



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

# UTBM

|                           |                        |
|---------------------------|------------------------|
| <b>DOMANDA NUMERO</b>     | <b>101980900000681</b> |
| <b>Data Deposito</b>      | <b>02/09/1980</b>      |
| <b>Data Pubblicazione</b> | <b>02/03/1982</b>      |

Titolo

UNITA' DI CONTROLLO DI UN MODULO INGRESSO-USCITA DI UN ELABORATORE  
ELETTRONICO

TESTO DELLA DESCRIZIONE

Il presente trovato si riferisce ad una disposizione circuitale microprogrammata atta a gestire una unità funzionale di un elaboratore elettronico, nel seguito detto modulo, comprendente una pluralità di circuiti di interfaccia ingresso-uscita e circuiti ausiliari.

Le operazioni di trasferimento dati tra la memoria centrale o l'unità logica centrale (CPU) di un elaboratore elettronico e le unità periferiche, tramite i circuiti di interfaccia associati a ciascuna unità periferica, richiedono l'esecuzione di una serie di microistruzioni. Per non impegnare inutilmente la CPU è conveniente delegare la gestione di queste operazioni ad organi decentrati che interagiscono direttamente con i circuiti di interfaccia. Tali organi decentrati, detti nel seguito unità di controllo, hanno preferibilmente una struttura semplice, il che limita il numero delle interfacce che ciascuna unità di controllo può gestire.

La parte ingresso-uscita dell'elaboratore assume così una struttura modulare in cui ciascun modulo comprende una pluralità di circuiti di interfaccia e l'unità di controllo che li gestisce, mentre il messaggio inviato dalla CPU contiene l'indirizzo del modulo, quello dell'interfaccia del modulo e l'ordine all'unità di controllo di compiere un'operazione attivando il microprogramma adeguato.

Scopo del presente trovato è la realizzazione di una unità di controllo comprendente circuiti atti a sincronizzarla rapidamente con la CPU, sia all'avvio di una microistruzione che nel corso della stessa, nonché circuiti atti a verificare la correttezza della sequenza dei mi-

microordini nell'ambito di una microistruzione ed il completamento della microistruzione stessa.

Un'unità di controllo secondo il trovato comprende in combinazione:

- una unità di codifica che, in risposta ad una richiesta di esecuzione di microistruzione, genera un codice che viene scritto in un primo registro;
- un contatore atto a scandire i microordini che costituiscono una microistruzione;
- una prima memoria, indirizzata dal codice scritto nel primo registro e dal contatore, contenente le microistruzioni operative;
- una seconda memoria indirizzata dal contatore e contenente le microistruzioni di scansione ciclica delle unità di interfaccia facenti parte del modulo;
- un primo circuito atto a sincronizzare l'inizio delle microistruzioni eseguite dall'unità di controllo con quelle della CPU;
- un secondo circuito atto ad arrestare una microistruzione in corso di svolgimento nell'unità di controllo in attesa della comparsa o scomparsa di almeno un segnale emesso dalla CPU;
- un terzo circuito atto a controllare la correttezza successiva dei microordini nell'ambito di una microistruzione ed il completamento della microistruzione stessa;
- mezzi atti ad abilitare l'esecuzione di una microistruzione operativa e della microistruzione di scansione ciclica.

Il trovato sarà ora descritto con riferimento ad un esempio non limitativo di realizzazione descritto nelle figure allegate dove:

- la figura 1 mostra uno schema a blocchi di un'unità di controllo secondo il trovato;

- la figura 2 mostra un esempio di realizzazione dei circuiti di sincronizzazione CSL e FL di figura 1;

- la figura 3 mostra un esempio di realizzazione del circuito MPC di figura 1.

In figura 1 è indicato lo schema a blocchi di un'unità di controllo secondo il trovato.

