



MINISTERO DELLO SVILUPPO ECONOMICO
DIREZIONE GENERALE PER LA TUTELA DELLA PROPRIETA' INDUSTRIALE
UFFICIO ITALIANO BREVETTI E MARCHI

UTBM

DOMANDA NUMERO	101982900000642
Data Deposito	24/11/1982
Data Pubblicazione	24/05/1984

Titolo

ORGANIZZAZIONE DI MEMORIA DI CONTROLLO

UFFICIO REGISTRO RHO

la polonizzazione del bello in via bene



DESCRIZIONE DIRETTORE di 1° Cl.
(Silvano Amadori)

dell'invenzione avente per titolo "ORGANIZZAZIONE DI MEMORIA
DI CONTROLLO" della ditta HONEYWELL INFORMATION SYSTEMS ITALIA
S.p.A. di Caluso (Torino) di nazionalità italiana.

Depositata il **24 NOV. 1982**

N. **24400A/82**

RIASSUNTO

In un sistema di elaborazione dati microprogrammato in cui una
sequenza di microistruzioni può essere interrotta ad ogni micro
istruzione per l'esecuzione di una sequenza di microistruzioni
più prioritaria, la memoria di controllo è organizzata in modo
da fornire microistruzioni di lunghezza variabile.

La lunghezza base delle microistruzioni è definita dal paralle
lismo di una prima memoria di controllo.

Per un primo campo di indirizzi di memoria di controllo una se
conda memoria letta in parallelo alla prima fornisce un campo
di microistruzione che si somma al campo base della prima memo
ria e aumenta la lunghezza della microistruzione.

Per un secondo campo di indirizzi di memoria di controllo, una
prima microistruzione può caricare un registro con un campo di
bit contenuti nella microistruzione stessa che viene associato
con la microistruzione seguente per aumentarne la lunghezza.

Per evitare che l'esecuzione della microistruzione di maggior
lunghezza venga influenzata da una interruzione di microprogram
ma occorrente tra l'esecuzione della prima microistruzione e la



lettura della microistruzione seguente, una logica differisce in caso di interruzione, l'associazione tra campo di bit e microistruzione seguente fino al rientro nel microprogramma interrotto.

Una organizzazione di memoria di controllo, così fatta è compatibile con interruzioni di microprogramma e consente sia l'esecuzione veloce di operazioni frequenti, con microistruzioni di maggiore lunghezza, sia l'esecuzione lenta con due cicli di microistruzione, di operazioni poco frequenti che tuttavia richiedono un numero elevato di bit di microistruzione.

.....

La presente invenzione riguarda l'organizzazione della memoria di controllo di un sistema di elaborazione dati microprogrammato.

Un sistema di elaborazione dati microprogrammato sviluppa ogni istruzione di programma mediante l'esecuzione di un microprogramma, ossia una sequenza di microistruzioni contenute in una memoria di controllo.

All'esecuzione di ogni microistruzione corrisponde un ciclo di macchina.

Nel corso di ogni ciclo di macchina la microistruzione corrispondente, attraverso decodifica dei bit che la costituiscono, genera dei segnali elettrici o microcomandi che controllano l'apertura/chiusura di porte logiche, il caricamento di registri di

macchina e in definitiva l'operazione dei singoli elementi elettronici costitutivi del sistema di elaborazione dati.

Ogni istruzione di programma che deve essere eseguita richiede il controllo indipendente di un certo numero di elementi elettronici e dal punto di vista temporale, l'esecuzione successiva di un certo numero di operazioni.

La capacità di una microistruzione di controllare in modo indipendente gli elementi elettronici del sistema dipende dal numero di bit costitutivo ossia dalla sua lunghezza.

Quanto maggiore è la sua lunghezza tanto maggiore è il numero di elementi elettronici che possono essere controllati in maniera indipendente.

Entro certi limiti il progettista può quindi scegliere la lunghezza delle microistruzioni, in modo da ridurre a un minimo la lunghezza delle microistruzioni, con l'attivazione il più possibile sequenziale nel tempo, dei diversi elementi elettronici (sviluppo serializzato delle istruzioni) oppure il numero di microistruzioni successive necessarie per l'esecuzione di una istruzione (sviluppo parallelo delle istruzioni).

Lo sviluppo serializzato delle istruzioni comporta l'uso di memorie di controllo di parallelismo ridotto e globalmente di memorie di controllo con minor numero di bit, e quindi più economiche, ma ha l'inconveniente di aumentare il tempo di esecuzione delle istruzioni e di degradare le prestazioni del sistema.

Lo sviluppo parallelo delle istruzioni riduce al minimo il tempo di esecuzione delle istruzioni ma richiede elevato parallelismo della memoria di controllo e dei circuiti associati, maggiore capacità di memoria e risulta in definitiva più costoso.

