



19 ES	11 NUMERO	10 A 1
	21 434.283	
	22 FECHA DE PRESENTACION	

PATENTE DE INVENCION

50 PRIORIDADES: 31 NUMERO	32 FECHA	33 PAIS
447.040	28 de febrero de 1.974	EE.UU. de A.

47 FECHA DE PUBLICIDAD	51 CLASIFICACION INTERNACIONAL	62 PATENTE DE LA QUE ES DIVISIONARIA
	G06F	

64 TITULO DE LA INVENCION
PROCEDIMIENTO Y SISTEMA DE PROCESO DE DATOS MEDIANTE UNA RED ACTIVA DA POR DATOS.

71 SOLICITANTE (S)
BURROUGHS CORPORATION
DOMICILIO DEL SOLICITANTE
Burroughs Place, Detroit, Michigan 48232, EE.UU. de A.
72 INVENTOR (ES)
Robert Stanley Barton, Ing., Alan Lynn Davis., Ing., Erwin Arthur Hauck, Ing., Don Martin Lyle, Ing., Lloyd Drayton Turner, Ing
73 TITULAR (ES)
74 REPRESENTANTE
D. JAIME GOMEZ-ACEBO Y MODET

PATENTE DE INVENCION

USSN. 447.040

## *Memoria Descriptiva*

*sobre:*

PROCEDIMIENTO Y SISTEMA DE PROCESO DE DATOS  
MEDIANTE UNA RED ACTIVADA POR DATOS.

-----

*Solicitante:* BURROUGHS CORPORATION, entidad norteamericana,  
residente en Burroughs Place, Detroit, Michigan  
.48232, EE.UU. de A.

-----

La presente invención se refiere a un sistema para procesar datos simultáneamente y en canalizaciones mediante una red activa y, de un modo más específico, se refiere a una red que comprende elementos de procesamiento distribuidos y también elementos de memoria distri-

buidos.

5. Los sistemas de proceso de datos de la tecnología anterior se han diseñado en general con un alto grado de centralización de control del sistema que residía en dispositivos tales como el elaborador central y las unidades de memoria principales. Este alto grado de centralización era el resultado de los costes de los elementos, tamaños de los componentes y los intentos realizados para que la utilización de los componentes fuera óptima. A su vez, se orientaron los algoritmos de operaciones del sistema hacia el control centralizado, en lugar del control descentralizado, y los algoritmos y lenguajes se orientaron hacia una ejecución de tareas en secuencia.

10. En los sistemas centralizados, se dedicaron dispositivos particulares a funciones particulares y la división del sistema entre procesos diferentes era, en el mejor de los casos, irregular. Se idearon grandes sistemas operacionales para que la utilización de los diversos componentes del sistema a dispositivos fuera óptima con el fin de compartirlos a través de un cierto número de programas y tareas. Los sistemas altamente

15. centralizados no aprovechan totalmente la economía que se consigue ahora con el empleo de bloques de circuito integrados, modulares, a gran escala (LSI)

20.

25. Para dar una mayor flexibilidad a la utilización de procesos, programas y tareas simultáneas, los sistemas centralizados a la tecnología anterior han evolucionado hacia sistemas de procesos múltiples y también paralelos compuestos por redes y formaciones de unidades que alcanzan un cierto grado de modularidad. O sea, las capacidades de elaboración pueden aumentarse añadiendo unidades de elaboración centrales

30. adicionales, la memoria principal puede aumentar añadiendo

un número adicional de módulos de memoria y las capacidades de I/O pueden aumentar añadiendo un número adicional de canales de I/O o unidades de control. No obstante, las unidades de elaboración individual y las unidades de control I/O están orientadas todavía hacia una ejecución de tareas en secuencia, el control del sistema está todavía relativamente centralizado, y la división de las diferentes unidades del sistema para manejar simultáneamente diferentes tareas es todavía relativamente irregular.

- 5.
10. Los desarrollos en la circuitería de LSI y en otra tecnología de los componentes han eliminado algunas de las influencias que dieron por resultado el diseño de un control altamente centralizado en los sistemas de ordenadores electrónicos. La tecnología de LSI exige regularidad y no una dedicación de algoritmos especializados o complejos y funciones a los bloquitos de los circuitos. Adicionalmente, las memorias de circuitos integrados son compatibles en interfase con los circuitos lógicos y, por lo tanto, la arquitectura del elaborador orientada hacia registradores puede eliminarse distribuyendo memorias de circuito integrado por todo el sistema.
15. Además, las tendencias del desarrollo en los dispositivos de almacenamiento periódico tales como discos, paquetes de discos, líneas de retardo, y similares, han ofrecido una forma muy barata de memoria. A pesar de que los tiempos de acceso de las memorias de tipo periódico o dinámico son relativamente lentos si se compara con las memorias de circuito integrado, la distribución de las memorias periódicas por todo el sistema ofrece la ventaja de una cantidad virtualmente infinita de almacenamiento, por lo que el tiempo de acceso
20. no es mayor que el tiempo de acceso de una memoria particular.
- 25.
- 30.

5. Para proporcionar modularidad de unidades llevadas a la práctica en circuitos integrados, el empleo de la microprogramación permite que el sistema puede fabricarse de un gran número de bloquitos elaboradores similares, dada uno de los cuales se dedica a una tarea particular por el microprograma almacenado en su memoria de microprograma. De éste modo, un cierto número de elaboradores similares llevados a la práctica en bloquitos de circuitos integrados puede dedicarse, respectivamente, a operaciones aritméticas, operaciones de entrada/salida, y similares.

10. Una característica arquitectónica particular que ha influido en el diseño de sistemas altamente centralizados ha sido el empleo de trayectos de datos paralelos de bitios y el acceso paralelo de bitios de la memoria. La transferencia de datos paralelos ha exigido una cierta amplitud uniforme de los segmentos de datos que, a su vez, no se acomoda fácilmente a las diferentes aplicaciones del utilizador. No obstante, el empleo de una transferencia de datos serial por bitios o serial por caracteres se acomoda fácilmente por segmentos de datos de cualquier longitud y evita las exigencias de que los trayectos de datos, registradores y lugares de memoria se instalen manteniendo uniformidad unos con otros.

15. Quizá la característica única más significativa de los sistemas de la tecnología anterior, que dió por resultado el diseño de un control del sistema centralizado y una ejecución de tarea-en secuencia, ha sido la de emplear ejecución de instrucciones activas por control. O sea, en la mayoría de los sistemas de la tecnología anterior, una cadena de instrucciones que representa un sistema, o procedimiento dentro de un programa, se han sacado por el aumento en secuencia de un

20.

25.

30.

5. contador de programa, y los operandos requeridos por la cadena de instrucciones deben disponerse para que lleguen al elaborador a tiempo de la ejecución de las instrucciones particulares en dicha cadena. Como la presencia de los operandos necesarios no ha sido el acontecimiento que pide ejecución de instrucciones, se ha creado un cierto número de métodos complicados para asegurar y la presencia de los operandos necesarios.

10. Contrariando con los procedimientos activos por control, los sistemas activados por datos son aquellos que se caracterizan porque las instrucciones particulares se secan para la ejecución en respuesta a la llegada de segmentos de datos respectivos en una unidad de elaboración particular. En los procedimientos activados por datos, no es necesario un contador de programa para el control del sistema y no hay necesidad de centralización de control del sistema.

15. En el sistema llevado por datos, las diversas instrucciones que componen un programa pueden asignarse a diferentes elaboradores y su ejecución tendría lugar en secuencia a medida que el dato respectivo se traslada de un elaborador a otro. De esta manera, el procedimiento puede dividirse fácilmente para servir para procedimientos completamente simultáneos o no relacionados. Por otro lado, cuando ha de ejecutarse un cierto número de acciones de elaboración en segmentos de datos semejantes, la corriente de la segmentos de datos semejantes puede transferirse desde un elaborador a otro para conseguir un efecto de canalización limitado solamente por los ritmos de transferencia desde un elaborador a otro.

25.

30. Por consiguiente, un objeto del presente invento es proporcionar un sistema de proceso de información con una or-

ganización más natural para la operación que la que existe en procedimientos activados por control de la tecnología anterior.

5. Otro objeto del presente invento es proporcionar un sistema de proceso de información que tiene unidades de distribución o de función y unidades de memoria, cuyo procedimientos no exige un control de sistema centralizado.

10. Otro objeto del presente invento es proporcionar un sistema de proceso de información perfeccionado que se puede dividir dinámicamente para servir para procedimientos, programas y tareas simultáneo.

15. Otro objeto del presente invento es proporcionar un sistema de proceso de información perfeccionado que comprende cualquier número de unidades de funciones similares que entonces pueden tener asignadas tareas diferentes que han de ser realizadas por el procedimiento.

20. Otro objeto adicional del presente invento es proporcionar un sistema de proceso de información perfeccionado que tiene una memoria distributiva la cual puede aumentar sin límite y sin aumentar el tiempo de acceso de la misma.

25. Con el fin de conseguir los objetos arriba descritos, el presente invento está dirigido hacia una red de módulos de procesado o funciones, cada uno de los cuales está provisto de su propio almacenamiento local, estando prevista la función de almacenamiento de la memoria principal por la pluralidad de almacenamientos locales diferentes. La transferencia de datos entre diversos módulos de función es asincrónica de una manera serial de bits o serial de caracteres. La ejecución por parte de cada uno de los módulos de funciones se inicia por la llegada de todos los componen

30.

tes de los datos requeridos por uno de los nodos de una red de programa contenida en la unidad de almacenamiento local acoplada al módulo de funciones. Los componentes de los datos requeridos respectivos forman estructuras de datos asociados.

5. Una de estas estructuras de datos es un nodo de instrucción u operador (operando) y una pluralidad de los nodos se almacena en uno o más de los almacenamientos locales diferentes para formar una red de programa. La otra estructura de datos comprende componentes de datos transistorios que proporcionan
10. conexión entre los nodos de la red del programa. Los componentes de datos transistorios comprenden por lo menos un operando (operador) y una dirección que apunta hacia el lugar que ocupa en un almacenamiento local el nodo de instrucción u operador (operando) que se ha de ejecutar. La operación
15. se acomoda a ambos nodos de operación monádica y diádica. Los nodos operadores están provistos de una dirección de destino a la cual se han de transmitir los resultados de la operación. El módulo de función se activa entonces por los datos, por lo que no existe necesidad de un gran sistema operatorio
20. que proporcione una programación de tareas que interrumpa la resolución de prioridades.

