    MVI    A,PI  ;DISPLAY : P ?
0530 32013C  STA    MSD   ;
0533 3E65    MVI    A,INTER ;
0535 32053C  STA    LSD   ;
0538 CD4104  CALL   STRB  ;CARICA IL NUM DI PERIODI
053B 320323  STA    CARD  ;DA CAMPIONARE
053E 320423  STA    CARD+1 ;
0541 210021  LXI    H,RAM2 ;AZZERA RAM2
0544 AF      XRA    A     ;
0545 77      AZZ:   MOV    M,A   ;
0546 2C      INR    L     ;
0547 C24505  JNZ    AZZ   ;
054A 3A0123  LDA    SOVRA ;CONTATORE=N+(256-N)/2
054D 1F      RAR    ;
054E F680    ORI    80H  ;
0550 57      MOV    D,A   ;
0551 CDEE04  CALL   ICE   ;
0554 210020  NOVA:  LXI    H,RAM1
0557 CD6604  INIZ:  CALL   CAMP
055A 77      MOV    M,A   ;NEL BUFFER ROTANTE
055B 47      MOV    B,A   ;
055C 2C      INR    L     ;
055D 3A0223  LDA    ATTES ;ATTESA
0560 4F      MOV    C,A   ;
0561 CDFA04  CALL   WAIT  ;
0564 3A0023  LDA    LEVEL ;INPUT > SOGLIA ?
0567 8B      CMP    B     ;
0568 D25705  JNC    INIZ
056B CD6604  VALID: CALL   CAMP
056E 77      MOV    M,A   ;
056F 2C      INR    L     ;
0570 3A0223  LDA    ATTES ;
0573 4F      MOV    C,A   ;
0574 CDFA04  CALL   WAIT  ;
0577 15      DCR    D     ;CONTATORE
0578 C26B05  JNZ    VALID ;
057B 3A0423  LDA    CARD+1 ;PRIMO PERIODO?
057E 5F      MOV    E,A   ;
057F 3A0323  LDA    CARD  ;
0582 BB      CMP    E     ;
```

```
0583 C2A205      JNZ      NEXT
0586 210020      LXI      H,RAM1
0589 CDD405      CALL     MAXF
058C 320523      STA      MMASS
058F CDE805      CALL     TRASF
0592 3A0323      UNION:  LDA      CARD
0595 3D          DCR      A
0596 32053C      STA      LSD
0599 320323      STA      CARD      ;TRASCORSO UN PERIODO
059C C25405      JNZ      NOVA      ;SE CARD NON = 0 ALTRO PERIODO
059F C3B105      JMP      NORM
05A2 210020      NEXT:   LXI      H,RAM1
05A5 CDD405      CALL     MAXF
05AB 320623      STA      MMASS+1
05AB CDF605      CALL     SOMM
05AE C39205      JMP      UNION
05B1 3A0423      NORM:   LDA      CARD+1 ;NUM. DI PERIODI
05B4 5F          MOV      E,A
05B5 CD1A06      CALL     DDIVI
05B8 3E65        MVI      A,INTER ;DISPLAY : ??
05BA 32053C      STA      LSD
05BD 32013C      STA      MSD
05C0 CD4104      CALL     STAB      ;CARICARE 00 SE ESCE SU
                                ;SCRIVENTE ALTRO NUMERO
                                ;SE SU OSCILLOSCOPIO

05C3 4F          MOV      C,A
05C4 210020      LXI      H,RAM1
05C7 7E          SIGHT:  MOV      A,M
05C8 32093C      STA      DAC2
05CB AF          XRA      A
05CC B9          CMP      C      ;SCRIVENTE?
05CD CC5206      CZ       WAIT2
05D0 2C          INR      L
05D1 C3C705      JMP      SIGHT

;CERCA IL MASSIMO FRA 256 BYTES INDIRIZZATI
;DA H,L . NE MEMORIZZA LA POSIZIONE

05D4 D5          MAXF:   PUSH     D
05D5 5E          MOV      E,M
05D6 55          MOV      D,L
05D7 2C          MAX1:   INR      L
05D8 CAE505      JZ       EXIT
05DB 7E          MOV      A,M
05DC BB          CMP      E
05DD DAD705      JC       MAX1
05E0 5F          MOV      E,A
05E1 55          MOV      D,L
05E2 C3D705      JMP      MAX1
```

```
05E5 7A      EXIT:  MOV    A,D
05E6 D1              POP    D
05E7 C9              RET
```

;TRASFERISCE RAM1 IN RAM3

```
05E8 210020   TRASF:  LXI    H,RAM1
05EB 7E       BANK:  MOV    A,M
05EC 2622              MVI    H,22H    ;RAM3
05EE 77              MOV    M,A
05EF 2C              INR    L
05F0 2620              MVI    H,20H    ;RAM1
05F2 C2EB05              JNZ    BANK
05F5 C9              RET
```

;SOMMA RAM1 IN RAM2,3
;IN MODO CHE LA POSIZIONE DELL'ATTUALE
;MASSIMO COINCIDA CON QUELLA DEL PRECEDENTE
;D,E : RAM1 H,L : RAM2,3