I segnali indicanti la microistruzione da svolgere, codificati dal codicificatore CE e memorizzati nel registro CC, costituiscono i bit più significativi degli indirizzi di una memoria CS (Control Store) nelle cui celle sono inseriti i microordini che compongono ciascuna microistruzione; i bit meno significativi dell'indirizzo sono generati da un contatore PC, che indirizza anche una seconda memoria PS dove sono inserite le microistruzioni di scansione ciclica dei circuiti di interfaccia per evidenziare richieste da inviare alla CPU. La scansione ciclica ha luogo quando l'unità di controllo non è impegnata per eseguire uno dei microprogrammi contenuti nella memoria CS. Le memorie CS e PS e quella che costituisce il contatore PC sono seguite da un registro R, temporizzato da un segnale CKP e resettati da un segnale RS: entrambi i segnali saranno esaminati in seguito.

Alcuni dei segnali che attivano una microistruzione corrispondono ad istruzioni del programma in corso nella CPU, altri derivano da richieste di operazioni di ingresso-uscita inviate alla CPU dalle periferiche; con riferimento alla figura sono originati dal programma della

CPU i segnali RW (trasferimento di un dato alla CPU), CW (trasferimento di un dato dalla CPU) e IOG (istruzioni di controllo per le interfacce), derivano da richieste inviate alla CPU i segnali DCY (primo ciclo di un DMA doppio), DI (DMA entrante), DO (DMA uscente) e INT (richiesta di interruzione).

Ricordiamo che con la dizione DMA (Direct Memory Access) si indica una procedura che consente il trasferimento rapido di dati tra la memoria centrale ed una periferica; con DMA doppio si indica un particolare DMA in cui più periferiche possono scrivere nella stessa area di memoria ed ogni parola è preceduta dall'indirizzo della periferica che l'ha generata.

Un esempio di realizzazione di un circuito facente parte del modulo ed atto a realizzare il DMA, compreso quello doppio, è descritto nella domanda di brevetto nr. 23659 A/80 del 24/7/1980.

Per una corretta esecuzione dei programmi è necessario sincronizzare l'inizio della microistruzione con il programma della CPU, ed arrestare se necessario la microistruzione in attesa di ricevere un consenso dalla CPU.

La sincronizzazione è ottenuta mediante il circuito CSL (logica di sincronizzazione del ciclo) che, tramite il segnale RS, resetta i registri R; l'arresto della microistruzione è realizzato mediante il circuito FL (logica di congelamento) che, tramite il segnale F, interdice il segnale CKP che temporizza i registri R.

Un esempio di realizzazione dei circuiti CSL e FL è riportato in figura 2.

Il circuito di controllo dei microprogrammi MPC, meglio illustrato nella figura 3, controlla che i microordini emessi da CS si succedano esattamente e che la microistruzione venga completata; l'eventuale errore viene segnalato (ME) alla CPU.

Come già detto in precedenza, la microistruzione di scansione IP emessa dalla memoria PS viene svolta da tutti i moduli di ingresso-uscita dell'elaboratore che non svolgono una delle microistruzioni di lavoro, emesso dalla memoria CS e indicate globalmente in figura con WP.

Come si può facilmente osservare, il contatore PC fa avanzare in ogni caso sia PS che CS: in particolare all'inizio di ogni ciclo PS emette il microordine CK che provoca il caricamento nei registri CC e RK dei segnali eventualmente presenti ai loro ingressi. CC riceve i segnali codificati da CE, RK abilita le microistruzioni di lavoro WP quando il modulo riconosce come proprio (ID) l'indirizzo associato ad un ordine IOG, RW o CW emesso dalla CPU, quando la CPU accetta (ASW) una richiesta trasmessa dal modulo oppure, nel secondo ciclo di un DMA doppio, in risposta ad un segnale DF emesso dal circuito DMA presente nel modulo.

Le uscite IP e WP sono ovviamente mutuamente esclusive poichè il modulo non può eseguire contemporaneamente una microistruzione di lavoro e una di scansione ciclica delle periferiche.

In figura 2 è illustrato un esempio di realizzazione della logica di sincronizzazione CSL e della logica di congelamento FL.

Il circuito CSL ha il compito di sincronizzare il modulo di ingres-

so-uscita con la CPU. Tale sincronizzazione assume particolare importanza quando le periferiche vengono gestite da due elaboratori operanti in sincronismo: i tempi di esecuzione dello stesso programma non sono perfettamente eguali per i due elaboratori ed è necessario garantire che l'inizio di un'operazione di ingresso-uscita avvenga contemporaneamente nei due elaboratori non appena entrambe le CPU sono in grado di gestirla.