Con el fin de adaptar la transmisión de datos simultáneos entre módulos de funciones, los conductores de entrada y de salida de una pluralidad de los módulos de funciones

25. se acoplan a un intercambiador y cada uno de los intercambiadores puede acoplarse a un intercambiador de nivel superior de una forma recursivamente jerárquica para adaptar cualquier número de los módulos de funciones y almacenamientos locales asociados como pueda ser necesario. Algunos de los módulos de funciones están provistos de tipos diferentes de interfaces I/O y los módulos de funciones respectivas están desti-
- 30.

nados a microprogramarse de forma que se puedan dedicar módulos de funciones diferentes a tareas de operaciones particulares tales como control de I/O y similares.

5. Por lo tanto, una característica del presente invento consiste en un procedimiento de elaboración de información formado por una red que comprende una pluralidad de módulos de funciones y una pluralidad de unidades de almacenamiento locales, acoplándose cada unidad de almacenamiento a un módulo de funciones diferentes para cooperar con el mismo. Cada uno de los módulos de funciones es microprogramable y las ejecuciones realizadas en el mismo se hacen en respuesta a un par de estructuras de datos asociados, una de las cuales es un nodo operador (operando), normalmente almacenado en la unidad de almacenamiento local y el otro que comprende una parte de dirección y una parte de argumento, que son partes de datos transitorios que conectan los nodos de una red de programa. La parte de dirección o localización apunta al lugar en una unidad de almacenamiento local del operador requerido (operando).

10. Otra característica del presente invento consiste en la red de módulos de funciones y unidades de almacenamiento local comprende una pluralidad de unidades intercambiadoras, una por cada grupo de las unidades de funciones, acoplándose cada una de las unidades de intercambio a otra del nivel siguiente de unidades intercambiadoras de una manera o jerárquica para adaptar cualquier número de unidades de funciones que pudiera ser necesario.

15. Otra característica del presente invento consiste en el procedimiento de elaboración de información que se caracteriza porque las unidades de almacenamiento local son cíclicamente

cas y porque la transferencia de información entre unidades de funciones es asincfona y de una manera serial por bitios o serial por caracteres.

5. Los objetos citados y otros objetos, ventajas y características del presente invento, resultarán evidentes por la descripción detallada que sigue tomando como referencia los dibujos adjuntos, en los que:

La figura 1 es un diagrama esquemático de la red activada por datos del presente invento.

10. La figura 2 es una tabla que representa el juego de caracteres empleados por un módulo de funciones del presente invento.

15. Las figuras 3A y 3B son representaciones de los formatos de estructuras de datos empleados en el presente invento.

La figura 4 es un diagrama de una red de programa que ilustra el funcionamiento del procedimiento en el presente invento.

20. La figura 5 es un diagrama esquemático del módulo de funciones empleado en el presente invento.

La figura 6 es un diagrama esquemático de las filas de esferas de entrada según se ilustran en la figura 5.

La figura 7 es un diagrama esquemático de la unidad lógico según se ilustran en la figura 5.

25. La figura 8 es un diagrama esquemático de la fila de espera de salida según se ilustra en la figura 5.

La figura 9 es un diagrama esquemático del adaptar de interfase de almacenamiento según se ilustra en la figura 5.

30. La figura 10 es un diagrama esquemático de la unidad

de control se ilustra en la figura 5, y

La figura 11 es un diagrama esquemático de un intercambiador según se ilustra en la figura 1.

5. El sistema del presente invento es aquel que aprovecha totalmente la tecnología de LSI para proporcionar un procedimiento flexible que, a su vez, puede dividirse dinámicamente según diversas exigencias simultáneas. El presente invento excluye específicamente el bastidor principal de elaboración central clásico, la memoria principal y las unidades de control de I/O. El procedimiento comprende una red de módulos de funciones cada uno con su propia unidad de almacenamiento local. La función de la memoria principal clásica se obtiene por la suma total de las diversas unidades de almacenamiento locales. Algunos de los módulos de funciones pueden dedicarse a funciones de I/O por instrucciones particulares almacenadas en sus unidades de almacenamiento local y por adaptadores de interfaces acoplados a las unidades de funciones.

10. El sistema se dice que es activado por datos en el sentido de que las instrucciones particulares de un programa se almacenan en las unidades de almacenamiento locales de diferentes módulos de funciones para establecer nodos en una red de programa, cuyas instrucciones se secan cuando el dato apropiado ha llegado a la unidad de funciones particulares. Así, las instrucciones de un programa particular se distribuyen por todo un número de elegidos de módulos de funciones cuyas instrucciones son ejecutadas en respuesta a la llegada del dato asociado, en lugar de ser ejecutadas las instrucciones de una manera simplemente en secuencia bajo el control de programa como es clásico en procedimientos

15.

20.

25.

30.

centralizados en la tecnología anterior.

Un sistema que emplea el presente invento se ilustra en la figura 1. Este sistema es una formación de módulos de funciones 10, cuya transferencia de datos interna es sincrónica. No obstante, la transferencia de datos entre módulos es asincrónica. Según se describirá con más detalle más adelante, la transferencia de datos puede considerarse como serial por bitios; no obstante, debido al juego de caracteres de dos bitios empleados, la transferencia de datos es realmente serial por caracteres.

Con cada módulo de funciones 10 se asocia una unidad de almacenamiento local 11, cuyas unidades de almacenamiento local son normalmente cíclicas por naturaleza para adaptar fácilmente la transferencia de datos serial por bitios. Las diversas unidades de almacenamiento y tiempo de acceso para adaptar las exigencias a diferentes programas o tareas. Así, dichas unidades de almacenamiento locales pueden componerse de una pluralidad de memorias diferentes, por ejemplo archivos de disco, paquetes de discos, líneas de retardo, dispositivos acoplados en carga, memorias de burbujas, y similares. No obstante, en ciertas situaciones, las unidades de almacenamiento local 11 serían memorias de acceso casual que podrían estar formadas por núcleo o circuitos integrados.

Para adaptar la transferencia de datos entre módulos de funciones, la interfase del módulo de funciones comprenden conductores de salida 10a y conductores de entrada 10b que acoplan los módulos de funciones respectivas 10 a un intercambiador de primer nivel 12. El intercambiador 12 es un sistema de conmutación que se describirá con más deta-

- lle más adelante. Si el número de módulos de funciones exigido en un sistema particular que excede del número para el que puede servir un intercambiador particular, entonces se puede acoplar una pluralidad de los intercambiadores de primer nivel 12 a un intercambiador de segundo nivel 16 por medio de una interfase de intercambiador 13 asociada con cada intercambiador de primer nivel. Con el fin de tamponar la transferencia de datos, la interfase de intercambiador 13 está provista de una fila de espera de salida 14 y una fila de espera de entrada 15. Aún cuando no se ilustra de una forma explícita en la figura 1, una pluralidad de los intercambiadores de segundo nivel 16 puede acoplarse entonces a un intercambiador de nivel superior siguiente por medio de una interfase de intercambiador 13, y así sucesivamente. Por lo tanto, se puede adaptar cualquier número de módulos de funciones por una jerarquía de intercambiadores encajados. Los intercambiadores de diversos niveles 16 son similares al intercambiador del primer nivel 12. Se comprenderá que si el número de módulos de funciones requerido puede adaptarse a dos intercambiadores de primer nivel 12, los dos intercambiadores pueden interconectarse por medio de una sola interfase de intercambiador 13 donde la fila de espera de salida 14 de un intercambiador pasa a ser la fila de espera de entrada 15 para el otro intercambiador. Según se ilustra adicionalmente la figura 1, ciertos módulos de funciones 10 pueden conectarse a una vía común 10c.

Según se ilustra en la figura 1, la jerarquía de los intercambiadores encajados es recursiva por naturaleza, o sea, cualquier módulo de función 10 puede ser reemplazado por una interfase de intercambiador 13 y un intercambiador 12 que se acopla a una pluralidad de módulos de funciones 10.

y viciversa. Se observará en la figura 1 que dos intercambiadores 12 se acoplan al intercambiador 16 por medio de interfaces de intercambiador 13 y, igualmente, un módulo de funciones 10 se acopla al intercambiador 16 por medio de vías 10d y 10e.

5.

Para adaptar la transferencia de datos y otras comunicaciones con dispositivos fuera del sistema, uno o más de los módulos de funciones está provisto de un adaptador de I/O 17 teniendo lugar la transferencia de datos bajo el control de instrucciones particulares almacenadas en el almacenamiento local del módulo de funciones correspondientes. Lógicamente, si los dispositivos periféricos externos al sistema contienen sus propias unidades de control, entonces aquellos dispositivos puede acoplarse directamente a un intercambiador de primer nivel de la misma manera que los módulos de funciones asociados con el adaptador de I/O 17.

10.

15.

Antes de describir el funcionamiento del sistema y la forma nodal de redes de programa, se expondrá una descripción del formato de instrucciones que comprenden un mecanismo de campo definible. Es decir, los campos de datos e instrucciones son de longitud variable. Se recordará que la transferencia de información puede considerarse como serial por bitios y, por lo tanto, no se imponen restricciones respecto a la longitud de segmentos de información como podría ocurrir en maquinas con anchuras de trayectos de datos fijos. Las estructuras de información están compuestas por una serie de caracteres que se ilustran en la figura 2. Cada caracter comprende dos bitios que permiten cuatro caracteres diferentes que, según se ilustran en la figura 2, son (,), 0 y 1. En toda esta descripción, los paréntisis puedan considerarse como corchetes y comprenderá que ambos términos

20.

25.

30.

se pueden intercambiar.

5. El formato de la estructura de datos formado a partir de éste juego de caracteres se describe ahora con relación a la figura 3a y la figura 3b. Estos formatos de estructura definen una serie de lo que se conoce generalmente como células de datos. Ciertas reglas de construcción tienen también aplicación para interpretación de dicho formato. Por ejemplo, el par adyacente de paréntesis de apertura o corchetes indica el comienzo de una estructura de datos y el par adyacente de paréntesis de cierre indica el final de la estructura. Los diversos campos dentro de la estructura de datos según definidos por los pares de paréntesis de apertura y cierre o corchetes. De este modo, la estructura así como los diversos campos dentro de las estructuras pueden tener cualquier longitud. Por definición, no pueden existir datos entre pares de paréntesis o corchetes iguales.
- 10.
- 15.

