



MINISTERO DELL'INDUSTRIA DEL COMMERCIO E DELL'ARTIGIANATO
D.G.P.I. - UFFICIO ITALIANO BREVETTI E MARCHI

BREVETTO PER INVENZIONE INDUSTRIALE

N. 01245221

Il presente brevetto viene concesso per l'invenzione oggetto della domanda sotto specificata:

<i>num. domanda</i>	<i>anno</i>	<i>U.P.I.C.A.</i>	<i>data pres. domanda</i>	<i>classifica</i>
000669	91	MILANO	13/03/1991	G-06F

TITOLARE AMSTRAD PUBLIC LIMITED COMPANY
A BRENTWOOD, ESSEX (GRAN BRETAGNA)

RAPPR. TE FAGGIONI GIOVANMARIA

INDIRIZZO FUMERO BREVETTI SNC
VIA SANTA AGNESE 12
20100 MILANO

TITOLO SISTEMA DI ELABORAZIONE

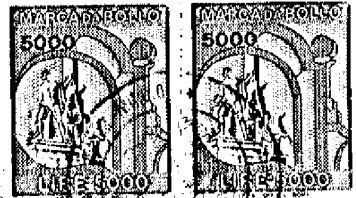
INVENTORE GANE STEPHEN DAVID

PRIORITA' GRAN BRETAGNA
DOMANDA BREV. INVENZIONE INDUSTRIALE
16 MARZO 1990 N. 90 05916.3

Roma, 13 SETTEMBRE 1994

IL DIRIGENTE
(GIOVANNA MORELLI)

AL MINISTERO DELL'INDUSTRIA DEL COMMERCIO E DELL'ARTIGIANATO
UFFICIO CENTRALE BREVETTI - ROMA
DOMANDA DI BREVETTO PER INVENZIONE INDUSTRIALE



A. RICHIEDENTE (1)

1) Denominazione **AMSTRAD PUBLIC LIMITED COMPANY**
 Residenza **Brentwood, Essex (GB)**
 2) Denominazione
 Residenza

codice
 codice

B. RAPPRESENTANTE DEL RICHIEDENTE PRESSO L'U.C.B.

cognome nome **FAGGIONI Giovanmaria e altri** | cod. fiscale
 denominazione studio di appartenenza **FUMERO STUDIO CONSULENZA BREVETTI S.n.c.**
 via **S. Agnese** | n. **12** | città **Milano** | cap **20123** (prov) **MI**

C. DOMICILIO EFFETTIVO DESTINATARIO **Vedi sopra**

via | n. | città | cap (prov)

D. TITOLO

classe proposta (ser/cl/sci)

"SISTEMA DI ELABORAZIONE"

E. INVENTORI DESIGNATI

cognome nome

cognome nome

1) **GANE Stephen David** | 3)
 2) | 4)

F. PRIORITÀ

nazione o organizzazione	tipo di priorità	numero di domanda	data di deposito	allegato S/R
1) <input type="checkbox"/> GRAN BRETAGNA	<input type="checkbox"/> D. INV.	<input type="checkbox"/> 90 05916.3	<input type="checkbox"/> 16/03/1990	<input type="checkbox"/> R
2) <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

G. CENTRO ABILITATO DI RACCOLTA COLTURE DI MICRORGANISMI, denominazione

H. ANNOTAZIONI SPECIALI

DOCUMENTAZIONI ALLEGATE

N. es.

Doc. 1) [PROV] n. pag **14** | riassunto con disegno principale, descrizione e rivendicazioni (obbligatori 2 esemplari)
 Doc. 2) [XXX] n. tav. **05** | disegno (obbligatorio se citato in descrizione, 2 esemplari)
 Doc. 3) [XXX] | lettera d'incarico, ~~XXXXXXXXXXXXXXXXXXXX~~
 Doc. 4) [RIS] | designazione inventore
 Doc. 5) [XXX] | documenti di priorità con traduzione in italiano
 Doc. 6) [RIS] | autorizzazione o atto di cessione
 Doc. 7) [RIS] | nominativo completo del richiedente

B) attestati di versamento, totale lire **TRECENTOSETTANTADUEMILA** | obbligatorio
 B) marche da bollo per attestato di brevetto di lire **DIECIMILA** | obbligatorio

COMPILATO IL **13/03/1991** | FIRMA DEL (1) RICHIEDENTE (1)

CONTINUA SI/NO **NO**

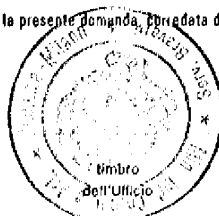
DEL PRESENTE ATTO SI RICHIEDE COPIA AUTENTICA SI/NO **SI**

UFFICIO PROVINCIALE IND. COMM. ART. DI **MILANO** | codice **15**
 VERBALE DI DEPOSITO | NUMERO DI DOMANDA **MI 91 A 000669** | Reg A
 L'anno milionovecento **Novantuno** | il giorno **TREDICI** | del mese di **MARZO**

Il (i) richiedente(i) sopraindicato(i) ha(hanno) presentato a me sottoscritto la presente domanda, corredata di n. **00** | fogli aggiuntivi per la concessione del brevetto soprariportato.

I. ANNOTAZIONI VARIE DELL'UFFICIALE ROGANTE

IL DEPOSITANTE
Alfonso



L'UFFICIALE ROGANTE
Poldighi A.