La CPU genera sempre un segnale di temporizzazione MCK e, quando sta per finire di eseguire l'istruzione precedente, un segnale PR di durata prefissata, multipla intera del periodo del clock MCK: alla fine di PR la CPU è certamente disponibile per gestire la nuova istruzione.

Il bistabile 1, di tipo J-K, che costituisce il circuito CSL genera il segnale RS che ha durata pari ad un periodo di MCK e tiene resettati i registri R (figura 1). Si osservi che il bistabile 1 è temporizzato da MCK: i fronti di RS coincidono perciò con due transizioni eguali (ad esempio fronti di salita) successive di MCK; i registri R sono temporizzati da  $\overline{\text{MCK}}$  ed iniziano a caricare i microordini presenti all'uscita delle memorie mezzo periodo di MCK dopo la fine di PR. Si chiarisce meglio il significato del segnale PR: il fronte iniziale indica che alla CPU manca un tempo prefissato per terminare un ciclo ed il modulo di ingresso-uscita si prepara al ciclo successivo, resettando i registri R; il fronte finale indica che la CPU ha finito il ciclo ed il modulo ingresso-uscita è pronto per il ciclo successivo.

Due elaboratori operanti in sincronismo si scambiano informazioni, tra cui lo stato del segnale PR: in ogni elaboratore lo stato di avanzamento della CPU nello svolgimento del ciclo in corso di elaborazione determina il fronte iniziale del segnale PR, mentre ciascuna CPU fa terminare il proprio PR solo quando sa che l'altra CPU ha terminato il proprio ciclo, cosicchè i fronti finali di entrambi i segnali PR coincidono.

Si osservi che se la durata di PR è un multiplo pari, maggiore di due, del periodo di MCK, si ha una serie di impulsi RS ossia i registri R vengono resettati più volte.

Il circuito CSL e la struttura data alle memorie PC, CS e PS consentono di far partire i microprogrammi di ingresso-uscita sincroni con la CPU ed in un tempo molto limitato.

Quando il segnale RS resetta i registri R, viene contemporaneamente fornito l'indirizzo di partenza per la memoria PS, mentre la memoria CS riceve i dati meno significativi della microistruzione che dovrà eseguire.

Alla fine del segnale PR la CPU invia al codificatore CE il segnale indicante la nuova microistruzione e contemporaneamente PS emette il segnale di temporizzazione CK, la memoria CS ha l'indirizzo completo delle istruzioni di partenza, che raggiunge tramite un salto condizionato nel successivo periodo di MCK. L'unità di controllo del modulo di ingresso-uscita è perciò in grado di iniziare la nuova microistruzione con un ritardo, prefissato e rigorosamente rispettato, di due periodi di MCK rispetto all'istante in cui la CPU, esaur

rito il ciclo precedente e tolto PR, le ha comunicato qual'è la nuova microistruzione da eseguire.

Sempre in figura 2 è illustrato un esempio di realizzazione del circuito FL. Durante l'esecuzione di alcuni cicli può essere utile o necessario sincronizzare le operazioni di ingresso-uscita con l'evoluzione del programma nella CPU, arrestando il microprogramma di ingresso-uscita finchè non si verifica un evento prefissato, come la comparsa o la sparizione di un segnale inviato dalla CPU.

In figura si è supposto che una microistruzione possa prevedere solo l'attesa del segnale di fornire un dato alla CPU (segnale LIO) o la fine di tale segnale: ai due casi corrispondono i microordini FUL e FEL inseriti nella microistruzione svolta dall'unità di controllo del modulo di ingresso-uscita.

Il circuito FL comprende una pluralità di porte (21, 22), in numero pari a quello dei microordini di arresto, ciascuna delle quali riceve il microordine ed il segnale di cui si deve attendere la comparsa o la scomparsa, e le cui uscite, raccolte da un circuito sommatore 11, sono portate all'ingresso dati di un secondo bistabile 2 temporizzato dal clock MCK.

L'uscita F del bistabile 2 interdice il transito del clock MCK attraverso il NAND 12 (o un circuito porta): viene così a mancare il segnale CKP che fa avanzare i registri R ed evolvere le microistruzioni.