- Según se ilustra en las figuras 3a y 3b, el contenido interior de un par de paréntesis o corchetes de cierre representa datos mientras que el contenido exterior o información que precede a una expresión de paréntesis es una indicación que define la significación del contenido interior de dicha expresión. Un juego representativo de indicaciones pueden definirse como sigue. 0 indica que el contenido interior siguiente constituye una dirección de módulos de funciones; 1 indica que el contenido interior siguiente constituye una dirección de almacenamiento local; 2 indica que el contenido interior siguiente es un calificador de dirección de índice; 3 indica que el contenido interior siguiente es un calificador de dirección de nombre; 4 indica que el contenido interior siguiente es un modificador de operando; 5 indica que el contenido interior siguiente es un código de operador o
- 20.
- 25.
- 30.

designador del tipo de célula; 6 indica que el contenido interior siguiente es una dirección de destino; y 7 indica que el contenido interior siguiente es un campo de operando. Un ejemplo más detallado de las indicaciones que se pueden emplear con el presente invento se expondrá más adelante. También se expondrá más adelante una explicación más detallada de las figuras 3a y 3b.

5.

10.

15.

Una característica particular del juego de caracteres según se ilustra en la figura 2, es la comprobación de paridad que se proporciona. Se observará que por cada paréntesis o carácter de corchete en una estructura o campo, habrá un carácter de paréntesis o corchete opuesto, comprendido cada carácter un solo bitio uno. Además, los caracteres de datos n o tendrán bitios uno o tendrán dos bitios uno. Por lo tanto, todas las estructuras, o campos dentro de una estructura, tendrán un número par de bitios uno si la instrucción se ha codificado apropiadamente. Esto proporciona un esquema de comprobación de paridad par.

20.

25.

Según se ha indicado anteriormente, el procedimiento del presente invento se activa por datos. En lugar de ser sacado por un contador de programa, un operador es sacado y ejecutado solamente cuando el dato apropiado ha llegado al módulo de funciones. Si la instrucción pide la combinación de dos componentes de datos, por ejemplo, A y B la instrucción será ejecutada solamente cuando ambos componentes de dato hayan llegado al módulo de funciones cualquiera que sea el orden en que se hayan recibido. La instrucción no será sacada para ejecución hasta que ambos componentes de datos hayan llegado al módulo de funciones.

30.

Con el procedimiento activado por datos, el programa

5. puede representarse entonces como una red de programa donde las instrucciones individuales del programa constituyen nodos en dicha red. Una red de programa que sirve de ejemplo se ilustra en la figura 4. El programa representado por esta red está destinado a calcular los valores de X y Z, de acuerdo con las ecuaciones algebraicas siguientes:

$$X = ((A + B(C - D)) / (E + F))$$

$$Y = X + G$$

$$Z = X \cdot X$$

10. En una red de programa estructurado en flujograma de la figura 4, el nodo N0 representa la operación A+B, el nodo N1 representa la operación C-D, el nodo N2 representa la operación E + F, N3 representa la multiplicación de A+B y C-D, el nodo N4 representa la división de producto de dicha multiplicación por E+F, el nodo N5 representa la multiplicación de X multiplicado por X, y el nodo N6 representa la adición de X y G.

15. Tomadas como enunciados FORTRAN, las ecuaciones anteriores se agrupan en 23 instrucciones de lenguaje de ordenador grande secuencial de la tecnología anterior. Ahora, si un módulo de funciones del procedimiento, en el presente invento se emplea para ejecutar los cálculos requeridos, y si el tiempo de transición a través de un nodo se considera una unidad de tiempo, el tiempo de ejecución de las ecuaciones anteriores sería de 7 unidades de tiempo. Esto se supone que todas las instrucciones necesarias para las operaciones respectivas, o sea los 7 nodos de programa, se almacenen en la unidad de almacenamiento local del módulo de funciones y que todas las entradas estén disponibles.

20. Se observará por el programa estructurado en árbol de

25. 30.

la figura 4 que los nodos N0, N1 y N2 se pueden ejecutar simultáneamente o en secuencia en cualquier orden y, por lo tanto, son simultáneos. De un modo similar, los nodos N5 y N6 son simultáneos. Se dice que dos tareas son simultáneas si, y solamente si, se pueden ejecutar simultáneamente o en secuencia en cualquier orden sin cambiar el resultado final.

5.

Si se emplea una formación de módulos de funciones como la ilustrada en la figura 1, para calcular las expresiones anteriores, se necesitarían tres módulos de funciones con las instrucciones que representan los nodos N0, N1 y N2, residiendo en módulos diferentes de funciones y los nodos N5 y N6 residiendo en módulos diferentes. Según se comprenderá por la red de programa de la figura 4, el tiempo de ejecución para el cálculo de las ecuaciones algebraicas será de cuatro unidades de tiempo. El empleo de más de tres módulos de funciones no produciría efecto alguno sobre la velocidad de este algoritmo.

10.

15.

20.

25.

30.

Además, se observará por la red de programa de la figura 4, que se realizan operaciones diferentes en A y B cuando dicho dato pasa del nodo N0 al nodo N3 al nodo N4 y así sucesivamente. Por lo tanto, se produce un efecto de canalización de clases, en las operaciones realizadas sobre los datos según se transfieren los datos de nodo a nodo o de módulos de funciones a módulos de funciones. El grado, con que se consigue este efecto de canalización depende, como es lógico, del ritmo de transferencia entre módulos de funciones. Se comprenderá que la red del programa presentada en la figura 4, puede expandirse para incluir tantos nodos como sean necesarios para representar cualquier programa dado o grupos de programas. O sea, el sistema se puede expandir para adaptar

fácilmente la multiprogramación que, en este sistema, comprende incidentalmente multielaboración.

5. En lo que se ha descrito, el sistema presente comprende una formación de módulos de funciones que se pueden programar para servir para una pluralidad de redes de programas diferentes. Según se ha explicado anteriormente, las unidades de almacenamiento local diferente pueden diferir desde archivo de discos a paquetes de discos, y así sucesivamente. Por lo tanto, el comportamiento de cada uno de los módulos de funciones puede diferir según exigieran los programas a realizar. O sea, puede haber una permuta de capacidad de almacenamiento contra la velocidad según fuera necesario para diferentes aplicaciones.

10. Habiéndose descrito de un modo general un sistema que emplea el presente invento, se expone a continuación una descripción más detallada de los módulos de funciones 10 que la expuesta al hacer referencia a la figura 1. Estos módulos de funciones son los bloques de construcción de los que se compone el sistema de la figura 1. Según se ilustra en la figura 5, el módulo de funciones está compuesto por una pluralidad de unidades funcionales que funcionan juntas para realizar las funciones que se han descrito anteriormente. Estas unidades comprenden la lista de espera de entrada 20, la unidad lógica 30, la lista de espera de salida 40, el adaptador de interfase de almacenamiento local 50 y la unidad de control 60.

20. La lista de espera de entrada recibe datos procedentes, por ejemplo, del intercambiador 12 de la figura 1, por medio de su vía correspondiente de entrada y almacena los datos mientras espera que la unidad de control 60 quede disponible.

30.

5. Cuando una estructura de datos alcanza la parte superior de la lista de espera de entrada 20, la unidad de control 60 procesa la estructura del dato junto con un nodo o instrucción en la unidad de almacenamiento local asociada 11 de la figura 1. La unidad de control 60 puede utilizar la unidad l6gica 30 para realizar la operaci6n necesaria. El resultado de la operaci6n se puede enviar a la lista de espera de salida 40 junto con una direcci6n o localizaci6n de destino. Cuando este resultado alcanza la parte superior de la lista de espera de salida 40, se envia al destino deseado por medio de la interfase del m6dulo de funciones que comprende la vfa de salida 10a y el intercambiador de primer nivel 12 segun se ilustra en la figura 1.

10. La lista de espera de entrada 20 se ilustra con m6s detalle en la figura 6. La lista de espera de entrada comprende la memoria de lista de espera de entrada 22 y la unidad l6gica para controlar la transferencia del dato entre la vfa 10b de la interfase del m6dulo de funciones y la memoria de lista de espera de entrada 22. La unidad l6gica de interfase comprende el dispositivo l6gico de sincronizaci6n 21 que sincroniza el dato entrante con un temporizador de memoria de lista de espera de entrada. La unidad de reconocimiento de parentesis 23 y el contador de par6ntesis 24 realizan detecci6n de error para la interfase. Se recordará por la descripci6n general de los formatos de estructura de los datos (vease anteriormente) que por cada par6ntesis o corchete de apertura en dicha estructura deberá tambi6n haber un parentesis o corchete de cierre. El contador de par6ntesis 24 es un contador binario ascendente/descendente y se incrementa en uno por cada par6ntesis de apertura o corchete detectado por la

15. unidad de reconocimiento de par6ntesis 23 y se reduce en uno

20.

25.

30.



de espera de entrada 22 está llena o vacía, dependiendo de que la última operación de la memoria única haya sido una operación de escritura o de respectivamente.

5. La unidad lógica 30 de la figura 5 se ilustra con más detalle en la figura 7, La unidad sumadora y lógica 32 realiza todas las operaciones lógicas que han de ser realizadas por el módulo de funciones y, en una modalidad de preferencia, es una memoria de búsqueda en tabla bidimensional. El acumulador 39 se utiliza para ayudar a las operaciones aritméticas que, según se recordará, se realizan de una forma serial. La ejecución de las diversas operaciones es iniciada por el microcontrolador 31 que decodifica los códigos de las operaciones en el registrador de códigos op 33. El registrador de campo C 35 proporciona información modificada de si los resultados de las operaciones lógicas han de transferirse a la lista de espera de salida o de nuevo a la unidad de almacenamiento local 11 de la figura 1. El microcontrolador 31 está provisto de un contador 31a que realiza cualquier aumento o reducción que pudiera ser necesaria por la información de control. Los

10. códigos de operación se suministran al registrador de códigos de op 33 desde la unidad de almacenamiento local 11 (vease la figura 1) por medio de puertas selectoras 36 y 37. La información de control se suministra al registrador de campos C35 desde la unidad de almacenamiento local o desde la lista

15. de espera de entrada 60 (véase la figura 5) por medio de puertas selectoras 36 y 38. El microcontrolador 31 está provisto también de un segundo registrador 34 que es una memoria temporal que tiene una capacidad de 1024 caracteres.