In relazione a questi aspetti ed al fatto che le diverse istruzioni hanno esigenze diverse in termini di operazioni sequenziali da eseguire, elementi elettronici da controllare, nonché frequenza di impiego diversa, è desiderabile avere sistemi di controllo con parallelismo variabile.

In questo modo, pur limitando globalmente la capacità delle memorie di controllo è possibile uno sviluppo il più possibile parallelo delle istruzioni di programma di uso frequente e/o che richiedono il controllo di molti elementi elettronici.

Diverse soluzioni sono state proposte per conseguire questo risultato.

Nella domanda di brevetto europea pubblicata il 7.7.82 con il N. 0055392 è descritto un sistema di controllo microprogrammato in cui a una memoria principale di controllo con N locazioni indirizzabili è associata in parallelo una memoria secondaria di controllo con $M \leq N$ locazioni indirizzabili.

Le due memorie, per indirizzi minori di M, sono indirizzate in parallelo e forniscono congiuntamente una microistruzione di parallelismo eguale alla somma del parallelismo delle due memorie.

Per indirizzi maggiori di M viene indirizzata la sola memoria

principale che fornisce microistruzioni di lunghezza eguale al suo parallelismo.

Questa soluzione ha l'inconveniente di limitare la possibilità di espansione delle microistruzioni a un campo determinato dei possibili indirizzi di memoria di controllo e impone un vincolo stringente nella allocazione dei microprogrammi.

Tutti i microprogrammi per i quali una microistruzione richiede una lunghezza maggiore del parallelismo della memoria principale di controllo devono infatti risiedere in una prefissata zona di memoria o alternativamente, l'accesso alle microistruzioni deve essere ottenuto mediante microistruzioni di salto residenti in un'altra zona, ciò che richiede aumento delle dimensioni di memoria e degradazione delle prestazioni.

Nel brevetto U.S.A. N. 4,251,862 è descritto un sistema di controllo microprogrammato in cui a una memoria principale di controllo con N locazioni indirizzabili è associata una memoria secondaria con $M \leq N$ locazioni indirizzabili.

Una microistruzione letta nella memoria principale può specificare, mediante un opportuno campo di bit rappresentativo di un indirizzo della memoria secondaria, e un bit di qualificazione, che la microistruzione successiva deve essere "espansa".

In questo caso con l'indirizzamento della memoria principale per la lettura della microistruzione successiva, viene anche indirizzata la memoria secondaria e viene letto un codice binario rappresentativo di una espansione di microistruzione.



In questo modo si ottiene una microistruzione composta di maggior lunghezza a scapito di una certa riduzione della microistruzione precedente.

Questa soluzione ha il vantaggio di consentire l'espansione delle microistruzioni in qualsiasi locazione di memoria ma presenta altri inconvenienti, tra i quali il più grave è che il sistema risulta incompatibile con la necessità, presente nei sistemi di controllo, di interrompere l'esecuzione di un microprogramma per l'esecuzione di un microprogramma più prioritario a seguito di una interruzione.

E' noto infatti che nei sistemi di elaborazione dati è generalmente previsto che un programma, per esempio di calcolo interno o di servizio per una periferica lenta, possa essere interrotto in qualsiasi istante per l'esecuzione di operazioni richieste da una periferica veloce, come per esempio una unità a dischi.

L'interruzione è in generale riconosciuta a livello di ciclo di macchina ossia di microistruzione. In altre parole può avvenire che nel corso dell'esecuzione di una microistruzione che richiama un codice di espansione per la microistruzione seguente di microprogramma, abbia luogo una interruzione di microprogramma.

In questo caso la microistruzione eseguita subito dopo è la prima microistruzione del microprogramma interrompente a cui viene erroneamente associato il codice di espansione.

Un altro inconveniente si presenta nel caso che il sistema debba soddisfare al requisito della modificabilità, ossia sia costi

tuito da memorie almeno in parte a lettura/scrittura.

In questo caso è necessario che sia la memoria principale che quella secondaria siano a lettura/scrittura pena la non modificabilità del codice di espansione.

Ciò comporta la necessità di predisporre circuiti logici di controllo di scrittura per entrambe le memorie primaria e secondaria con complicazioni e costi non indifferenti.

Ovvia a questi inconvenienti l'organizzazione della memoria di controllo oggetto della presente invenzione che consente di ottenere microistruzioni di lunghezza variabile, in grado di soddisfare esigenze operative diverse e comunque compatibili con la possibilità di interruzione del microprogramma.

In particolare l'organizzazione oggetto dell'invenzione consente di ottenere microistruzioni di maggior lunghezza per un campo prefissato di indirizzi, dette microistruzioni essendo utilizzabili per lo sviluppo parallelo e veloce di istruzioni di uso frequente e microistruzioni di maggiore lunghezza per il rimanente campo di indirizzi, dette microistruzioni essendo utilizzabili per lo sviluppo di operazioni che richiedono il controllo parallelo di più elementi elettronici, ma sono di uso poco frequente.