In particolare resta bloccato il contatore PC (la memoria CS resta indirizzata sulla cella contenente il microordine di arresto) e il registro R associato a CS: il microordine di arresto permane finchè non

cessa il segnale F.

Può succedere che il segnale atteso non possa presentarsi (ad esempio se la memoria è occupata la CPU non richiede al modulo di interfaccia di inviarle un dato): per impedire all'unità controllo modulo di bloccarsi in uno stato di stallo si prevede la presenza di un segnale SD che la CPU toglie quando si accorge di non poter inviare il segnale atteso.

Come già accennato con riferimento alla figura 1, le unità di controllo dei moduli di interfaccia eseguono una scansione ciclica delle interfacce pilotata dal microprogramma contenuto in PS e fatto avanzare da PC, che fa però avanzare anche CS: per evitare che un eventuale microordine di arresto blocchi PC, e quindi la scansione ciclica, le porte (21, 22) sono abilitate dal segnale RK presente nel solo modulo indirizzato dalla CPU.

Senza uscire dall'ambito del trovato è possibile prevedere la presenza di ulteriori microordini di interruzione oltre, o in sostituzione, a quelle (FUL, FEL) indicati in figura, adeguando di conseguenza il numero delle porte connesse agli ingressi del sommatore 11.

La figura 3 contiene un esempio di realizzazione del circuito di controllo MPC delle microistruzioni.

Tale circuito MPC comprende un circuito per il controllo della parità PA<sub>4</sub> ai cui ingressi sono applicati i bit dell'indirizzo, ricavati dal contatore PC e dal registro CC, ed il bit di parità PA dell'indirizzo successivo, contenuto nel microordine emesso dalla memoria CS.

Si ricorda che i registri associati a PC e CS sono fatti avanzare

dallo stesso clock CKP che, mentre carica nel registro R associato a CS una microistruzione, incrementa di uno l'indirizzo presente all'uscita di PC: l'indirizzo fornito da PC è quello della microistruzione successiva. Conoscendo la sequenza corretta dei microordini è possibile in sede di stesura di ciascuna microistruzione conferire al bit PA di ciascun microordine il valore richiesto.

Per l'ultimo microordine EOP di ciascuna microistruzione non è possibile prevedere quale sarà la microistruzione successiva e perciò su quale indirizzo basarsi per calcolare il bit PA.

Per ovviare a tale ambiguità sono possibili molte soluzioni tra cui, ad esempio:

- inibire con l'ultimo microordine EOP il circuito  $PA_4$  o l'invio dell'eventuale segnale di errore  $S_4$  alla CPU;

- utilizzando microistruzioni di lunghezza costante, realizzare il contatore PC in modo che, esaurita la sua capacità di conteggio ossia finita la microistruzione, permanga nello stato raggiunto: il bit PA dell'ultimo microordine EOP viene calcolato in base all'indirizzo del microordine stesso.

Se un ciclo giunge regolarmente a termine, all'arrivo del segnale PR (che, ricordiamo, indica che la CPU sta finendo il proprio ciclo) nel modulo deve essere attivo l'ultimo microordine EOP: l'assenza di EOP, evidenziata ad esempio mediante il bistabile 3, viene segnalata alla CPU ad esempio mediante il sommatore 14 che raccoglie anche il segnale errore  $S_4$  per formare il segnale ME inviato alla CPU.

Il segnale  $C_4$  consente su comando software di alterare il bit di pa-

rità PA in modo da far generare il segnale errore  $S_4$ ; ciò consente alla CPU di verificare il corretto funzionamento del circuito  $PA_4$ .