20.

25. Volviendo de nuevo a la figura 5, la lista de espera de salida 40 es esencialmente una memoria tampón para mensajes

30.

5. enviados desde un módulo de funciones particular hasta módulos de funciones. Además, la lista de espera de salida 40 puede acoplarse a la lista de espera de entrada 20 por medio de la interfase del módulo de funciones que comprende la vía de salida 10a, la vía de entrada 10b y el intercambiador 12.
10. La lista de espera de salida se ilustra de una forma más detallada en la figura 8. La lista de espera de salida consiste en dos listas de espera independientes, una para operandos y otra para direcciones de destino. La primera se ilustra en la figura 8 como una memoria de operandos 42 y la segunda se ilustra en la figura 8. como una memoria de destinos 41. El registrador indicador de lectura 45 contiene la dirección donde se ha de llegar en la memoria de destinos 41 para operaciones de lectura, mientras que el registrador indicador de escritura 46 contiene la dirección a la que se ha de llegar en la memoria de destino 41 para una operación de escritura. De un modo similar, el registrador indicador de lectura 47 y el registrador indicador de escritura 48 contiene las direcciones a las que se ha de llegar en la memoria de operandos 42 para operaciones de lectura y escritura, respectivamente. La unidad de control de entrada 43 recibe operandos y direcciones de destino desde la unidad lógica 30 para transferencia a la memoria de destinos 41 o a la memoria de operandos 42.
25. La unidad de control de salida 44 recibe operandos y direcciones de destino desde las memorias respectivas y crea la estructura de datos que se ha de transferir por la vía de salida 10 a la de interfase del módulo de funciones. La transferencia de datos desde la lista de espera de salida comienza por la iniciación de la transmisión de la dirección o localización de destino. La lista de espera de salida determina en
- 30.

entonces la posición apropiada en la que se ha de insertar la estructura del dato para el operando.

5. El adaptador de interfase de almacenamiento 50 de la figura 5 se ilustra con más detalle en la figura 9. Este adaptador recibe direcciones y datos para transferencia a la unidad de almacenamiento local. El adaptador particular ilustrado en la figura 9, está destinado a obtener acceso paralelo a una memoria de acceso casual. Con este fin, el registrador tampón 51 es un registrador de 16 carácter destinado a recibir una palabra de dichos caracteres en paralelo desde la interfase de memoria 54 para transferencia a un "multiplexor" 53 para conversión a una cadena serial de dichos caracteres. Esta conversión se consigue leyendo las posiciones de caracteres diferentes en el "multiplexor" 53 de acuerdo con los cuatro caracteres menos expresivos de la dirección o localización de almacenamiento local en el registrador de direcciones 56. Para la transferencia de datos a la memoria los caracteres se reciben de una forma serial por parte del "D multiplexor" 52 donde se ensamblan en palabras de 16 caracteres los lugares de los caracteres determinados por los cuatro caracteres menos expresivos en el registrador de direcciones 56. Las palabras ensambladas se transfieren desde el "D multiplexor" 52 al registrador tampón 51 y desde éste a la interfase de la memoria 54. Las partes restantes del registrador de localización o direcciones 56 se transfieren también a la interfase de memoria 54 para que las dirección de la memoria tengan acceso a la unidad de almacenamiento local. La unidad de control adaptadora 55 sirve para iniciar una transferencia de dirección desde el registrador de direcciones 56 hasta la interfase de la memoria 54 y también para aumentar y reducir el control de direcciones.
- 10.
- 15.
- 20.
- 25.
- 30.

Se comprenderá que cuando se emplean otros tipos de memoria, por ejemplo almacenamiento o memorias periódicos o ciclos, se podría emplear una interfase de memoria y adaptador de almacenamiento local diferentes. No obstante, la interfase entre el adaptador y el módulo de funciones será la misma y, como en la figura 9, comprenderá: una línea de datos serial por caracteres desde el módulo de funciones, una línea de datos serial por bits desde el módulo de funciones, una línea de direcciones serial por bitio al módulo de funciones, una línea de control de lectura/escritura, una línea de control de direcciones de incremento y decremento, y una línea de control de direcciones de transmisión.

Refiriéndonos de nuevo a la figura 5, la unidad de control 60 sirve para procesar identificaciones y para enviar datos, definidos por la identificaciones, a una unidad lógica 30, si dicho dato es un operador o un operando, a la lista de espera de salida 40 si dicho dato es una dirección o localización de destino, y a una unidad de almacenamiento local 11., de la figura 1, si dicho dato es un operando o una dirección que se ha de almacenar en la memoria local. Todos los datos procedentes de la lista de espera de entrada 20 pasarán primero a través de la unidad de control 60. La entrada de datos se acepta desde la lista de espera de entrada 20 desde la unidad de almacenamiento local 11 (véase la figura 1) como fuente de origen.

La unidad de control 60 se ilustra con más detalle en la figura 10. Según se ilustra en esta figura, los datos procedentes de la lista de espera de entrada y procedentes de la unidad de almacenamiento local, reciben respectivamente, en el contador de paréntesis de lista de espera de entrada 71 y el contador de paréntesis de almacenamiento local 70. El montaje de

- paréntesis o corchetes determina si el dato constituye el contenido interior o exterior de la estructura de datos. Si el dato es una estructura interior, representa un operador, operando, u otra información y se transfiere a la unidad de selección de trayecto de datos 72 por el detector de contenido interior/exterior 68 ó 69, dependiendo de si el dato se ha recibido, respectivamente, de la memoria local o de la lista de espera de entrada. Si el dato es un contenido exterior representa una identificación y se transfiere respectivamente al registrador de indicaciones de la memoria local 67 o al registrador de identificación de la línea de espera de entrada 66, de nuevo dependiendo de si procede de la unidad de almacenamiento local o de la lista de espera de entrada. El contenido del registrador de identificaciones de la lista de espera de entrada 66 y el registrador de identificaciones del almacenamiento local 77 son elegidos por la puerta selectora 65 para transferencia al verificador de esencias de identificaciones 63 o se pueden guardar en el registrador de conservación de identificaciones 64 para futura referencia. El secuenciador de microcontrol 61 es una memoria de lectura solamente de búsqueda en tabla (ROM) y proporciona las diversas secuencias de las señales de control requeridas para iniciar las operaciones respectivas en las otras unidades del módulo de funciones bajo el control del verificador de secuencia de identificaciones 63 y otros datos modificadores según se reciben de esas otras unidades así como los datos procedentes del contador 73 y los basculadores de estado 62 que indican el estado particular de la máquina en el que se encuentra el módulo de funciones. Los basculadores de estado 62 se colocan por la acción del microcontrolador de acuerdo con la secuencia particular pedida por el operador de identifica-
- 5.
- 10.
- 15.
- 20.
- 25.
- 30.

ciones.

Los intercambiadores respectivos 12 y 16 de la figura 1 se ilustran con más detalle en la figura 11. Según se ha descrito anteriormente, cada estructura de dato transmitida por una vía de interfase contendrá una designación de módulos de funciones como primer campo o componentes de datos en dicha estructura. Esta designación del módulo de funciones es una dirección mediante la cual una estructura de datos se commuta el módulo de funciones apropiado por parte del intercambiador. En la figura 11, las estructuras de datos se reciben por una vía respectiva 10a en un decodificador de direcciones correspondiente 80. El decodificador de direcciones 80 decodifica la designación del módulo de funciones y pone en condiciones una de las puertas de transmisión 81 para elegir la vía apropiada 10b para la transmisión del módulo de funciones designados.

Los decodificadores de direcciones respectivos 80 se interconectan por una resolución de prioridad y el circuito de exclusión (no ilustrado en la figura 11) por lo que dos transmisiones de estructura de dato simultáneas no puede designar el mismo módulo de funciones. Las vías respectivas 10a está provistas de líneas de señales de reensallo para indicar el módulo de funciones transmisor que se debe volver a iniciar la transmisión del dato.

En la situación en que dos o más módulos de funciones se acoplen a las mismas vías de interfases, cada módulo de funciones estará provisto de su propio decodificador de designaciones y los codificadores de designaciones respectivos 80 en la figura 11 elegirían la misma vía de salida 10b en respuesta a dos o más designaciones diferentes de funciones.

5. Cuando un sistema emplea dos o más intercambiadores de primer nivel 12 que se acoplan a intercambiador de segundo nivel 10, el campo de designaciones de módulos de funciones se ampliará en longitud y los decodificadores respectivos 80 del intercambiador 16 pueden adaptarse para decodificar solamente una parte del campo de designación con el fin de elegir un intercambiador de primer nivel apropiado 12. Los decodificadores respectivos 80 de intercambiador 12 decodificarían solamente una segunda parte del campo de designación para elegir el módulo de funciones apropiados.

10. Cuando se emplean tres niveles de intercambiadores, el campo de designaciones se ampliará de nuevo y así sucesivamente.

15. La interpretación de una instrucción comienza analizando en primer lugar el componente del dato presentado por la lista de espera de entrada. El análisis de espera de entrada continúa hasta que se detecta un estado no cumplido, después de lo cual comienza el análisis del componente del dato de la memoria local. La interpretación de la instrucción continúa hasta que se ha interpretado el par completo de células (contaje cero de corchete) o se ha detectado un estado sin sentido.

20. El ciclo de interpretación de instrucción normal termina cuando el conteo de corchete es igual a cero y, a su vez, activa la interpretación del segundo componente o célula de dato de la lista de espera de entrada. Los ciclos de interpretación de instrucciones que terminan debido a la detención de un estado sin sentido da por resultado una interrupción generada en la máquina. La ejecución real de la máquina se realiza bajo control de microinstrucciones de la máquina que se derivan de la memoria de lectura solamente del secuenciador de microcontrol 61 (véase la figura 10) de la unidad de control por inter

25.

30.

pretación por parte de la unidad de control de ambos datos de contenido interior y exterior de los pares de células de datos representados por la lista de espera de entrada y el área de almacenamiento local.