Questi vantaggi sono ottenuti mediante una organizzazione di memoria di controllo comprendente una memoria principale, una memoria secondaria in parallelo a un campo della memoria principale e un registro secondario di microistruzione destinato a

memorizzare una porzione di microistruzione per il suo impiego come codice di espansione della microistruzione successiva.

Sono inoltre previsti dei circuiti logici molto semplici che consentono di espandere selettivamente una microistruzione con il codice estratto dalla memoria secondaria o da detto registro e che associano correttamente il codice di espansione alla relativa microistruzione anche in caso di interruzione di microprogramma.

L'organizzazione trova un ulteriore vantaggioso impiego in sistemi di controllo in cui una porzione almeno della memoria di controllo è di tipo a lettura/scrittura, e consente la modificabilità delle microistruzioni e del codice di espansione per almeno detta porzione richiedono circuiti logici di controllo di scrittura per una sola memoria.

L'organizzazione trova inoltre vantaggioso impiego in sistemi di controllo microprogrammati in cui i microprogrammi risiedono almeno in parte nella memoria di lavoro del sistema anzichè nella memoria di controllo come per esempio descritto nel brevetto britannico N. 1,440,856 del 26.10.1976.

Questi vantaggi e le caratteristiche dell'invenzione risulteranno più chiare dalla descrizione di una forma preferita di realizzazione e dai disegni allegati in cui:

La fig. 1 rappresenta a titolo esemplificativo il formato di una microistruzione;

La Fig. 2 rappresenta in schema a blocchi una organizzazione pre

ferita di memoria di controllo secondo l'invenzione.

La Fig. 3 rappresenta in diagramma di temporizzazione alcuni segnali di temporizzazione per l'organizzazione di memoria di controllo di Fig. 2.

Prima di descrivere l'organizzazione della memoria di controllo oggetto dell'invenzione è opportuno illustrare la struttura dei microprogrammi che presiedono al funzionamento dei sistemi microprogrammati di elaborazione dati.

Un microprogramma è un insieme ordinato di microistruzioni.

L'azione di controllo esercitata da una microistruzione è detta esecuzione della microistruzione.

A ogni ciclo di macchina corrisponde l'esecuzione di una microistruzione e il richiamo o fetching della microistruzione da eseguire nel ciclo successivo.

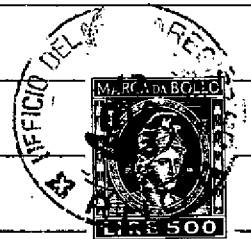
L'organizzazione dei microprogrammi è in generale sequenziale.

In altre parole un microprogramma è costituito da microistruzioni collocate nella memoria di controllo a indirizzi consecutivi di memoria.

Questa organizzazione consente l'indirizzamento di una microistruzione in modo semplice, mediante incrementazione dell'indirizzo della microistruzione precedente.

Dalla microistruzione di indirizzo generico K si passa cioè alla microistruzione di indirizzo K+1.

Tuttavia non è possibile nè conveniente seguire integralmente questo criterio organizzativo.



E' quindi previsto che da una microistruzione di indirizzo K si possa saltare a una microistruzione di indirizzo generico A o K+C.

Le microistruzioni che comandano questo modo di aggiornare l'indirizzo per il richiamo della microistruzione seguente sono dette di salto assoluto e relativo rispettivamente.

Esse definiscono il codice di indirizzo di salto C in forma diretta o indiretta (mediante riferimento a un registro che contiene detto codice).

Le microistruzioni di salto possono essere incondizionate o condizionate.

Le microistruzioni di salto incondizionato eseguono l'operazione di salto in ogni caso.

Le microistruzioni di salto condizionato, condizionano l'operazione di salto al verificarsi di eventi prestabiliti nel sistema di elaborazione dati.

E' la microistruzione stessa che in forma diretta o indiretta definisce la condizione da verificare.

Infine le microistruzioni di salto possono prevedere la riserva o il salvataggio di indirizzo.

In altre parole una microistruzione di salto di indirizzo K può eseguire il fetching di una microistruzione di indirizzo A o K+C e al tempo stesso può eseguire il salvataggio dell'indirizzo successivo K+1 in un opportuno registro.

Il salvataggio di indirizzo è utile per il rientro nel flusso

sequenziale di un microprogramma da cui si sia usciti con una microistruzione di salto, per l'esecuzione di una subroutine di microprogramma.

In questo caso la microistruzione di rientro definisce in forma indiretta, con riferimento al registro di salvataggio, l'indirizzo assoluto di rientro.