RIASSUNTO INVENZIONE CON DISEGNO PRINCIPALE, DESCRIZIONE E RIVENDICAZIONE

NUMERO DOMANDA [MI 91 A - 000669] REG. A

DATA DI DEPOSITO [/ /]

NUMERO BREVETTO []

DATA DI RILASCIO [/ /]

D. TITOLO

["SISTEMA DI ELABORAZIONE"]

L. RIASSUNTO

Un sistema di elaborazione è previsto per uso con una sorgente di programma asportabile ed inseribile, quale una cartuccia ROM (9) o disco flessibile (12), e comprende una CPU (1) per eseguire istruzioni di programma immagazzinate, direttamente dalla cartuccia ROM odopo essere state lette in una memoria RAM (3). Un chip di generazione video/audio (2) fornisce operazioni di base aperte a tutti i programmi, e funzioni perfezionate aperte solo a programmi selezionati. L'accesso alle funzioni perfezionate viene abilitato comparando una sequenza di byte emessa dalla CPU, in risposta alle istruzioni del programma, con una sequenza di riferimento fornita da un generatore di sequenza pseudocasuale (fig. 2), ed abilitando le funzioni perfezionate dipendentemente dal risultato della comparazione.

M. DISEGNO



Descrizione dell'invenzione avente per titolo:

"SISTEMA DI ELABORAZIONE"

A nome: AMSTRAD PUBLIC LIMITED COMPANY a Brentwood, Essex (Gran Bretagna)

Inventore: Stephen David GANE

Depositata il: 13 MAR. 1991

MI 91 A/00669

* * * * *

L'invenzione riguarda un sistema di elaborazione ed in particolare un elaboratore che riceve il suo programma di software da una sorgente di programma asportabile, quale cartuccia ROM, disco flessibile, o nastro in cassetta.

E' pratica comune che un elaboratore sia progettato e fabbricato da una società specialista in hardware, e che il software per tale elaboratore sia scritto e venduto da molte diverse società specialiste in software. Tale software può coprire molte diverse applicazioni, può essere fornito da numerosi venditori concorrenti, e può essere disponibile su supporti di vario tipo, fra cui nastro in cassetta, disco flessibile e cartuccia ROM. In tali condizioni il fornitore di hardware, che produce l'elaboratore, può non essere in grado di regolare o controllare la qualità o la conformità di tale software. Egli farà tuttavia ogni sforzo per garantire che qualunque variazione nella progettazione o nella fabbricazione dell'elaboratore non lo rendano incompatibile con software scritto per generazioni precedenti dello stesso elaboratore.

E' spesso possibile per il produttore conseguire desiderabili riduzioni nel costo di produzione o miglioramenti nell'affidabilità del prodotto incorporando gli ultimi progressi della tecnologia, pur senza pregiudicare la compatibilità con generazioni precedenti di software. Tali cambiamenti, tuttavia, non possono riguardare le funzioni dell'elaboratore, quali il numero di modi di visualizzazione o le possibilità di generazione di musica e suono.

Il produttore di elaboratori può tuttavia desiderare di potenziare il suo elaboratore aggiungendo funzioni migliorate, quali modi di visualizzazione addizionali o nuove possibilità di generazione di suono e musica, pur mantenendo le funzioni disponibili precedentemente ed anche mantenendo compatibilità con generazioni precedenti di software.

In tal caso, il progettista di elaboratori deve essere consapevole che il suo elaboratore deve funzionare correttamente, innanzitutto, con software perfezionato che è stato scritto tenendo presenti le nuove funzioni perfezionate e specificamente destinato a sfruttarle e, secondariamente, con software vecchio che è stato scritto senza la conoscenza delle nuove funzioni perfezionate. In particolare, il progettista di elaboratori deve garantire che software vecchio non attivi accidentalmente le nuove funzioni. L'elaboratore richiede un metodo per assicurare che le nuove funzioni perfezionate siano disponibili solo per il software nuovo, che è stato scritto con piena comprensione dell'uso cor-

retto di tali nuove funzioni.

In passato i produttori di elaboratori hanno pubblicato talvolta specifiche per i loro prodotti, indicanti che determinate aree della mappa di ingresso/uscita e di memoria sono riservate per espansione o perfezionamento futuro. Vi sono tuttavia casi in cui non si è provveduto a ciò.

Così pure, in passato, alcuni produttori di elaboratori desiderosi di perfezionare i loro prodotti hanno cercato di controllare la qualità o conformità di software prodotto da terzi pubblicando, ad esempio, liste di software adatto. Un tale metodo è applicabile solo quando il numero di titoli di software da qualificare è noto e sufficientemente piccolo.

Così pure, in passato, alcuni produttori di elaboratori hanno insistito perché il software da usare con le nuove funzioni perfezionate fosse provvisto di una chiave di hardware o chiave elettronica, nota come "dongle". Un tale metodo richiede la spesa di qualificare innanzitutto il software a verificare la sua attitudine all'uso col nuovo elaboratore e, secondariamente, la spesa di installare una chiave di hardware o "dongle" in ogni articolo di software venduto.

Nel brevetto US-4 462 076 si propone un sistema che autentica una capsula asportabile, richiedendo alla CPU (unità centrale di elaborazione) di leggere e verificare il contenuto di posizioni scelte nella cartuccia. Un metodo simile è usato nel sistema di elaborazione Commodore 64, vedi Commodore 64 Programmer's Refe-

rence guide, 1982, Commodore Business Machines Inc., pag. 263. Tali proposte richiedono che l'elaboratore abbia microprogrammi speciali (firmware, ad esempio software ROM) per consentirgli di cercare il codice di autenticazione.