5. El módulo de funciones, según se ilustra en la figura 5, ejecuta de un par de estructuras de datos asociadas, una en la lista de espera de entrada 20 y una en la unidad de almacenamiento local 11 (véase la figura 1). Cada estructura en la lista de espera de entrada 20 consiste en una parte de dirección y una parte de argumento. La parte de dirección indica hacia el lugar de almacenamiento local que permite la asociación de las dos estructuras de datos. El control comienza con la primera construcción de la estructura del dato de la lista de espera de entrada y continúa hasta que encuentra una construcción que no es un sucesor válido de la construcción anterior según define el flujo de secuencia válida. La última construcción se conserva y el control se transfiere a la construcción siguiente en la unidad de almacenamiento local 11 que se somete a prueba como posibles sucesor válido. Si no es así el resultado es un error. Si la construcción siguiente es un sucesor válido, se continúa el control con la unidad de almacenamiento local hasta que se encuentra un sucesor no válido, en cuyo momento se transfiere de nuevo al control a la lista de espera de entrada 20.
10. 15. 20. 25. 30.
- Existen ocasiones en que la lista de espera de entrada 20 y la memoria local 11 se exploran simultáneamente. Esto ocurre durante una operación diádica cuando ambos operandos están presentes en las unidades respectivas. Al completarse la exploración simultánea, el control se transfiere a la unidad de almacenamiento local. Este proceso continúa hasta que se

detecta el final de la estructura del dato de la lista de espera de entrada, en cuyo caso la operación ha finalizado.

- Como el procedimiento de explorar las estructuras respectivas es simultáneo, no tiene significado alguno la comprobación de la validez de una estructura de dato en sí. Por lo tanto, las estructuras respectivas deberán considerarse como pares significativos. Las estructuras válidas son pares de estructuras de datos, una en la lista de espera de entrada y otra en la memoria o unidad de almacenamiento local que, si se explora en la forma de exploración descrita anteriormente da por resultado una secuencia válida de construcciones. Una secuencia válida de construcciones se define como una secuencia válida de identificaciones. Para interpretar los flujos de secuencias válidas, no es necesariamente la secuencia del encuentro real de identificaciones lo que es importante, sino la secuencia en que se han de ejecutar. Debido a las reglas de exploración y naturaleza de las estructuras, las estructuras de datos que se introducen en la lista de espera de entrada deben adoptar la forma (( ) 0 ( ) 1 ( ) ———). En dicha estructura, los paréntesis o corchetes antes del número 0 indican el comienzo de una estructura de dato; el contenido de los paréntesis o corchetes después del número 0 representan una dirección del módulo de funciones y el contenido de los corchetes o paréntesis después del número 1 representan la dirección en la unidad de almacenamiento o local de la estructura del dato asociada necesaria para completar la rutina conjunta. Las construcciones restantes en dicha estructura del dato pueden ser de cualquier secuencia de construcciones que tenga un contaje de paréntesis equilibrado y que de por resultado una secuencia de identificaciones válidas. De un modo similar, las estructuras de datos
- 5.
- 10.
- 15.
- 20.
- 25.
- 30.

5. contenidas en la unidad de almacenamiento local debe adoptar la forma (( )-----). Durante la transferencia del dato desde la lista de espera de entrada hasta otras unidades del módulo de funciones, incluyendo la memoria local, los campos se transfieren por separado con la estructura de corchetes de apertura (( ) la estructura de corchetes o paréntesis se añade de nuevo a la estructura del dato según se forma en la memoria local.

10. Las estructuras de datos pueden procesarse o elaborarse en una u otra dirección, izquierda a derecha o derecha a izquierda. La dirección de elaboración está determinada por el paréntesis o corchete inicial localizado en la unidad de almacenamiento local. Una exploración de derecha a izquierda se interpreta como una imagen de espejo de una exploración de izquierda a derecha. Durante la exploración de izquierda a derecha, los caracteres de los datos se interpretan directamente según se exploran y los paréntesis de izquierda y derecha se interpretan como paréntesis de derecha e izquierda, respectivamente.

15. Las estructuras de los datos se localizan siempre comenzando en un paréntesis o corchete. La dirección de la exploración puede ser cambiada solamente por una instrucción explícita.

20. ta.

25. Según se ha descrito, los juegos de operandos y de datos de control pertenecientes a un ciclo de instrucciones integrales de la máquina se obtienen de los componentes de datos presentados por la lista de espera de entrada 20 y la unidad de almacenamiento local 11. El módulo elaborador se estimula en un ciclo de interpretación de instrucciones al reconocer la presencia de un componente de datos en la parte superior de la lista de espera de entrada. El componente del dato presentado

30. por la lista de espera de entrada va precedido de una dirección

que lo enlaza a su componente de dato correspondiente en la unidad de almacenamiento local del módulo de funciones.

5. En la figura 3a se ilustra un ejemplo de una estructura de datos que se almacenan normalmente en la unidad de almacenamiento local, mientras que la figura 3b es una estructura de datos del tipo recibido normalmente como información de entrada. Se observará por la figura 3a y 3b que cada estructura de dato comienza, en una exploración de izquierda a derecha, con dos paréntesis o corchetes de apertura en sucesión seguidos por un paréntesis o corchete de cierre. El primer campo precedido por el número 5 es un campo de condiciones que se emplea como una ramificación de camino N a una de las listas de direcciones de destino N. Cada lista de direcciones puede contener direcciones de destino múltiples según se indica en la figura 3a. El segundo campo en la estructura de datos de la figura 3a va precedido por el carácter C que indica que el contenido interior de la célula constituye información de control. El tercer campo de la figura 3a es un operando según indica el contenido exterior o identificación 7 como se ha descrito anteriormente de una forma general. Los campos restantes son direcciones de destino.

10.

15.

20.

25. En la figura 3B el primer campo después del comienzo de la estructura de dato es una designación de módulos de funciones indicada por el contenido exterior precedente o identificación cero. El segundo campo es la dirección de una estructura de dato asociada en la unidad de almacenamiento local, como la que se ilustra en la figura 3a. En la figura 3b, está dirección de unidad de almacenamiento local se designa por el contenido exterior anterior o identificación uno. El campo siguiente en la estructura del dato de la figura 3a es

30.

un calificador de dirección según indica el contenido exterior o identificación AQ. El último campo de la estructura del dato de la figura 3b es un operando.

5. A continuación se describen los diversos tipos de identificación o contenidos exteriores. Estas identificaciones comprenden identificaciones de dirección o localización, identificaciones de nivel, identificaciones de modificador de dirección, identificaciones de operador y control, identificaciones de operando, identificaciones de dirección de destino, 10. identificaciones de operador de microcadena, e información de estado.

15. Las identificaciones de direcciones comprenden direcciones de módulos de funciones, direcciones de almacenamiento o memorias locales, y calificadores de direcciones. A continuación se describen otras identificaciones de direcciones. La identificación ops diádica indica que el contenido interior de la expresión de paréntesis siguiente especifica que el operando siguiente es un operando de izquierda o derecha. La identificación del campo de condiciones indica que uno de los contenidos interiores siguientes especifica si el operando acompañante ha de tratarse como un Boolean o un operando. 20. La identificación del campo de composición indica que el contenido interior siguiente especifica el subfijo del bitio de presencia en el campo de operandos correspondiente en el que se ha de escribir el operando. 25. La identificación del campo de descomposición indica que el contenido interior siguiente especifica el subfijo del bitio de presencia en el campo de operandos correspondientes en el que se ha de escribir el operando. La identificación del campo de descomposición indica que el contenido interior siguiente especifica el número de bitios que se ha de ele 30.

gir del campo de operandos en la lista de espera de entrada. Estos bits han de utilizarse como indicador del índice para elegir una microcadena. La identificación del campo de bloqueo indica que el contenido interior siguiente especifica si la red protegida por el campo del bloqueo ha de bloquearse o desbloquearse.

5.

Las identificaciones de nivel están indicadas por el carácter alfanumérico L e indican que el contenido interior siguiente de una expresión de paréntesis constituye información de nivel. Esta información es despreciada por el módulo de funciones.

10.

Las identificaciones de modificador de direcciones son la identificación de índice de nodos, identificación de nombres, e identificación del modificador de direcciones.

15.

La identificación del índice de nodos es indicada por el número 2 y especifica que la dirección de almacenamiento local ha de ponerse en secuencia con el nodo indicado al nivel corriente. Al completarse, el nivel de interpretación desciende un nivel. La identificación de nombres está indicada por el

20.

número 3 y especifica que la dirección de la memoria local ha de ponerse en secuencia con el nodo indicado comparando la identificación de nombres contra el contenido exterior de la estructura de dirección al nivel corriente. Al completarse, el nivel de interpretación desciende un nivel. La identificación

25.

del modificador de direcciones está representada por los caracteres AM e indican que el contenido anterior siguiente constituye un código de operador que se utiliza como modificador de direcciones para la dirección de la memoria local.

30.

A continuación se describe las identificaciones de operador y control. El número 4 indica que el contenido inte-

- rior de la expresión de paréntesis siguiente constituye un código op modificador de operando. El número 5 indica que el contenido interior siguiente constituye el código de operador diádico, código de operador monádico, o tipo de célula. El carácter
5. A indica que el contenido interior siguiente constituye un código de operador diádico acumulador. El carácter I indica que el contenido interior siguiente constituye un código de operador diádico de índice. El carácter Z indica que el contenido interior siguiente constituye un código de operador diádico
10. "zap". Los caracteres ZA indican que el contenido interior siguiente constituye un código operador diádico acumulado "zap". Los caracteres ZI indican que el contenido interior siguiente constituyen un código operador de índice "zap". Los caracteres
15. MS indican que el contenido interior siguiente constituye una microcadena de código que se utiliza en la función de descomposición. El carácter C indica que el contenido interior siguiente constituye información de control para una célula específica u operador.
20. Las identificaciones de operadores de microcadenas se describen a continuación. El carácter "indica que el contenido interior de la expresión siguiente constituye un literal que ha de transferirse a la lista de espera de salida. E. carácter E indica que el contenido interior siguiente constituye un operando sin designar que define un valor de salto del registrador de direcciones de la memoria local. El salto lleva la misma dirección de la corriente. El carácter F indica que el contenido interior siguiente constituye un operando sin designar que define un valor de salto del registrador de direcciones de lista de espera de entrada. La dirección del salto es de izquierda a derecha. El carácter B indica que el contenido inte-
- 25.
- 30.

rior siguiente constituye un operando que especifica el número de bits que se han de transferir de el operando en la lista de espera de entrada a la lista de espera de salida. El carácter M indica que el contenido interior siguiente constituye un código de operando que ha de utilizarse dentro de una microcadena.

5.