Poichè in generale da una subroutine si può passare ad una ulteriore subroutine, ossia è consentito l'"annidamento" di subroutine, l'una dentro l'altra, sono usati più registri di salvataggio, organizzati in forma di catasta o stack in cui si accumulano uno sopra l'altro i diversi indirizzi salvati e da cui si estraggono i diversi indirizzi salvati in ordine inverso a quello di salvataggio.

L'ultimo indirizzo salvato è cioè il primo ad essere estratto.

L'uso di una catasta di registri o stack, consente inoltre l'interruzione di un microprogramma.

Nei moderni sistemi di elaborazione dati è infatti previsto che un processo in corso, eseguito mediante un microprogramma o una sequenza di microprogrammi, possa essere interrotto in qualsiasi momento per effettuare operazioni più prioritarie.

Per esempio alcune unità periferiche come le unità a dischi non bufferizzate richiedono che lo scambio di informazioni tra di esse e l'unità centrale avvenga ad istanti prestabiliti determinati dalla periferica stessa.

L'unità centrale deve quindi essere in grado di interrompere le

operazioni in corso per iniziare il processo di scambio di informazioni richiesto dalla periferica, al termine del quale può riprendere il processo interrotto.

Questi tipi di richiesta vengono presentati all'unità centrale mediante appositi segnali di interruzione o semplicemente interruzioni.

Una interruzione ricevuta nel corso di esecuzione di una microistruzione K, provoca il salvataggio dell'indirizzo K+1 e la generazione di un indirizzo corrispondente alla prima microistruzione di un programma di trattamento dell'interruzione per servire la periferica interrompente.

Alla fine del trattamento dell'interruzione il microprogramma interrotto può quindi essere ripreso all'indirizzo K+1.

La figura 1 rappresenta a titolo di esempio il formato di una microistruzione.

Le microistruzioni, per esempio di 32 bit ciascuna, contengono un campo per esempio di 8 bit, che definisce un codice operativo OP di microistruzione.

I rimanenti 24 bit assumono un significato variabile in funzione del codice operativo.

Per una microistruzione di trasferimento, possono per esempio definire in codice due registri tra i quali deve essere trasferita una informazione.

Per una microistruzione di salto, definiscono un nuovo indirizzo assoluto, una costante di salto, l'eventuale condizione di

salto, l'eventuale riserva di indirizzo.

Per una microistruzione operativa (somma, sottrazione, ecc.) de-
finiscono in codice i registri che contengono gli operandi, il
registro di destinazione del risultato dell'operazione ecc.

Tra le diverse microistruzioni è compresa anche la microistru-
zione di caricamento costante, che è rappresentata in fig. 1.

Tale microistruzione comprende un campo (OP) di operazione (bit
07), un campo (ADDR) di registro a cui è destinata la costante,
(bit 8-11), un campo CONST (bit 24-31) che contiene un codice
di 8 bit rappresentativi della quantità o costante desiderata.

I bit 12-23 sono disponibili per altri scopi.

Il formato descritto è puramente esemplificativo.

Si può ora considerare l'organizzazione della memoria di control-
lo oggetto della presente invenzione rappresentata in Fig. 2.

Con riferimento alla fig. 2 una memoria di controllo principa-
le MCS 1, avente per esempio un parallelismo di 32 bit e una
capacità di 64K locazioni indirizzabili, è indirizzata attra-
verso un canale 2 a 16 bit connesso alle uscite di un registro
di indirizzamento ROSPA 3.

Le uscite della memoria principale sono connesse attraverso il
canale 4 agli ingressi di un registro di microistruzione ROR 5,
le cui uscite ROR (0-31) sono connesse agli ingressi di una re-
te di decodifica 6.

Una unità di generazione indirizzi AGU 7, provvede a generare
gli indirizzi di microistruzione.



Le sue uscite sono connesse attraverso un gruppo di porte tristate 8 e il canale 9 agli ingressi del registro ROSPA 3.

L'unità di generazione indirizzi AGU 7 è di tipo convenzionale e riceve in ingresso:

il contenuto del registro 3, attraverso un canale 10;

il contenuto di parte del registro 5 (per esempio i bit ROR (15-31) attraverso un canale 11;

il contenuto di uno tra più registri del sistema di elaborazione dati, attraverso un canale 12.

L'unità AGU 7 può per esempio essere realizzata con circuiti integrati prodotti dalla ditta AMD con codice Am2930.

Una descrizione dettagliata dell'unità 7 è irrilevante ai fini dell'invenzione.

È sufficiente menzionare che l'unità 7 contiene una rete di somma, una rete di incrementazione, circuiti multiplexer e uno stack per consentire la generazione di indirizzi sequenziali o di salto ed i salvataggi di indirizzo già descritti.

Ciò è fatto utilizzando le informazioni presenti sui canali 10, 11, 12, per effetto di comandi opportuni ricevuti dall'unità 7.