Il brevetto europeo 67 875 ed il brevetto US-4 688 169 descrivono entrambi dei sistemi nei quali un codice specifico di macchina o "firma" è scritto su ogni copia di un programma, così che esso funzioni solo su una macchina specifica identificata da quel codice o firma. Ancora è necessario che il microprocessore verifichi quella particolare posizione del nastro di programma.

In un altro esempio, i brevetti europei 206 704 e 217 688 descrivono una cartuccia ROM che contiene il suo proprio microprocessore, cosicchè l'uscita di tale microprocessore possa essere comparata con quella di un dispositivo corrispondente nell'elaboratore principale per autenticare la cartuccia. Tuttavia, avere un microprocessore nella cartuccia ROM è costoso e complesso.

Secondo l'invenzione si fornisce un sistema di elaborazione comprendente una sorgente di programma asportabile ed inseribile, mezzi di memoria per immagazzinare istruzioni di programma, una CPU per eseguire istruzioni di programma immagazzinate, mezzi di funzione contenenti funzioni alle quali si accede con istruzioni dalla CPU, e mezzi di abilitazione di accesso per permettere accesso ad almeno alcune funzioni scelte, i mezzi di abilitazione di accesso comprendendo mezzi per comparare una sequenza di byte emessi dalla CPU in risposta alle istruzioni di programma con una

sequenza di riferimento di byte, e per abilitare le caratteristiche scelte nei mezzi di funzione dipendentemente dal risultato della comparazione.

Nella presente invenzione il programma stesso fa emettere in uscita dalla CPU una sequenza di byte. Solo quando ciò avviene, e quando la sequenza così generata si accorda con la sequenza di riferimento, sono rese disponibili al programma le funzioni addizionali perfezionate.

L'invenzione fornisce anche una sorgente di programma, quale una cartuccia ROM, ecc., da usare in un sistema di elaborazione, ed un metodo per abilitare selettivamente funzioni in un sistema di elaborazione.

La presente invenzione ha il vantaggio di essere semplice da realizzare, richiedendo solo che il processore, sotto controllo da parte di software, scriva una sequenza predeterminata di valori in uno specifico punto di connessione, per rendere disponibili funzioni perfezionate. Non sono richiesti speciali microprogrammi (firmware) incorporati, come sarebbero necessari in elaboratori che cercano "codici di sicurezza" ad indirizzi specifici nella sorgente di programma, se richiesto. Non è richiesto alcun microprocessore separato nella sorgente di programma.

L'invenzione verrà ora descritta in maggiore dettaglio, a titolo di esempio, con riferimento ai disegni allegati, nei quali:

fig. 1 è uno schema a blocchi di un sistema di elaborazione che applica l'invenzione;

fig. 2 è uno schema a blocchi di circuito che illustra un insieme di circuiti di bloccaggio nel chip di generazione video/audio nel sistema di fig. 1;

fig. 3 è un diagramma di flusso che illustra le fasi in un tentativo riuscito di sbloccare le nuove funzioni perfezionate nell'elaboratore;

fig. 4 è un diagramma di flusso che illustra le fasi in un tentativo non riuscito di sbloccare le nuove funzioni; e

fig. 5 è un diagramma di flusso illustrante le fasi in un tentativo riuscito di bloccare le nuove caratteristiche.

La fig. 1 mostra un elaboratore avente una CPU (unità centrale di elaborazione), 1 che esegue istruzioni del software applicativo comunemente caricato. Questo software viene caricato tramite un bus 25 in RAM (memoria ad accesso casuale) 3 e può avere origine da una cartuccia ROM (memoria a sola lettura) 9, connessa sul bus CPU tramite un connettore di cartuccia ROM 8, o da un disco flessibile 12, il cui contenuto viene letto mediante un'unità 11 a dischi flessibili ed un'unità 10 di controllo di dischi flessibili; o un nastro in cassetta 14, il cui contenuto viene letto mediante un'unità 13 a cassette. Benché particolarmente adatta per uso con una cartuccia ROM, questa invenzione non è limitata ad un particolare metodo di caricamento del software. Quando il programma è su disco o nastro, esso viene trasferito in RAM. Questo modo di funzionamento è anche possibile quando il programma è una cartuccia ROM, benché si preferisca che la CPU

esegua istruzioni una ad una direttamente dalla cartuccia. Nella descrizione seguente si supporrà che il programma venga trasferito in RAM, ciò essendo applicabile a tutti i tre tipi di sorgente di programma.

Durante il normale corso di esecuzione di software, la CPU esegue istruzioni che leggono e scrivono il contenuto della memoria RAM 3; leggono lo stato di una tastiera 7 e leggono e scrivono registri e memoria interni in un chip 2 di generazione video/audio, che controlla le funzioni video e audio dell'elaboratore. Questi registri e memoria interni ne comprendono alcuni relativi al controllo delle funzioni vecchie o di base, e anche altri supplementari relativi alle funzioni nuove o perfezionate. Le funzioni di base sono aperte a tutti i programmi, ma le funzioni perfezionate sono aperte solo ai programmi scritti con tali funzioni perfezionate specificamente previste. Il chip 2 fornisce anche il percorso di comunicazione per la tastiera 7 e RAM 3 nel bus 25.

Pure compreso nel chip 2 di generazione video/audio è un circuito di blocco che impedisce che software vecchio attivi accidentalmente le nuove funzioni. Questo blocco ha due stati. Quando è bloccato, le nuove funzioni perfezionate non sono accessibili alla CPU; quando è sbloccato, tali funzioni sono disponibili.