Se comprenderá que los campos siguientes se consideraran como operandos: designación del módulo de funciones, dirección de la unidad de almacenamiento o memoria local, índice de nodos, modificador de dirección, operadores de microcadenas que se han definido anteriormente designados por los caracteres", E, F, y B, así como el propio campo de operandos. La identificación de operando 7 se utiliza también en estructuras que los operadores de lectura y escritura.

10.

15.

Los operandos con signo tienen el formato (S AOAI...An) donde AO es el bitio menos expresivo y S es el bitio con signo, siendo positivo S = 0. Los operandos sin signo tienen el formato (AOAI...An). La longitud del campo es variable hasta una cierta cantidad específica. Como se define que son operandos el contenido interior de un campo, no se permiten paréntesis o corchetes dentro del campo de operandos. Los campos siguientes tienen formatos similares a los operandos sin signos: modificador de dirección de identificación de nombre, el código OP de modificador de operando, el código de operador diádico/modádico o tipo de célula, el código de operador diádico acumulador, el código de operador diádico de índice, el código de operador diádico "zap", el código operador diádico acumulado "zap", el código de operador de índice "zap", las direcciones de destino, el modificador de direcciones, y el código de operador que se emplean dentro de una microcadena.

20.

25.

30.

El campo de control puede comprender una célula de condiciones, una célula de posiciones, una célula de bloqueo o un operador diádico. La condición tiene el formato C(R, CO, PB, PO, BO, BI, ..., Bn). R es un bitio de reposición que, si se coloca, indica que el operando siguiente, dirigido a su nodo, debiera ignorarse y la única acción necesaria es la reposición del bitio de reposición. PB es un bitio de presencia de Boole. PO es un bitio de presencia de operando, CO es un bitio de constante de operando, BO....Bn son bitios de Boole con una longitud mínima de un carácter y una longitud máxima de 8 caracteres.

La célula de composición como campo de control tiene el formato C(CO, PO, CI, ..., Cn, Pn) donde Cn es un bitio constante para el operando nth y Pn es el bitio de presencia del operando nth. La célula de bloqueo tiene un solo bitio de bloqueo y es del formato C (L). El operador diádico tiene el formato C(R, C, PL, PR) donde C es el bitio constante y, si es igual a uno, indica que el bitio P no se tiene que reponer. PL es el bitio de presencia para el operando izquierdo y PR es el bitio de presencia para el operando derecho. Los campos calificadores de direcciones comprenden una célula de composición, una célula de condición, ops diádicos, una célula de descomposición y una célula de bloqueo. La célula de descomposición es del formato AQ(n) donde n es un operando sin signo cuya longitud tiene un máximo de 8 bitios. La célula de condición es una célula de un bitio del formato AQ(n) donde n = 0 indica un algebra de Boole y n = 1 indica un operando. Los ops diádicos son del formato AQ(n), donde n = 0 indica un operando izquierdo y n = 1 indica un operando derecho. La célula de descomposición es del formato AQ(n), donde n es un operando sin signo con una longitud máxima de 10 bitios. La célula de bloqueo es

una célula de un bitio del formato AQ(n), donde n = 0 especifica bloqueo y n = 1 especifica desbloqueo.

La dirección de destino es una estructura que debe adoptar el formato 6(( ) 0 ( ) 1 (LS) ...) donde la única limitación en el dato que sigue a la dirección de memoria local LS es que dicho dato y el nodo al que se dirige la localización de destino debe cumplir las exigencias sintácticas de las estructuras de los datos. Una excepción al formato de direcciones de destino anterior es la dirección nula que es del formato 6 (( )).

Una microcadena es una estructura que debe adoptar el formato NS(( )...). La microcadena puede contener solamente identificaciones de 6 para células condicionales. Para células de descomposición, la microcadena puede contener combinaciones de las siguientes identificaciones: ",B,E,F,W,AQ,MS, y 6, cuyas identificaciones se han definido anteriormente.

La célula condicional se utiliza como una ramificación de camino N a una de las listas de direcciones de destino N. Cada lista de direcciones puede contener direcciones de destino múltiples, La célula de composición se utiliza para recoger y catenar entonces una pluralidad de campos de operandos. El resultado puede ser uno o más operandos. Cuando han llegado todos los operandos, los operandos resultantes se envían a los destinos. La célula de descomposición se utiliza para descomponer el operando presente por la lista de espera de entrada según determine la microcadena de la célula. La célula de bloqueo se utiliza para bloquear y desbloquear una red.

Los operadores monádicos aritméticos/lógicos se describen a continuación. ANY I produce la transferencia de un operando con un valor "1" a los destinos apropiados si el ope-

- rando contiene cualquier carácter con un valor de 1. De otro modo se transfiere un 0. ANY 0 produce la transferencia de un operando con el valor de 1 a destinos apropiados si el operando contiene cualquier carácter con el valor de 0. De otro modo
5. se transfiere a un 0. EL CONTAJE DE UNOS produce la transferencia de un operando a un destino apropiado con un valor igual al número de caracteres "1" presentes en el operando. EL CONTAJE DE CEROS produce la transferencia de un operando a un destino apropiado con un valor igual al número de "ceros" presentes
10. en el operando. Existen 5 operadores de lectura/escritura que se describen a continuación. LA ESTRUCTURA DE LECTURA produce la transferencia del contenido de la memoria local a la que se ha enviado la dirección o localización a destinos apropiados. Si el subespacio de la memoria local a la que se ha dirigido la dirección contiene una estructura, la estructura se transfiere como un campo con la identificación de 7. ANULACION DE
15. PARENTESIS produce una búsqueda de la estructura localizada y una anulación de todos los paréntesis o corchetes y contenido exterior. El resultado se envía a un destino apropiado con una
20. identificación de 5. LECTURA DE ESTRUCTURA produce la transferencia de la estructura que sigue al operador desde la parte superior de la lista de espera de entrada al interior del subespacio de la memoria local a la que se ha enviado la dirección. La igualdad de la definición de paréntesis o corchetes se com-
25. prueba antes y después de la transferencia. LA ESCRITURA DE ESTRUCTURA TRANSPARENTE produce la transferencia de la estructura siguiente desde la parte superior de la lista de espera de entrada al interior del subespacio de la memoria local a la que se ha enviado la dirección. La estructura se almacena en
30. su forma entre paréntesis o entre corchetes. LA ESCRITURA DE

- VECTORES ANALIZADOS produce la transferencia del operando siguiente desde la parte superior de la lista de espera de entrada al interior del subespacio de la memoria local a la que se ha enviado la dirección. El vector de bitio entrante se analiza en la memoria local según indica la definición de paréntesis del subespacio de memoria local. LA LECTURA DE CARACTERES PRODUCE LA TRANSLACION O TRADUCCION DE CARACTERES EN EL SUBESPACIO DE MEMORIA LOCAL AL JUEGO O PARES DE CARACTERES SEGUN SE INDICAN EN LA TABLA DE LA FIGURA 2. El resultado se transfiere como un operando al destino apropiado. LA LECTURA DE CARACTERES produce la traslación o traducción de un operando en la lista de espera de entrada desde una representación de pares de caracteres a una representación de caracteres. El resultado se escribe en el subespacio de memoria local direccionalizada.

- A continuación se describen los dos operadores de descripción de campo. La indicación HACER DESCRIPCION DE CAMPO crea una descripción de campo desde el subespacio de memoria local a la que se ha dirigido la dirección copiando los paréntesis o corchetes y contenido exterior y reemplazando el contenido interior de la estructura por el contaje que representa el número de caracteres en el contenido interior. La descripción del campo resultante se transfiere al destino apropiado. La indicación HACER ESTRUCTURA crea una estructura del subespacio de memoria local a la que se ha enviado la dirección que contiene una descripción de campo. La estructura resultante se transfiere al destino apropiado. Los otros operadores monádicos son el operador de MARCA que causa una transferencia del contenido del registrador de direcciones de la memoria local a un destino apropiado la operación de REPOSICION que repone

Los bits de presencia en el subespacio de memoria local a la que se ha enviado la dirección en cualquier de los bits de presencia o un 1. De otro modo, se coloca el bit de reposición; y el operador NOOP que hace que no se inicie la operación.

5. Existen 7 operadores modificadores de operandos que colocan respectivamente el primer bit del operando en 1 ó 0, complementan el primer bit del operando, catenan un 0 o un 1 al extremo menos expresivo del operando sin signo, anulan el bit menos expresivo del operando sin signo y el complemento lógico.

10. Los operadores diádicos realizan las operaciones normales aritméticas y lógicas. Los operadores diádicos acumulados son una variante de algunos de los operadores diádicos normales. El operador acumulado hace que se envíe un resultado a la dirección de destino apropiado y también al campo de operandos en la memoria local. Los operadores diádicos del índice son

15. idénticos a los operadores diádicos acumulados excepto que no se proporcionan direcciones o localizaciones de destino. Los operadores diádicos "zap" son variantes de los operadores diádicos normales y se emplean cuando se desean una operación de tipo diádico a realizar sobre un campo distinto a un operando.

20. Según se ha descrito anteriormente, el procedimiento y método del presente invento emplean una red de módulos de elaboración de funciones y unidades de almacenamiento locales o memorias asociadas con cada uno de los módulos de funciones.

25. La ejecución por parte de los módulos de funciones se inicia por la presencia de todas las estructuras requeridas de los datos, una de las cuales reside normalmente en la memoria local y contiene un operador, recibiendo la otra estructura de los datos en el módulo de funciones y comprendiendo un argumento y una dirección que apunta al lugar de la memoria local don

30.

de reside la estructura del dato asociada. De esta manera, se puede distribuir una serie de instrucciones que constituyen un procedimiento, programa o tarea por toda la red para aumentar el comportamiento o rendimiento de proceso al adaptarse a pro-

5. procesos simultáneos. Además, se puede operar sobre campos de datos según se transfieren desde módulo de funciones a módulos de funciones para conseguir un efecto de canalización que depende solamente del ritmo de transferencia entre módulos de funciones. La transferencia de datos es serial por caracteres y sincrónica con los módulos de funciones pero asincrónica entre módulos de funciones. Las unidades diversas de almacenamiento local o memoria pueden diferir en naturaleza, capacidad y tiempo de acceso para adaptarse a diferentes exigencias de procesos de datos. Las unidades de almacenamiento locales o memorias
10. son normalmente periódicas por naturaleza y pueden comprender unidades de paquetes de discos, unidades de archivo de discos, memorias de líneas de retardo, dispositivos acoplados de carga y memoria de burbujas. Las unidades de almacenamiento locales o memoria pueden ser memorias de acceso casual cuando esto sea conveniente.
15. 20.