Tali comandi, non illustrati sono ottenuti mediante decodifica della microistruzione in corso di esecuzione.

Il registro ROSPA 3, può essere caricato oltre che dalle uscite dell'unità 7 anche da un canale di indirizzi di interruzione 13 connesso al canale 9 attraverso un gruppo di porte tristate 14.

Un segnale di interruzione EINT, opportunamente temporizzato,

abilita quando presente il gruppo di porte 14 e inibisce il gruppo di porte 8, attraverso il NOT 15.

L'indirizzo di interruzione può essere un indirizzo "forzato" ossia ottenuto connettendo opportunamente i diversi fili del canale 13 al livello logico elettrico "1" o "0".

Alternativamente può essere ottenuto da un conveniente registro di macchina selezionato dal segnale di interruzione EIINT.

Le uscite del registro 5, ROR (0-31) sono connesse agli ingressi di un decodificatore 6, che decodifica la microistruzione contenuta nel registro ROR 5 in una pluralità di microcomandi MC1, MCN, ciascuno dei quali controlla uno o più elementi elettronici del sistema di elaborazione dati, eventualmente in associazione con segnali di temporizzazione.

Alcuni di questi microcomandi sono inviati, come già accennato, all'unità 7, ma poichè non sono rilevanti ai fini dell'invenzione non sono rappresentati nei disegni.

Una unità di temporizzazione TU16, di tipo convenzionale, genera i segnali ciclici di temporizzazione necessari per il funzionamento del sistema.

Per gli scopi dell'invenzione è sufficiente indicare che essa genera tra l'altro i segnali di temporizzazione rappresentati nel diagramma di Fig.3.

All'istante t_0 , corrispondente all'inizio di un ciclo di macchina viene generato un segnale STRORA.

Il fronte di salita di STRORA è usato per caricare nel registro

ROR 5 la microistruzione letta nella memoria di controllo 1.

All'istante t_1 intermedio nel ciclo di macchina viene generato un segnale STINT.

Il segnale STINT è usato come temporizzazione per i segnali di interruzione.

All'istante t_2 successivo a t_1 viene generato un segnale STMI usato come temporizzazione per il caricamento di registri di macchina.

All'istante t_3 viene generato il segnale STADRA.

Il fronte di salita di STADRA è usato per caricare il registro ROSPA 3.

Il ciclo di macchina risulta quindi di due fasi.

Nella prima fase, esecutiva, la microistruzione da eseguire è resa disponibile nel registro ROR 5, decodificata ed eseguita.

Vengono definite le modalità di definizione dell'indirizzo della microistruzione da eseguire nel ciclo seguente, ivi compreso il riconoscimento di eventuali interruzioni di microprogramma.

Nella seconda fase, che ha inizio con STADRA, viene completata l'esecuzione della microistruzione corrente e viene avviata, con il caricamento di ROSPA, la lettura della microistruzione successiva.

La seconda fase corrisponde cioè a un ciclo di lettura della memoria di controllo.

Al termine della seconda fase, ha inizio un nuovo ciclo di mac

china.

La porzione della organizzazione di memoria di controllo fin qui descritta è del tutto convenzionale.

Vengono ora considerati gli elementi innovativi con riferimento alla Figura 1.

Una memoria secondaria di controllo 17, con un numero di locazioni indirizzabili opportuno e inferiore a quello della memoria principale, per esempio 2K, è associata in parallelo alla memoria principale.

La memoria 17 è indirizzata, attraverso il canale 18, dai bit meno significativi CSA (05-15) in uscita dal registro 3.

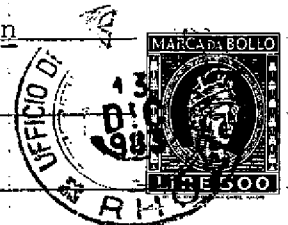
Le uscite della memoria 17, che può avere un parallelismo opportuno, per esempio 8 bit, sono connesse, mediante un canale 19 a un primo gruppo di ingressi di un multiplexer 20.

Le uscite del multiplexer 20, sono connesse a corrispondenti ingressi di un registro 41 indicato con ROR A, in quanto svolge la stessa funzione del registro ROR 5.

Il registro 41, come il registro ROR 5 è abilitato dal segnale STRORA.

I due registri 5 e 41, per quanto rappresentati come distinti, possono costituire di fatto un unico registro con parallelismo eguale alla somma del parallelismo della memoria 1 e della memoria 17.

Le uscite del registro 41 sono connesse a corrispondenti ingressi del decodificatore 6.



L'informazione contenuta nel registro 41, convenientemente decodificata fornisce microcomandi addizionali a quelli generati con la decodifica del contenuto di ROR 5.

Un secondo gruppo di ingressi 20A del multiplexer 20 è permanentemente connesso a livello logico "0".