La fig. 2 mostra i dettagli dell'insieme di circuiti di blocco internamente al chip di generazione video/audio. Esso

comprende un generatore di sequenza binaria pseudocasuale (PRBS) consistente di bistabili da 15 a 22 e porte OR esclusivo ad essi associate. I bistabili da 15 a 22 funzionano in pratica come due generatori PRBS, ognuno della larghezza di quattro bit, per dare un PRBS della larghezza complessiva di otto bit. La costruzione di un generatore PRBS è ben nota e comprende fundamentalmente un registro a scorrimento con diversi stati, quattro nel caso mostrato, le uscite di stadi scelti essendo combinate e reintrodotte come dati ingresso al primo stadio. Il generatore PRBS può assumere sedici stati diversi, fra cui gli stati tutti zero. Ciascuno stato ha un unico valore a 8 bit, che appare sul bus PRBS (7:0). I sedici stati sono indicati con PRBS (0) fino a PRBS (15). Come è ben noto con generatori PRBS, essendo n il numero di stadi nel registro, il generatore PRBS è in grado di generare una sequenza pseudocasuale di 0 ed 1. La sequenza si ripete con una lunghezza che, se le connessioni con le porte sono scelte correttamente, ha un massimo di $m=2n-1$.

Lo stato del blocco è determinato dal bistabile 24 con segnale di uscita ENF. Quando ENF è zero, le funzioni perfezionate sono disabilitate (bloccate). Quando ENF è uno, le funzioni perfezionate sono abilitate e disponibili alla CPU (sbloccate).

Il generatore PRBS fa avanzare il suo stato ogni volta che la CPU esegue un'istruzione di uscita al punto di connessione A, questo essendo un punto di connessione nel chip 2 video/audio, col quale il circuito di blocco è interfacciato al bus 25, cosicché

un'istruzione di uscita al punto di connessione A è un'istruzione di uscita al blocco. Una tale istruzione di uscita dà luogo ad un impulso sul segnale CP, che sincronizza gli otto bistabili da 15 a 22 costituenti il generatore PRBS.

Durante il bordo crescente (attivo) di questo impulso il bus di dati CPU appare sul bus DI (7:0) e viene comparato con lo stato PRBS (7:0) di PRBS in un comparatore 23 a 8 bit. Così, ad esempio, se PRBS (7:0) è uguale a 3F (esadecimale) e la CPU esegue una istruzione che esce con il valore di dato 3F al punto di connessione A, il generatore PRBS avanza di uno stato, con l'uscita 23 di comparatore indicante una vera comparazione.

Su ciascun bordo positivo dell'impulso di orologio (CP) il generatore PRBS:

- a) avanzerà per la sua sequenza, se i dati su DI (7:0) sono conformi allo stato PRBS (7:0) di PRBS, o
- b) posizionerà a tutti 1, cioè rilancerà la sequenza dall'inizio, se il comparatore indica che DI (7:0) non è conforme a PRBS (7:0).

Su ciascun bordo positivo di orologio (CP), il bistabile di uscita 24:

- a) andrà ad uno stato 0 (bloccato) se PRBS è in stato PRBS 14, o
- b) andrà ad uno stato 1 (sbloccato) se PRBS è in stato PRBS 15, o
- c) manterrà il suo stato precedente.

L'uscita di CPU dipende naturalmente dal programma caricato in essa. Questa programma è ricavato dalla sorgente di programma, e cioè ROM 9, unità a disco flessibile 12 o nastro in cassetta 14.

Con la comparazione sopra descritta l'elaboratore può così dire se il software sulla ROM, sul disco flessibile o sul nastro, è stato scritto con le funzioni perfezionate o no. Solo se è stato scritto tenendo conto delle funzioni perfezionate, tali funzioni vengono sbloccate e rese disponibili al software. Quindi, software scritto prima dell'avvento delle funzioni perfezionate non può chiamarle inavvertitamente alterandole.

Ogniqualevolta è necessario azionare il blocco ed è sconosciuto lo stato del generatore PRBS, esso deve essere sincronizzato scrivendo dapprima tutti 1 e quindi tutti 0. Ciò rimetterà PRBS allo stato iniziale.

Le figg. 3, 4 e 5 mostrano sequenze di valori di dati di uscita di CPU per attivare il blocco. Queste figure si spiegano da sé e non vengono perciò descritte qui in dettaglio. La fig. 3 è un diagramma di flusso che mostra le fasi comprese in un tentativo riuscito di sbloccare le funzioni perfezionate. La fig. 4 è un diagramma di flusso che mostra le fasi comprese in un tentativo non riuscito di bloccare o sbloccare le funzioni perfezionate. La fig. 5 è un diagramma di flusso che mostra le fasi comprese in un tentativo riuscito di bloccare le nuove funzioni. I blocchi rappresentati contengono valori esadecimali da scrivere nel punto di connessione A. Anche PRBS (0-15) rappresentano gli stati PRBS in ordine.

Si vede così che il sistema comprende due parti. La prima è un meccanismo di blocco hardware, che compara valori di dati

scritti dalla CPU nel punto di connessione A con lo stato di un generatore PRBS; e la seconda è una routine di software per inviare la corretta sequenza di valori di dati, PRBS (0-15) al punto di connessione A. Questa sequenza di valori è scelta deliberatamente in modo da essere così oscura che la probabilità che software vecchio attivi accidentalmente la chiave è così piccola da essere trascurabile. Questo software può essere implementato in forme diverse; ad esempio, esso può contenere una serie di istruzioni di uscita, ciascuna con un operando di dati associato corrispondente allo stato PRBS; in alternativa può comprendere un ciclo con una tabella di consultazione associata.