===

===

| ===

===

===

===

===

===

===

===

===

===

===

===

===

===

===

===

===

===

===

===

===

### RIVENDICAZIONI

1) Unità di controllo di un modulo ingresso-uscita di un elaboratore elettronico dalla cui unità logica centrale, nel seguito detta CPU, riceve ordini e segnali di temporizzazione, caratterizzata dal fatto di comprendere in combinazione tra loro:

- una unità di codifica (CE) che, in risposta ad una richiesta di esecuzione di una microistruzione, genera un codice che viene scritto in un primo registro (CC);

- un contatore (PC) atto a scandire i microordini che costituiscono una microistruzione;

- una prima memoria (CS), indirizzata dal codice scritto nel primo registro (CC) e dal contatore (PC), contenente le microistruzioni operative;

- una seconda memoria (PS) indirizzata dal contatore (PC) e contenente le microistruzioni di scansione ciclica delle unità di interfaccia facenti parte del modulo;

- un primo circuito (CSL) atto a sincronizzare l'inizio delle microistruzioni eseguite dall'unità di controllo con quelle della CPU;

- un secondo circuito (FL) atto ad arrestare una microistruzione in corso di svolgimento nell'unità di controllo in attesa della comparsa o scomparsa di almeno un segnale emesso dalla CPU;

- un terzo circuito (MPC) atto a controllare la corretta successione dei microordini nell'ambito di una microistruzione ed il completamento della microistruzione stessa;

- mezzi (RKEN) atti ad abilitare l'esecuzione di una microistru

zione operativa e della microistruzione di scansione ciclica.

2) Unità di controllo come alla rivendicazione 1 caratterizzata dal fatto che la seconda memoria (PS) genera all'inizio di ciascun ciclo un primo segnale (CK) che abilita detti mezzi (RKEN) e comanda la scrittura nel primo registro (CC) del codice generato dal codificatore (CE).

3) Unità di controllo come alla rivendicazione 1 caratterizzata dal fatto che all'uscita del contatore (PC) e della prima e seconda memoria (CS, PS) sono presenti registri (R) resettati da un secondo segnale (RS) generato dal primo circuito e fatti avanzare da un terzo segnale (CKP) di temporizzazione ricavato mediante un quarto segnale (P), generato dal secondo circuito (FL), da un quinto segnale (MCK) di temporizzazione generato dalla CPU; caratterizzato inoltre dal fatto che anche il contatore (PC) è costituito da una memoria indirizzata insieme alla prima e seconda memoria (CS, PS), dall'uscita del registro (R) associato al contatore (PC).

4) Unità di controllo come alle rivendicazioni 1 e 3 caratterizzata dal fatto che il primo circuito (CSL) comprende un primo bistabile (1) di tipo J-K, temporizzato dal quinto segnale (MCK), ai cui ingressi di controllo (J-K) è applicato un sesto segnale (PR) generato dalla CPU quando manca un tempo prefissato alla fine del ciclo precedente e tolto alla fine del ciclo stesso, l'uscita del primo bistabile (1) costituendo il secondo segnale (RS).

5) Unità di controllo come alle rivendicazioni 1 e 3 caratterizzata dal fatto che il secondo circuito (FL) comprende:

- una pluralità di porte (21, 22), in numero pari a quello dei microordini (FUL, FEL) di arresto presenti nelle microistruzioni contenute nella prima memoria (CS), ciascuna porta ricevendo ad un ingresso il microordine di attendere l'arrivo o la cessazione di un segnale e sull'altro ingresso il segnale;

- un sommatore (11) che raccoglie le uscite delle porte ed ha l'uscita collegata all'ingresso dati di un secondo bistabile (2);

- un secondo bistabile (2), temporizzato dal quinto segnale (MCK), la cui uscita costituisce il quarto segnale (F).

6) Unità di controllo come alla rivendicazione 5 caratterizzata dal fatto che tutte le porte associate ad un microordine di attendere l'arrivo di un segnale ricevono su un ulteriore ingresso un settimo segnale (SD) inviato dalla CPU in sostituzione del segnale atteso quando, per l'interruzione del ciclo di elaborazione, tale segnale atteso non può essere generato.

7) Unità di controllo come alle rivendicazioni 1 e 3 caratterizzata dal fatto che un bit ( $PA$ ) prefissato di ciascun microordine costituisce il bit di parità dell'indirizzo del microordine successivo e dal fatto che il terzo circuito (PCM) comprende un circuito per il controllo della parità ( $PA_4$ ) ai cui ingressi sono applicati i bit di indirizzo presenti all'uscita del primo registro (CC) e del registro (R) associato al contatore (PC) nonchè il bit di parità ( $PA$ ), ed un terzo bistabile (3) temporizzato dal sesto segnale (PR) ed al cui ingresso dati, negato, è applicato l'ultimo microordine (EOP) della microistruzione, le uscite del circuito ( $PA_4$ ) per il controllo di parità e

del terzo bistabile (3) essendo sommate per generare un segnale di al  
larne (ME) per la CPU.