El procedimiento puede ampliarse desde un módulo simple de funciones y la memoria local asociada a cualquier número de los módulos de funciones y unidades de almacenamiento o memoria locales que puedan ser servidas por una red encajada de intercambiadores. De esta manera, se puede crear una familia de sistemas para proporcionar un espectro de capacidades de proceso o elaboración que alcance desde el espectro de un sistema muy pequeño hasta un sistema muy grande. No obstante, la unidad básica de la que se forma los sistemas sería uniforme,

25. 30. de manera que no tuvieran que crearse circuitos y diseños ar-

quitectónicos diferentes.

A pesar de que solamente se han descrito ciertas modalidades del presente invento, resultará evidente a los expertos en la materia que se pueden efectuar diversos cambios y modificaciones sin desviarse del espíritu y alcance del invento según se indica en las reivindicaciones adjuntas.

N O T A

Descrita suficientemente la naturaleza del invento, así como la manera de realizarlo en la práctica, debe hacerse constar que las disposiciones anteriormente indicadas son susceptibles de modificaciones de detalle en cuanto no alteren su principio fundamental. También se hace constar que el invento corresponde a una solicitud de patente presentada en Norteamérica con el nº Ser. No. 447.040 de 28 de Febrero de 1.974, acogiéndose por lo tanto a los beneficios que conceden los Convenios Internacionales en vigor, siendo lo que constituye la esencia del referido invento y por lo que se solicita Patente de Invención por 20 años, sobre: PROCEDIMIENTO Y SISTEMA DE PROCESO DE DATOS MEDIANTE UNA RED ACTIVADA POR DATOS, caracterizándose por lo siguiente:

1.- Procedimiento y sistema de proceso de datos mediante una red activada por datos, cuyo procedimiento se caracteriza porque comprende cuando una fuente de segmentos de datos acoplada al dispositivo de interconexión, transfiere un segmento de datos desde la fuente hasta una de las unidades programables para que se realice una operación sobre el mismo; y transfiere un segundo segmento de dato a otra de las unidades programables para que se realice so-

bre el mismo una operación simultánea.

2.- Procedimiento según la reivindicación 1, caracterizado porque comprende la fase de transferir un segmento de dato desde una unidad programable hasta una segunda unidad programable para que se realice sobre el mismo una segunda operación después de haberse realizado una primera operación sobre el segmento de dato por parte de la unidad programable.

5.

10.

15.

20.

25.

30.

3.- Sistema para la realización del procedimiento según las reivindicaciones anteriores, caracterizado porque comprende una pluralidad de unidades programables; medios que interconectan las unidades programables para la transferencia de datos entre las mismas; y una pluralidad de unidades de almacenamiento o memorias, una por cada unidad programable, acoplándose cada unidad de almacenamiento o memoria a su unidad programable respectiva, conteniendo cada unidad de almacenamiento o memoria uno y en caso dado más operadores de control almacenados en las mismas; comprendiendo cada unidad programable una unidad lógica para realizar operaciones sobre segmentos de datos, medios de entrada acoplados a la unidad lógica para recibir segmentos de datos con el fin de transferirlos a la unidad lógica, y una unidad de control acoplada a los medios de entrada y la unidad lógica y la unidad de almacenamiento y en caso dado memoria para tener acceso a la unidad de almacenamiento o memoria para un operador de control en respuesta a la recepción de un segmento de datos por parte del dispositivo de entrada y para indicar a la unidad lógica que realice una operación sobre el segmento del dato.

4.- Sistema según la reivindicación 3, caracterizado porque comprende una fuente de segmento de datos acopladas a los medios interconexión para suministrar segmentos de datos a

dos de las unidades programables para operación simultánea.

5. 5.- Sistema según la reivindicación 3, caracterizado porque una unidad programable se acopla a otra unidad programable para recibir un segmento de datos de la misma y realizar una operación sobre el mismo después que la otra unidad programable ha realizado una operación sobre el segmento de dato.

6.- Sistema según la reivindicación 3, caracterizado porque los medios de interconexión comprenden una vía para la transferencia de datos seriales.

10. 7.- Sistema según la reivindicación 3, caracterizado porque comprende una fuente de datos periférica donde una de las unidades programables se acopla a la fuente de datos periférica para controlar la transferencia de datos entre las mismas.

15. 8.- Sistema según la reivindicación 3, caracterizado porque cada unidad de control de las unidades programables comprende una memoria donde se almacenan señales de control.

20. 9.- Sistema según las reivindicaciones anteriores, caracterizado porque comprende; una pluralidad de unidades programables; medios que interconectan las unidades programables para la transferencia de datos serial entre las mismas de segmentos de dato de longitudes diferentes, estando los segmentos de datos precedidos en el tiempo por una señal de segmentos de iniciación y seguidos de una señal de segmentos de finalización; comprendiendo cada una de las unidades programables medios receptores acoplados a los medios de interconexión para recibir los segmentos de datos desde otra unidad programables, comprendiendo los medios receptores medios de detección de la señal de segmento de iniciación y medios de detección de la  
25. señal de segmento de finalización para determinar el comienzo  
30.

y final de las transferencias serial de los segmentos de datos diferentes.

5. 10.- Sistema según la reivindicación 9, caracterizado porque los segmentos de datos están compuestos por un juego de cuatro caracteres, estando formado cada carácter por dos dígitos binarios, y porque los medios de detección de la señal de segmento de iniciación comprenden una circuiteria lógica para detectar una primera diferencia entre el primer y el segundo dígitos binarios de un carácter; y porque los medios de detección de la señal de segmento de finalización comprenden circuiteria lógica para detectar una segunda diferencia entre el primer y el segundo dígitos binarios de un carácter.

15. 11.- Sistema según la reivindicación 10, caracterizado porque comprende una fuente de segmentos de dato acoplada a los medios de interconexión para suministrar segmento de datos a dos de las unidades programables para operación simultánea.

20. 12.- Sistema según la reivindicación 10, caracterizado porque una unidad programable se acopla a otra unidad programable para recibir un segmento de dato de la misma y para realizar una operación sobre el segmento de dato después que la otra unidad programable ha realizado una operación sobre el segmento de datos.

25. 13.- Sistema según las reivindicaciones anteriores, caracterizado porque comprende una pluralidad de unidades programables, cada una de las cuales tiene un dispositivo de entrada y un dispositivo de salida; un intercambiador que interconecta las unidades programables para acoplar de una forma selectiva el dispositivo de salida de cualquier unidad programable al dispositivo de entrada de cualquier unidad programa-
- 30.

ble para la transferencia de datos entre las mismas; y una pluralidad de unidades de almacenamiento o memorias, una por cada unidad programable, acoplándose cada unidad de almacenamiento o memoria a su unidad programable respectiva, conteniendo cada unidad de almacenamiento o memoria uno o más operadores de control almacenados en las mismas; comprendiendo cada unidad programable una unidad lógica para realizar operaciones sobre segmentos de datos, medios de entrada acoplados a la unidad lógica para recibir segmentos de datos para la transferencia a la unidad lógica, y una unidad de control acoplada a los medios de entrada y la unidad lógica y la unidad de almacenamiento o memoria para tener acceso a la unidad de almacenamiento o memoria para un operador de control en respuesta a la recepción de un segmento de dato por parte de los medios de entrada y para indicar a la unidad lógica que realice una operación sobre los segmentos de datos.

14.- Sistema según la reivindicación 13, caracterizado porque comprende dos vías en serie, de las cuales una acopla al intercambiador al dispositivo de entrada de dos unidades programables y la otra de las vías acopla el intercambiador al dispositivo de salida de las dos unidades programables.

15.- Sistema según la reivindicación 13, caracterizado porque comprende una unidad de control de entrada-salida que tiene medios de entrada y medios de salida acoplados a los dispositivos intercambiadores para transferir datos a una unidad programable; y un dispositivo periférico acoplado a los medios de control de entrada - salida.

16.- Sistema según la reivindicación 13, caracterizado porque comprende una unidad programable separada de la pluralidad de unidades programables y que dispone de un dispositi-

tivo de entrada y un dispositivo de salida; y un segundo intercambiador acoplado a las unidades programables separadas para conectar de una forma selectiva el dispositivo de salida de la unidad programable al dispositivo de entrada de la unidad programable; acoplándose al segundo intercambiador al intercambiador para la transferencia de datos entre las unidades programables separadas y una de las pluralidad de unidades programables.

5.

17.- Sistema según la reivindicación 16, caracterizado porque comprende una unidad de interfase de intercambiador acoplada entre el intercambiador y el segundo intercambiador con el fin de tamponar la transferencia de datos entre los mismos.

10.

18.- Sistema según las reivindicaciones anteriores, caracterizado porque comprende una pluralidad de unidades programables que tiene un dispositivo de entrada y un dispositivo de salida; un primer intercambiador que interconecta dicha pluralidad de unidades programables para acoplar de una forma selectiva cualquier dispositivo de salida a cualquier dispositivo de entrada; una unidad programable separada de la pluralidad

15.

de unidades programables y que tiene un dispositivo de entrada y un dispositivo de salida; y un segundo intercambiador acoplado a las unidades programables separadas para conectar de una forma selectiva el dispositivo de salida de la unidad programable al dispositivo de entrada de la unidad programable; acoplándose al segundo intercambiador al primer intercambiador para la transferencia de datos entre las unidades programables separadas y una de la pluralidad de unidades programables.

20.

25.

19.- Sistema según la reivindicación 18, caracterizado porque comprende una segunda unidad programables separada de las unidades programables y que dispone de un dispositivo

30.

de entrada y un dispositivo de salida; y un tercer intercambiador acoplado a la segunda unidad programable separada para conectar de una forma selectiva el dispositivo de salida de la segunda unidad programable separada a su dispositivo de entrada; acoplándose al tercer intercambiador al segundo intercambiador para la transferencia de datos entre la segunda unidad programables separada y otra unidad programable en el sistema.