Il sistema descritto non richiede controllo dettagliato, da parte del produttore dell'elaboratore, del software da utilizzare su di esso, e non richiede nemmeno l'uso di un "dongle" o simili. L'invenzione può essere usata ove non siano state riservate aree di mappa di ingresso/uscita e di memoria.

L'invenzione non è limitata ad alcuna particolare tecnica di software o struttura di programma, ma comprende qualunque software che possa emettere la corretta sequenza di valori di dati. Inoltre, essa non è limitata ad alcun mezzo particolare per immagazzinare il programma e, in particolare, comprende cartuccia ROM, disco flessibile e cassetta.

RIVENDICAZIONI

1) Sistema di elaborazione comprendente: una sorgente di programma asportabile ed inseribile; mezzi di memoria per imma-

gazzinare istruzioni di programma; una CPU per eseguire istruzioni di programma immagazzinate; mezzi di funzioni contenenti funzioni a cui si accede con istruzioni dalla CPU; e mezzi di abilitazione di accesso per permettere l'accesso ad almeno alcune di tali funzioni opportunamente selezionate; i mezzi di abilitazione di accesso comprendendo mezzi per comparare una sequenza di bytes emessi dalla CPU in risposta alle istruzioni di programma, con una sequenza di riferimento di bytes, e per abilitare le funzioni selezionate nei mezzi di funzioni dipendentemente dal risultato della comparazione.

2) Sistema di elaborazione come nella rivendicazione 1), in cui la sorgente di programma comprende una cartuccia ROM.

3) Sistema di elaborazione come nella rivendicazione 2), in cui i mezzi di memoria sono almeno in parte composti dalla cartuccia ROM, e la CPU esegue direttamente istruzioni contenute nella cartuccia ROM.

4) Sistema di elaborazione come nella rivendicazione 1), in cui la sorgente di programma comprende un disco flessibile.

5) Sistema di elaborazione come nella rivendicazione 1), in cui la sorgente di programma comprende un nastro in cassetta.

6) Sistema di elaborazione come nella rivendicazione 2), 4) o 5), in cui i mezzi di memoria comprendono una memoria accessibile alla CPU e non nella sorgente di programma asportabile, e comprendente inoltre mezzi di caricamento per caricare un programma nella sorgente di programma, nella memoria per l'esecuzione

da parte della CPU.

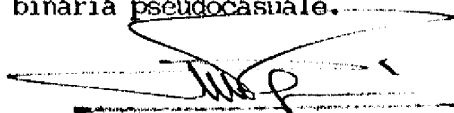
7) Sistema di elaborazione come in una delle rivendicazioni precedenti, in cui i mezzi di abilitazione d'accesso comprendono un generatore di sequenze binarie pseudocasuali, per generare la sequenza di riferimento di byte.

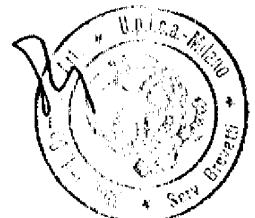
8) Sorgente di programma asportabile da usare in un sistema come in una delle rivendicazioni precedenti e comprendente mezzi per far eseguire dalla CPU istruzioni di programma, quali emettere una sequenza di byte per comparazione, nei mezzi di abilitazione di accesso, con la detta sequenza di riferimento.

9) Metodo per abilitare selettivamente funzioni in un sistema di elaborazione avente una CPU, il metodo comprendendo le fasi di connettere al sistema di elaborazione una sorgente di programma asportabile ed inseribile, eseguire istruzioni immagazzinate nella CPU così da far emettere una sequenza di byte, comparare la sequenza di byte così emessa con una sequenza di byte di riferimento, ed abilitare selettivamente accesso, da parte della CPU, a funzioni operative selezionate, dipendentemente dal risultato della comparazione.

10) Metodo come nella rivendicazione 9), in cui la sequenza di riferimento comprende una sequenza binaria pseudocasuale.

(pt)


dott. Giovanmaria Foggioni della
FUMERO - STUDIO CONSULENZA BREVETTI
iscritto all'Aibo con il N° 35