Un terzo e quarto gruppo di ingressi del multiplexer 20 sono connessi attraverso il canale 21, alle uscite di un registro 22.

Gli ingressi del registro 22, a loro volta sono connessi attraverso il canale 23 a convenienti uscite del registro ROR 5, per esempio le uscite ROR (24-31).

L'ingresso di clock CK del registro 22 è connesso all'uscita di una porta AND 24 che riceve in ingresso il segnale STMI e un microcomando MC1.

Quando la microistruzione in corso di esecuzione è una microistruzione di caricamento costante e indica il registro 22 come destinatario della costante, MC1 è a livello logico "1" e la costante, presente sul canale 23 è caricata nel registro 22.

L'organizzazione della memoria di controllo è completata da due flip-flop e dalle porte logiche 27, 28, 29, 30, 31.

Il flip-flop 25, di tipo JK riceve sull'ingresso J il microcomando MC1.

Il suo ingresso K è permanentemente connesso a livello logico "1".

Al suo ingresso di clock CK è applicato il segnale STMI.

Con il fronte di salita di STMI se MC1=1 il flip-flop 25 è posto in set.

In caso contrario non cambia di stato.

Il flip-flop 26 di tipo JK ha l'ingresso J connesso all'uscita della porta AND 27.

Il suo ingresso K è permanentemente connesso a livello logico "1".

Al suo ingresso di clock CK è applicato il segnale STMI.

Il flip-flop 26 è posto in set dal fronte di salita di STMI se contemporaneamente il segnale all'uscita della porta 27 è a livello logico "1".

In caso contrario non cambia di stato.

La porta AND 27 riceve in ingresso un segnale di interruzione INT e il segnale di temporizzazione STINT e genera all'uscita il segnale EIINT.

Il flip-flop 26 ha l'ingresso di reset R connesso all'uscita della porta NAND 28, e viene posto in reset da un segnale a livello logico "0" applicato all'ingresso di reset.

La porta NAND 28 riceve in ingresso il segnale STMI e un micro comando MC2.

Il microcomando MC2 è a livello logico "1" quando una microistruzione è di fine interruzione, ossia comanda il rientro in un microprogramma interrotto.

Il flip-flop 25 ha l'ingresso di reset R connesso all'uscita della porta NAND 29.

La porta NAND 29 riceve a un ingresso il segnale STRORA.

Un secondo ingresso è connesso all'uscita \bar{Q} del flip-flop 27.

Il multiplexor 20 è controllato in selezione, attraverso due ingressi di selezione S_1 , S_2 .

L'ingresso S_1 è connesso all'uscita della porta AND 30, che ha gli ingressi connessi rispettivamente all'uscita Q del flip-flop 25 e all'uscita \bar{Q} del flip-flop 26.

L'ingresso S_2 è connesso all'uscita di un OR 31, che riceve in ingresso i bit CSA (00-04) di indirizzamento della memoria di controllo.

Quando i bit CSA (00-04) sono tutti a livello logico 0, ossia l'indirizzo è $\leq 2K$, l'uscita dell'OR 31 è a livello logico 0.

Il funzionamento del multiplexor 20 è il seguente:

Quando $S_2 = S_1 = 0$, il canale 19 del multiplexor è connesso alle uscite.

Quando $S_2 = 1$ e $S_1 = 0$ il gruppo di ingressi 20A è connesso alle uscite del multiplexor.

Per $\sqrt{S_1} = 1$, indipendentemente dal livello logico di S_2 , il canale 21 è connesso alle uscite del multiplexor.

Il funzionamento dell'organizzazione descritta è il seguente.

Quando la memoria di controllo principale 1 è indirizzata a indirizzi minori di 2K, ossia CSA (00-04) = 0 parallelamente alla memoria principale 1 viene indirizzata la memoria ausiliaria 17, e l'informazione letta in tale memoria è trasferita attraverso il multiplexor 20 e caricata nel registro 41.

Si ottiene cioè ad ogni ciclo di macchina, una microistruzione "allargata" con numero di bit eguale alla somma del parallelismo della memoria 1 e della memoria 17.

Quando la memoria di controllo principale 1 è indirizzata a indirizzi maggiori o eguali a 2K, ossia almeno un bit CSA.(00-04) = 1, attraverso il multiplexor 20 viene forzato nel registro ROR 1 un codice "tutti zeri".

Si ottiene cioè ad ogni ciclo di macchina una microistruzione di lunghezza normale, con numero di bit eguale al parallelismo della memoria di controllo 1.

Tuttavia a qualsiasi indirizzo K della memoria di controllo 1 è possibile collocare una microistruzione di caricamento costante che impone il caricamento di una costante nel registro 22.

In questo caso, quando la microistruzione di indirizzo K viene letta e caricata nel registro ROR 5, nasce il microcomando MC1.