8) Unità di controllo come alle rivendicazioni 1 e 7 caratterizza  
ta dal fatto che il contatore (PC) esaurita la sua capacità di conteg  
gio, permane nello stato raggiunto e dal fatto che il valore del bit  
di parità (PA) dell'ultimo microordine (EOP) coincide con quello del  
microordine precedente.

9) Unità di controllo come alla rivendicazione 1 caratterizzata  
dal fatto che i mezzi (RKEN) di abilitazione comprendono un secondo  
registro (RK), temporizzato dal primo segnale (CK), che attiva la pro  
pria uscita quando il modulo riconosce come proprio l'indirizzo asso  
ciato ad un messaggio inviato dalla CPU o in risposta ad un segnale  
generato da un circuito facente parte del modulo ed atto a gestire un  
DMA doppio, il livello logico dell'uscita del secondo registro (RK) de  
terminando l'invio ai circuiti di interfaccia di una microistruzione  
operativa (WP) o di quello (IP) di scansione ciclica.

10) Unità di controllo come alle rivendicazioni 5 e 9 caratterizza  
ta dal fatto che tutte le porte (21, 22) del secondo circuito (FL) so  
no abilitate dal segnale di uscita del secondo registro (RK).

11) Unità di controllo secondo quanto illustrato nella precedente  
descrizione e nei disegni allegati come pure qualsiasi parte di essa i  
solatamente o in combinazione.

BREV/CGG/pel



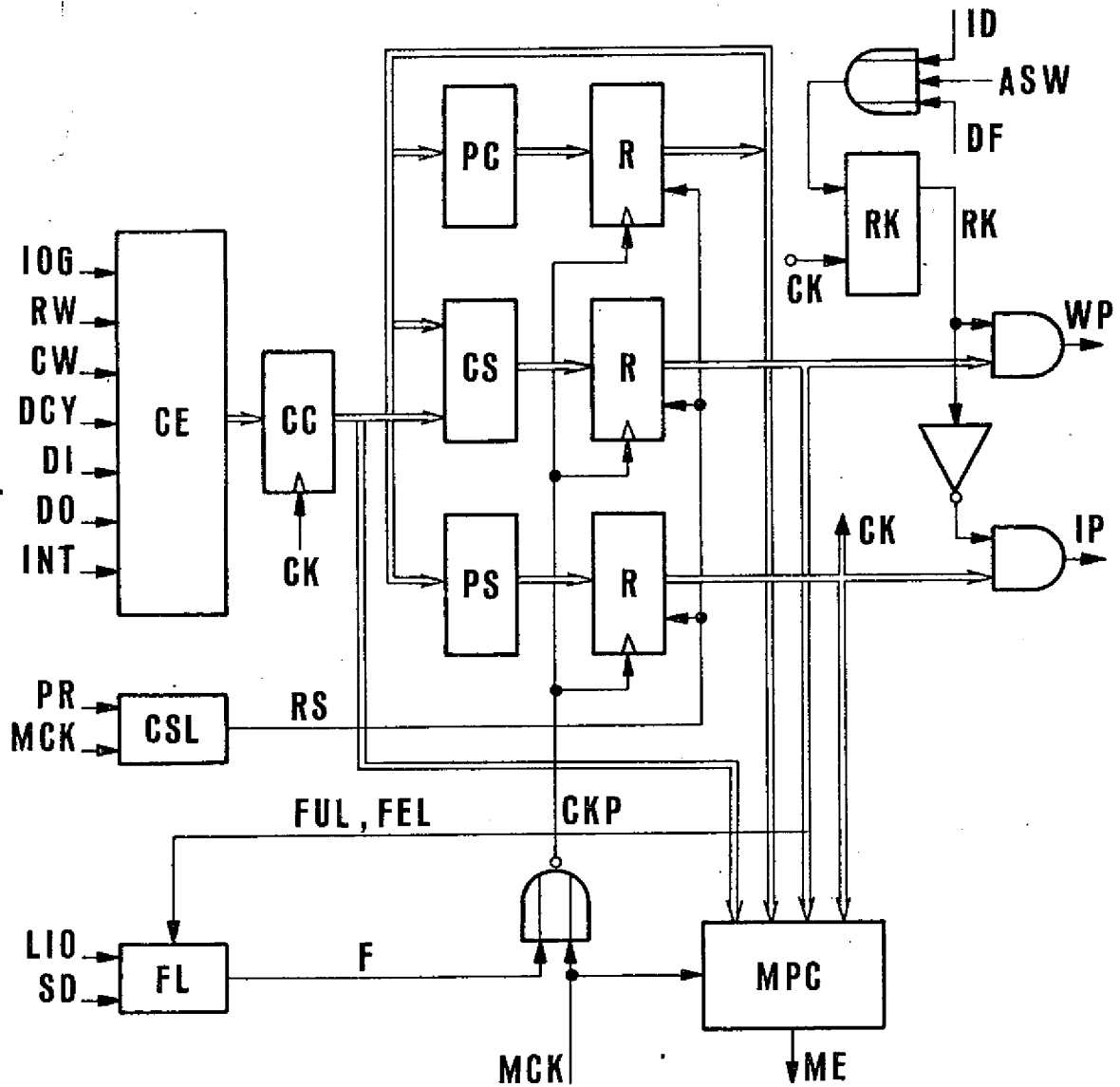
*L'Ufficiale Rogante*  
*(Idillia Russo)*