TAV. 1

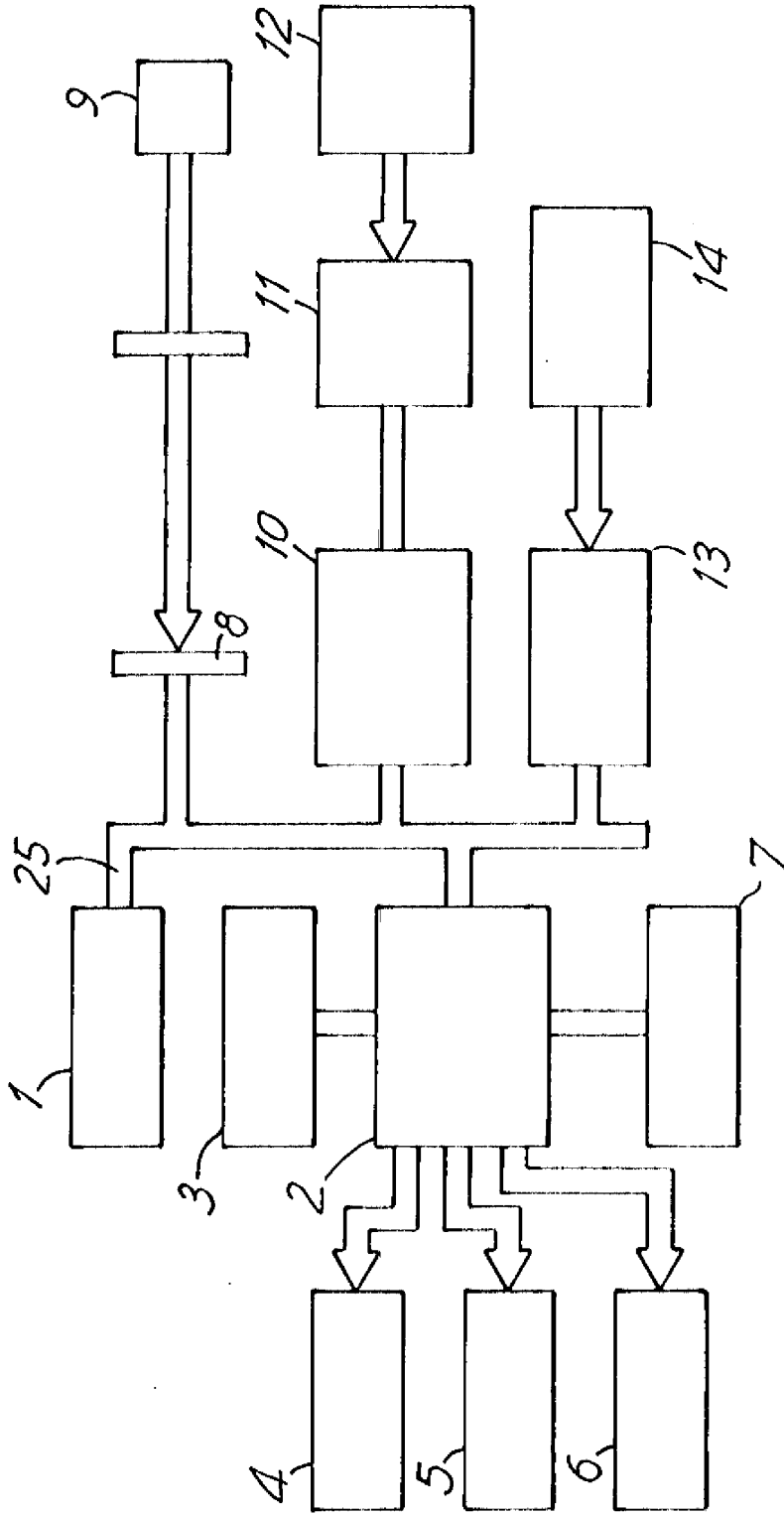


FIG. 1



[Signature]
dott. Giovanna Fagnoli della
FIMERO - STUDIO CONSULENZA BREVETTI
iscritto all'Albo con il n. 35