Durante l'esecuzione di tale microistruzione, con il segnale STMI viene posto in set il flip-flop 25 e una costante opportuna contenuta nella microistruzione è caricata nel registro 22.

Se nessuna interruzione viene riconosciuta durante l'esecuzione della microistruzione, il flip-flop 26 rimane in stato di reset e l'uscita della porta AND 30 è perciò a livello logico 1.

Con l'inizio del ciclo seguente, una nuova microistruzione è caricata nel registro ROR 5 e contemporaneamente l'informazione contenuta nel registro 22 è trasferita nel registro 41, con il fronte di salita di STRORA.



Si ottiene cioè una microistruzione "allargata".

L'espansione della microistruzione è ottenuta a spese della microistruzione precedente che deve essere di caricamento costante e nella quale dunque un certo campo deve essere impiegato per definire la costante.

Al tempo stesso il segnale STRORA, attraverso la porta NAND 29 pone in reset il flip-flop 25.

S_1 cade a "0" con un ritardo, rispetto a STRORA, determinato dalla propagazione dei segnali nella catena logica AND 29, flip-flop 25 AND 30.

Se nel corso della esecuzione della microistruzione di indirizzo K viene riconosciuta una interruzione, insieme al flip-flop 25 viene posto in set il flip-flop 26.

Pertanto S_1 si mantiene a livello logico "0" e il contenuto del registro 22 non viene trasferito, con il segnale STRORA seguente nel registro 41.

Per tutti i cicli di macchina nel corso dei quali si sviluppa il microprogramma di trattamento dell'interruzione, i flip-flop 25 e 26 rimangono in stato di set.

E' solo con l'ultimo ciclo di macchina, che una microistruzione di fine trattamento interruzione e rientro nel microprogramma interrotto, con la generazione del microcomando MC2, determina il reset del flip-flop 26.

Pertanto con il successivo caricamento della microistruzione nel registro ROR 5, (segnale STRORA), anche il contenuto del

registro 22 viene trasferito nel registro 21 e il microprogramma interrotto può essere ripreso senza perdita di informazioni.

E' da notare che con l'organizzazione descritta, un microprogramma di interruzione non può usare microistruzioni "allargate" mediante l'uso del registro 22, perchè questo può essere già carico con informazioni necessarie nella esecuzione di un microprogramma interrotto.

Questa limitazione è di fatto senza conseguenze, perchè il microprogramma di interruzione può risiedere a indirizzi della memoria di controllo compresi tra 0 e 2K e per i quali una espansione delle microistruzioni può essere ottenuta per mezzo della memoria secondaria 17.

Appare quindi evidente che con l'organizzazione oggetto della presente invenzione si conseguono numerosi vantaggi.

Per un campo di indirizzi della memoria di controllo le microistruzioni possono essere "allargate" mediante l'uso di una memoria secondaria di capacità ridotta indirizzata in parallelo alla memoria principale.

Eventuali interruzioni di microprogramma non affettano la corretta esecuzione delle operazioni di controllo.

Questa espansione delle microistruzioni può essere vantaggiosamente impiegata per l'esecuzione veloce in parallelo di operazioni che richiedono il controllo di molti elementi elettronici del sistema e/o che occorrono con notevole frequenza.

Questo è il caso di operazioni di interpretazione di istruzioni

ni e di trattamento di interruzioni.

Per tutta la capacità di memoria le microistruzioni possono essere "allargate" mediante l'uso di una porzione della microistruzione precedente come espansione della microistruzione seguente.

Questa espansione comporta la necessità di far precedere alla microistruzione allargata una microistruzione che in generale ha la sola funzione di caricare una costante nel registro 22 e non può essere usata per altri scopi.

Ne deriva, in generale un rallentamento delle operazioni compiute dal sistema.

Questa espansione delle microistruzioni può quindi essere vantaggiosamente impiegata per l'esecuzione di operazioni diagnostiche che richiedono il controllo parallelo di molti elementi elettronici, ma che sono di uso infrequente.

Anche in questo caso, eventuali interruzioni di microprogramma non affettano la corretta esecuzione delle operazioni di controllo.

E' evidente che la descrizione che precede riguarda solo una forma di realizzazione preferita dell'invenzione e che molteplici varianti possono essere effettuate senza dipartirsi dallo scopo dell'invenzione.

In particolare la memoria secondaria 17 può essere associata a un qualsiasi campo di indirizzi della memoria principale 1, anzichè al campo di indirizzi di minor peso.

In questo caso la porta OR 31 è sostituita da un decodificatore di indirizzi.