SOCIETA' ITALIANA TELECOMUNICAZIONI

SIEMENS s.p.a.

*per incarico speciale*

24 40 6 A/80



Ufficiale Rogante  
(Idillio Rosso)

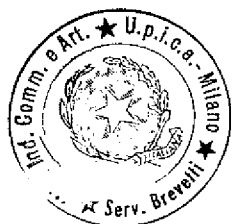
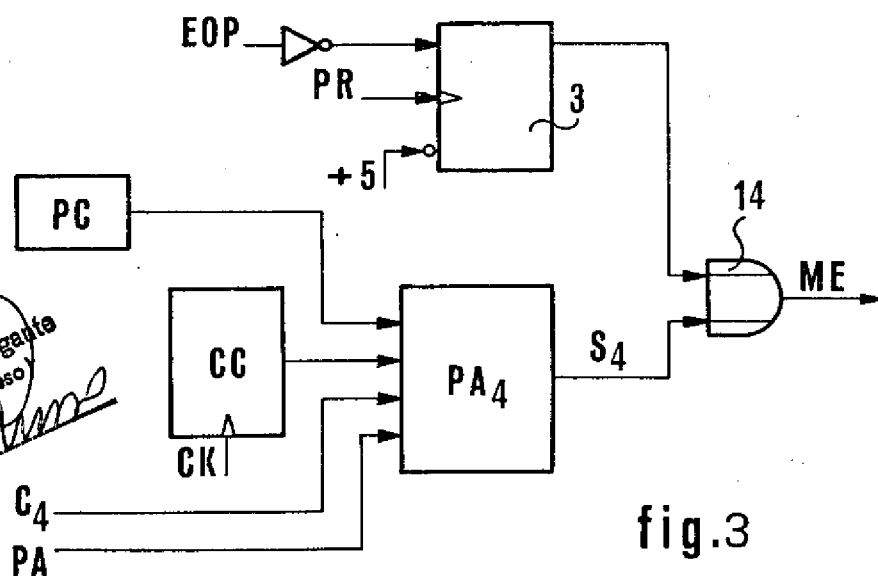
fig.1

SOCIETA ITALIANA TELECOMUNICAZIONI  
SIEMENS s.p.a.

per incarico speciale

*[Signature]*

*[Signature]*



l'Ufficiale Rogante  
Tatiana Russo

fig.3

**SOCIETÀ ITALIANA TELECOMUNICAZIONI**

**SIEMENS s.p.a.**

*per incarico speciale*