TAV. 2

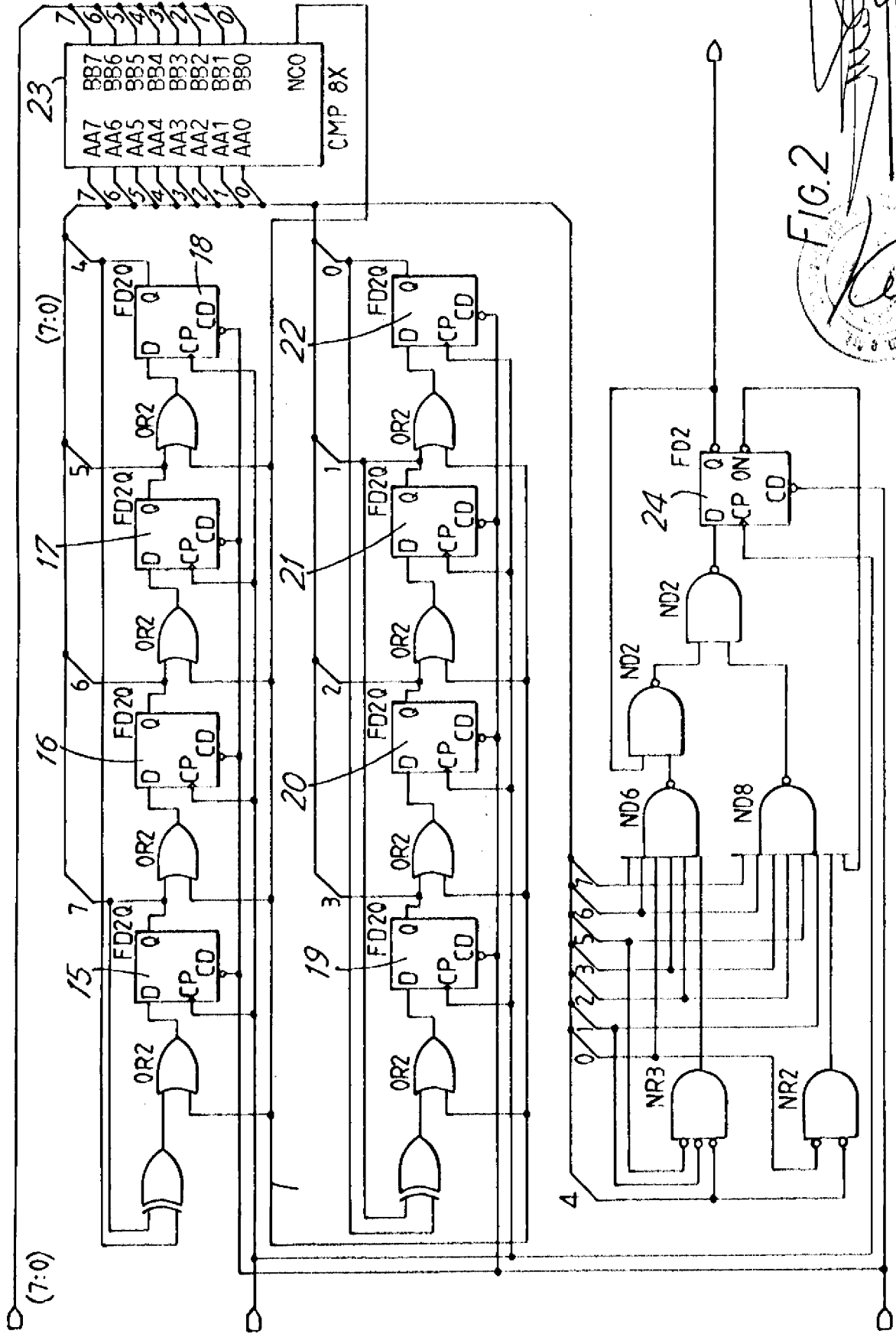


FIG. 2



dot. Giovanni Esposito della
 FIVERO - 00190 ROMA
 iscritto all'Albo con il N. 35

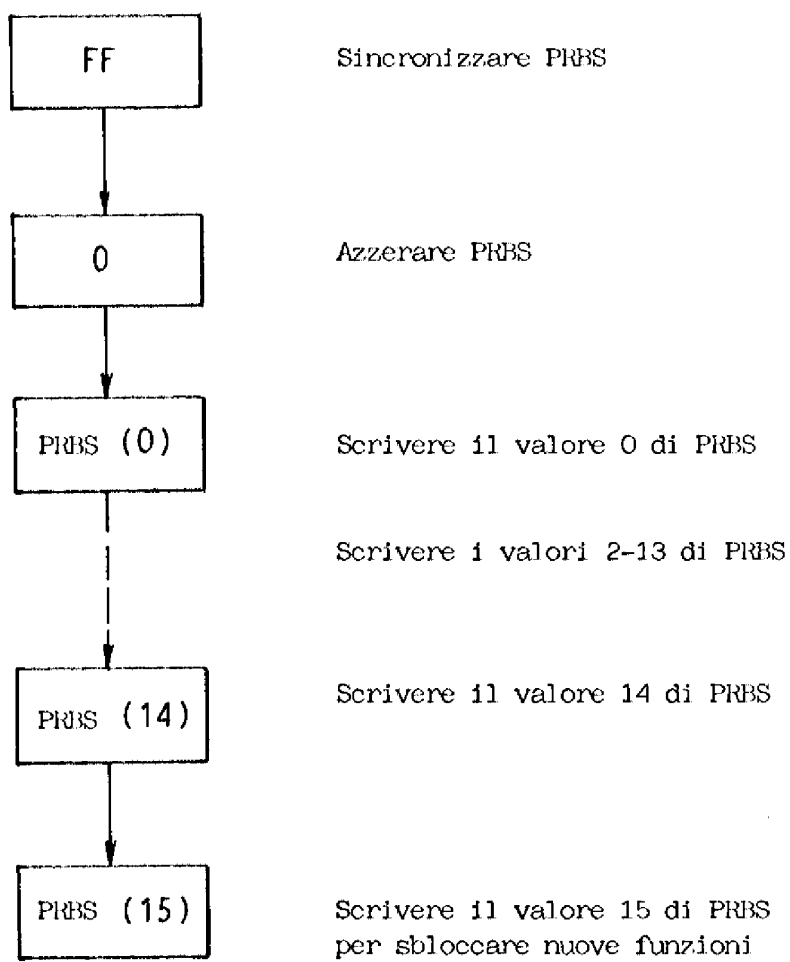


FIG.3

Giovanni Fagnoli

dott. Giovanni Fagnoli della
FUMERU - STUDIO CONSULENZA BREVETTI
iscritto all'Albo con il n. 35