RIVENDICAZIONI

1. Organizzazione di memoria di controllo, per calcolatore microprogrammato, in cui una sequenza di microistruzioni, ciascuna eseguita in un ciclo di macchina può essere interrotta, ad ogni ciclo di macchina, per l'esecuzione di una sequenza di microistruzioni più prioritaria per effetto di un segnale di interruzione, comprendente:

- una memoria di controllo principale con N locazioni indirizzabili, provvista di ingresso di indirizzamento ed uscita
- una memoria di controllo secondaria con $M < N$ locazioni indirizzabili, provvista di ingresso di indirizzamento ed uscita
- mezzi di indirizzamento congiunto di dette memorie connessi all'ingresso di indirizzamento di dette memorie,
- un registro di microistruzione avente un primo campo connesso all'uscita di detta memoria principale e un secondo campo,

caratterizzata da ciò che comprende inoltre:

- un registro di espansione attivato da una microistruzione prestabilita nel corso del ciclo di esecuzione corrispondente per memorizzare un campo di detta microistruzione,
- mezzi di trasferimento selettivo, controllati da un primo e secondo segnale di selezione, provvisti di una pluralità



di gruppi di ingresso e di una uscita, un primo gruppo di ingressi essendo connesso all'uscita di detta memoria secondaria, un secondo gruppo di ingressi essendo connesso alle uscite di detto registro di espansione, un terzo gruppo di ingressi essendo alimentato da una configurazione logica prestabilita, detta uscita essendo connessa al secondo campo di detto registro di microistruzione,

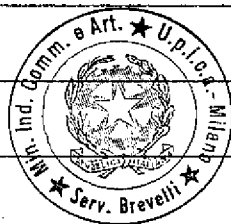
- primi circuiti logici, riceventi in ingresso una porzione almeno di indirizzi di microistruzione, da detti mezzi di indirizzamento, per generare detto primo segnale di selezione per predeterminati indirizzi di microistruzione, detto primo segnale abilitando detti mezzi di trasferimento al trasferimento di informazioni da detto primo gruppo di ingressi all'uscita e
- secondi circuiti logici, attivati da detta microistruzione prestabilita, bloccati da detto segnale di interruzione e riposti in stato inattivo da una seconda microistruzione prestabilita per produrre detto secondo segnale di selezione all'inizio del ciclo di esecuzione della microistruzione che segue, nel microprogramma, a detta microistruzione prestabilita, detto secondo segnale abilitando detti mezzi di trasferimento al trasferimento di informazioni da detto secondo gruppo di ingressi all'uscita.

2. Organizzazione di memoria di controllo come a rivendicazione 1 caratterizzata da ciò che detti mezzi di trasferimento

selettivo consistono in un multiplexer.

HONEYWELL INFORMATION SYSTEMS ITALIA S. p. A

Carlo Talub



l' Ufficiale Rogante
(Gilia Russo)

Gilia Russo

UFFICIO REGISTRO RHO

Le polizza di belbo in un bene

*UFFICIO RHO Cl.
(Filippo Amaducci)*



24400A/82

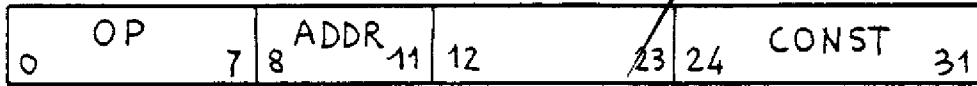


FIG. 1

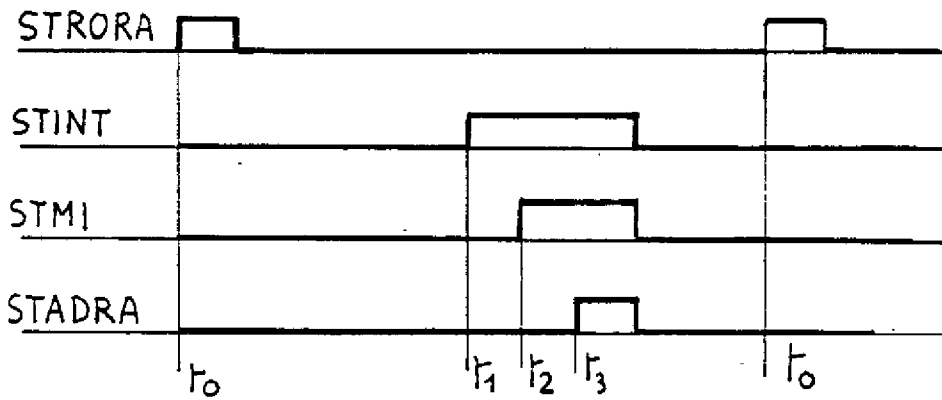
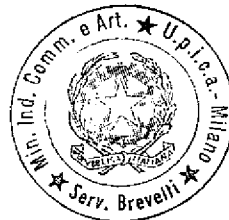


FIG. 3



l'Ufficiale Rogante
(Gillia Russo)

[Signature]

HONEYWELL INFORMATION SYSTEMS ITALIA S.p.A.

[Signature]