```
05F6 D5       SOMM:  PUSH  D
05F7 C5              PUSH  B
05F8 110020     LXI    D,RAM1
05FB 3A0623     LDA    M+1
05FE 5F         MOV    E,A
05FF 210022     LXI    H,RAM3
0602 3A0523     LDA    M
0605 6F         MOV    L,A
0606 0600       MVI    B,00    ;CONTA FINO A 256
0608 1A       SOMM1: LDAX  D    ;RAM1---->A
0609 86         ADD    M
060A 77         MOV    M,A
060B D21106     JNC    PROXS
060E 25         DCR    H    ;RIPORTO IN RAM2
060F 34         INR    M
0610 24         INR    H
0611 2C       PROXS: INR    L
0612 1C         INR    E
0613 04         INR    B
0614 C20806     JNZ    SOMM1
0617 C1         POP    B
0618 D1         POP    D
0619 C9         RET
```

;DIVISIONE
;H,L : DIVIDENDO
;E : DIVISORE
;D : RISULTATO

```
061A 010021 DDIVI: LXI B, RAM2
061D 0A PREPD: LDAX B ;BYTE PIU' SIGNIFICATIVO
061E 67 MOV H, A
061F 04 INR B
0620 0A LDAX B ;BYTE MENO SIGNIFICATIVO
0621 6F MOV L, A
0622 C5 PUSH B
0623 CD3406 CALL DIVID
0626 7A MOV A, D
0627 C1 POP B
0628 05 DCR B
0629 05 DCR B ;PUNTA SU RAM1
062A 02 STAX B
062B 0C INR C
062C CA3306 JZ FINE
062F 04 INR B ;PUNTA SU RAM2
0630 C31D06 JMP PREPD ;ALTRA DIVISIONE
0633 C9 FINE: RET
```

```
;H, L : DIVIDENDO
;E : DIVISORE
;D : RISULTATO
;B, C : DATI PROVVISORI
```

```
0634 010000 DIVID: LXI B, 00
0637 1600 MVI D, 00
0639 7B SCATT: MOV A, E
063A 81 ADD C
063B 4F MOV C, A
063C D24006 JNC CONFR
063F 04 INR B
0640 7C CONFR: MOV A, H
0641 B8 CMP B
0642 DA5106 JC OK
0645 CA4C06 JZ SECND
0648 14 ALTRO: INR D
0649 C33906 JMP SCATT
064C 7D SECND: MOV A, L
064D B9 CMP C
064E D24806 JNC ALTRO
0651 C9 OK: RET
```

```
;0.4 SECONDI DI ATTESA
```

```
0652 16FF WAIT2: MVI D, 0FFH
0654 1EFF RIT3: MVI E, 0FFH
0656 1D RIT4: DCR E
```

0657 C25606	JNZ	RIT4
065A 15	DCR	D
065B C25406	JNZ	RIT3
065E C9	RET	
0000	END	

ALTRA 04CE	ALTRO 0648	ATTES 2302	AUM 04C8
AZZ 0545	BANK 05EB	CAMP 0466	CARD 2303
CNTR1 3C03	CNTR2 3C07	CNTR3 3C08	COMP 3C0A
CONFR 0640	DAC1 3C08	DAC2 3C09	DDIVI 061A
DIVID 0634	EFPE 0047	ENNE 0015	ESSE 005B
EXIT 05E5	FINE 0633	FULL 0525	FUORI 043A
ICE 04EE	INGR 04B6	INIZ 0557	INPUT 0429
INTER 0065	LAMP 0507	LEVEL 2300	LOOP 046C
LSD 3C05	MAX1 05D7	MAXF 05D4	MEDIA 052A
MMASS 2305	MSD 3C01	NEXT 05A2	NORM 05B1
NOVA 0554	OK 0651	OVERF 0520	PI 0067
PREPD 061D	PROSS 04E6	PROX 04B2	PROXS 0611
RAM1 2000	RAM2 2100	RAM3 2200	RIPR 04AA
RIT 04FD	RIT1 045B	RIT2 045D	RIT3 0654
RIT4 0656	SCATT 0639	SCND 048B	SECND 064C
SIGHT 05C7	SOMM 05F6	SOMM1 0608	SOVRA 2301
STACK 2400	STRB 0441	SWCL 3C04	SWCR 3C00
TARAT 0080	TRASF 05E8	UNION 0592	VALID 056B
WAIT 04FA	WAIT1 0459	WAIT2 0652	WORD1 0098
WORD2 0089	ZERO 007E		

CONCLUSIONE

Le tre applicazioni che costituiscono il nostro lavoro sono relativamente semplici ma fortemente indicative dell'uso generale e della flessibilità dei μ P.

Queste applicazioni ne suggeriscono immediatamente altre: si consideri il caso delle rilevazioni sismiche continue. Gli apparecchi che registrano su carta ne consumano tanta per registrare soltanto qualche evento ogni tanto. Un μ P potrebbe pilotare l'accensione dell'apparecchio solo per registrare un evento appena trascorso che il μ P stesso ha già memorizzato. Potrebbe inoltre tenere conto della distanza di tempo tra un evento e l'altro e pilotare una scrivente alfanumerica che elenchi questi intervalli, le intensità e le durate degli eventi. Oppure potrebbe selezionare solo gli eventi compresi entro una certa banda di intensità.

Problemi simili si possono presentare anche nello studio di eventi nucleari rari o dei raggi cosmici.

B I B L I O G R A F I A

- MICROCOMPUTER SYSTEM AND SOFTWARE DATA BOOK

Ed. INTEL 1977

- MICROCOMPUTER AND MICROPROCESSOR COMPONENTS DATA BOOK

Ed. INTEL 1977

- CALCUL DIFFERENTIEL ET INTEGRAL (Tomo II)

N: PISKOUNOV

Ed. MIR

- INTEGRATED ELECTRONICS

MILLMAN-HALKIAS

Ed. McGRAW-HILL