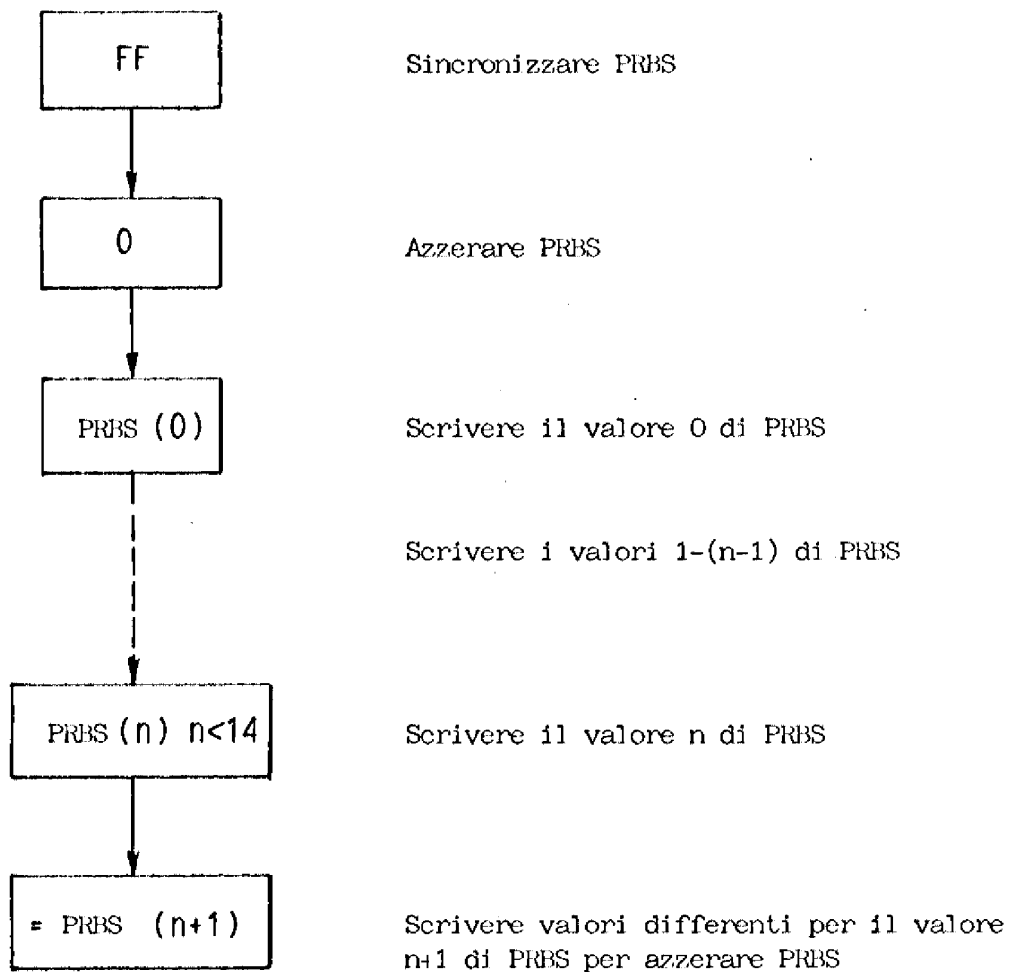


FIG.4



dotl. Giovanni Faggioni della
FUNERO - STUDIO CONSULENZA BREVETTI
iscritto all'Albo con il N. 35

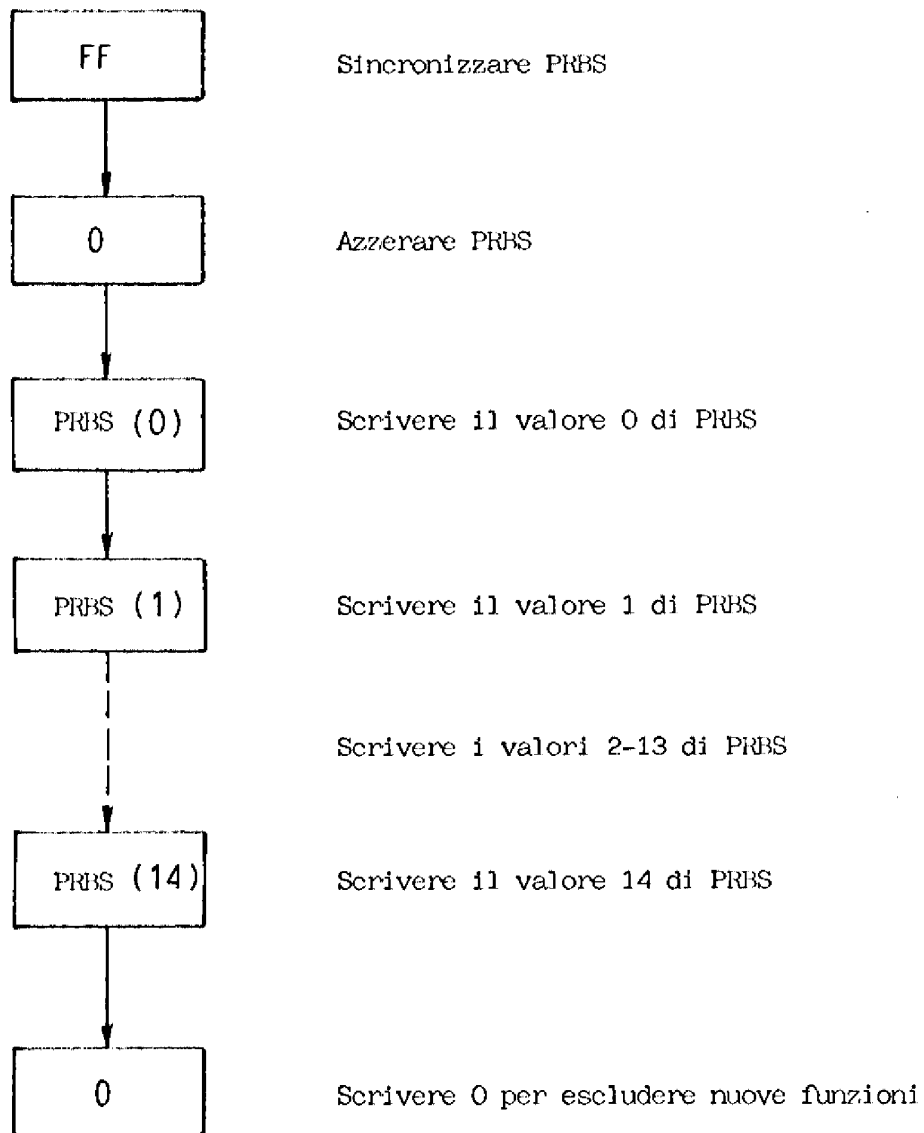
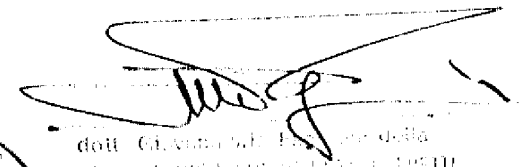


FIG.5


dott. Giovanni B. ... della
FUMERIO S.
iscritto all'Albo con il n. 35