20.- Sistema según las reivindicaciones anteriores, caracterizado porque comprende una pluralidad de unidades programables cada una de las cuales comprende una unidad lógica, una pluralidad de unidades de almacenamiento o memoria cada una de las cuales se acopla a una unidad programable respectiva para contener uno o más operadores de control almacenados en las mismas, y un dispositivo de interconexión que interconecta las unidades programables para la transferencia de datos entre las mismas, el procedimiento que comprende: recibir un segmento de dato en una de las unidades programables; tener acceso a la unidad de almacenamiento respectiva acoplada a la unidad programable en respuesta al segmento de dato recibido para buscar un operador de control de una unidad de almacenamiento o memoria; e indicar a la unidad lógica de la unidad programable que realice una operación sobre los segmentos de datos recibidos.

21.- Procedimiento y sistema de proceso de datos mediante una red activada por datos, tal y como queda sustancialmente descrito en la presente Memoria e ilustrado en los dibujos adjuntos.

Esta Memoria consta de 49 hojas escritas a máquina

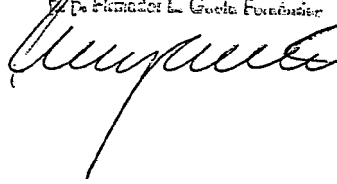
por una sola cara.

26 OCT. 1976

Madrid,

BURROUGHS CORPORATION

~~El Encargado de Negocios y Representante~~  
~~de la Burroughs L. Gueta Forés~~

A handwritten signature in dark ink, appearing to be 'L. Gueta Forés', written over the typed name.

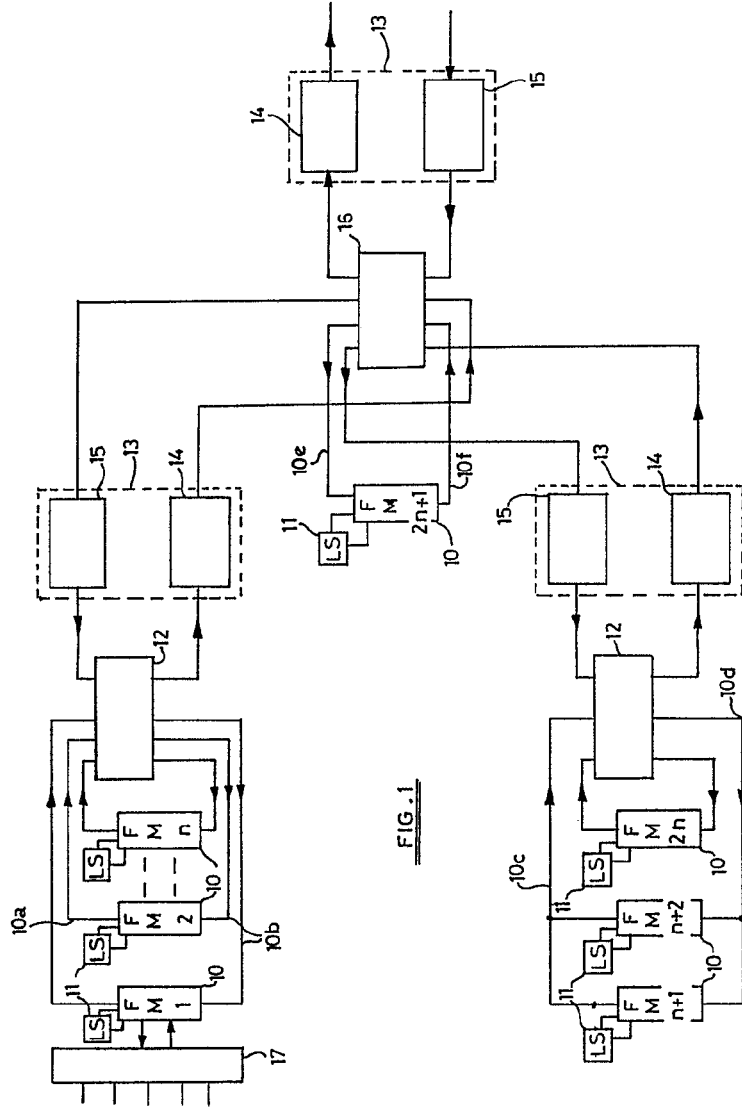


FIG. 1

Handwritten signature and date: 1953

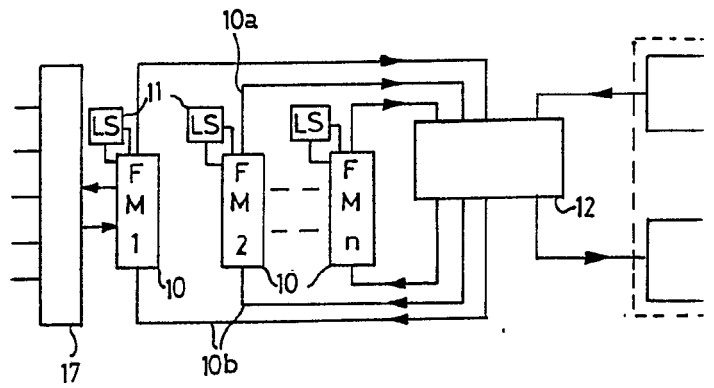
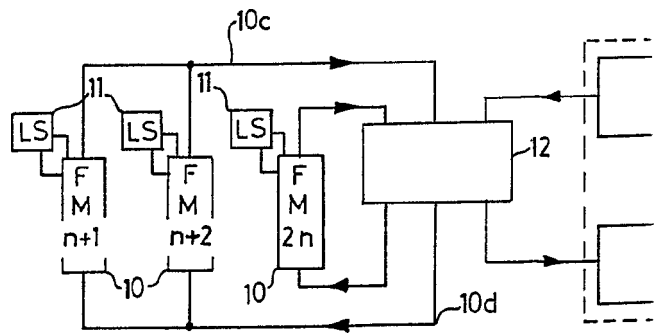
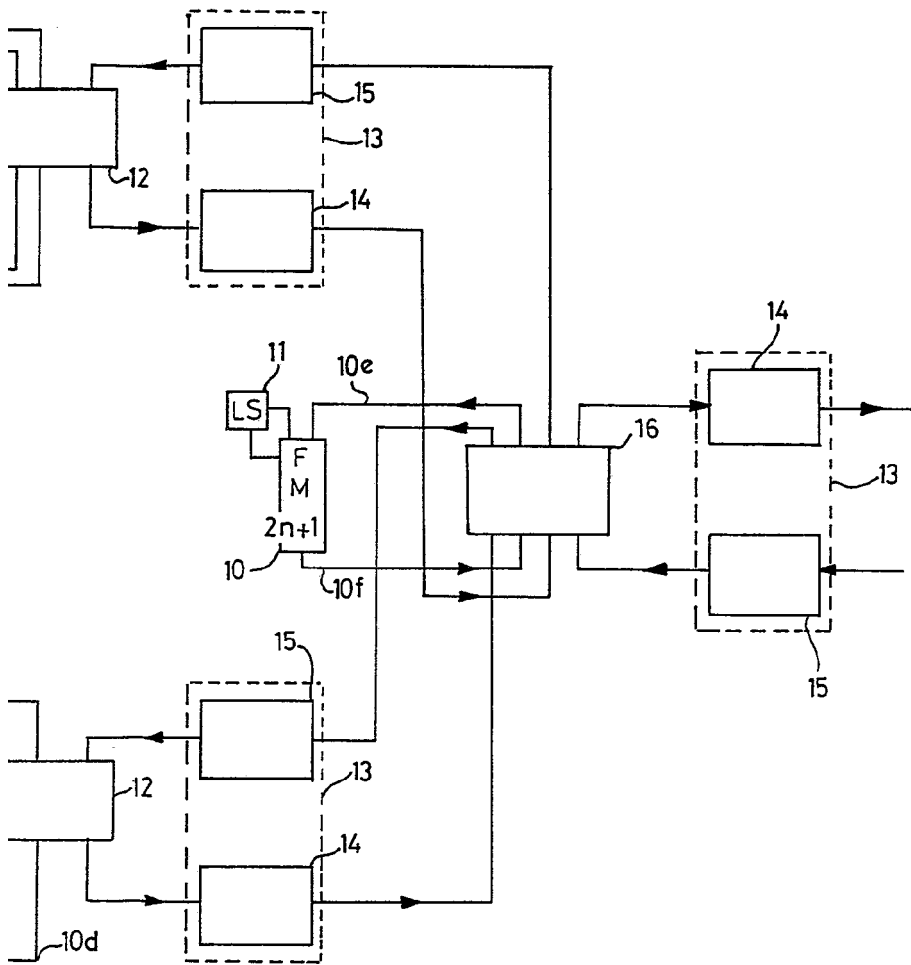


FIG. 1





Madrid 1979

FIG. 2

	$b_1$	$b_2$
(	1	0
0	0	0
1	1	1
)	0	1

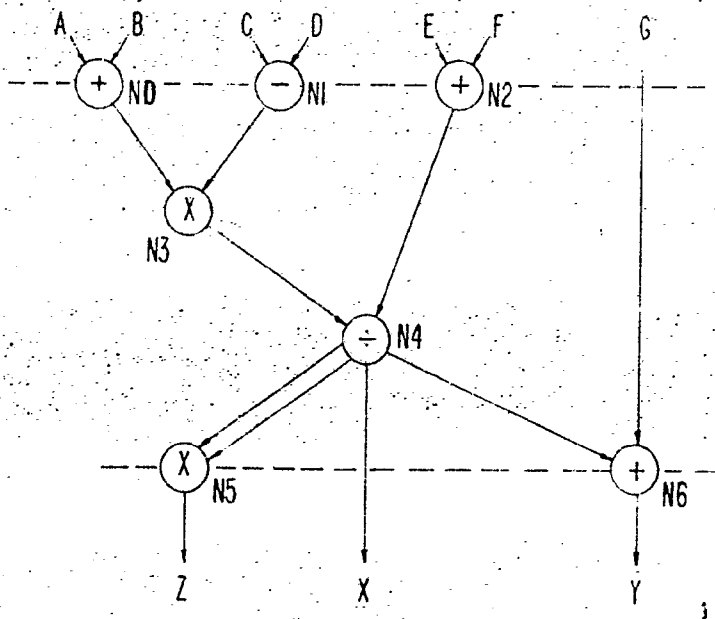
FIG. 3A

(( )5 ( ( )7( )MS0( )MSI( ) MSN( ) )

FIG. 3B

(( )0( )I( )AQ(N)7(A))

FIG. 4



POOR QUALITY

FIG. 5

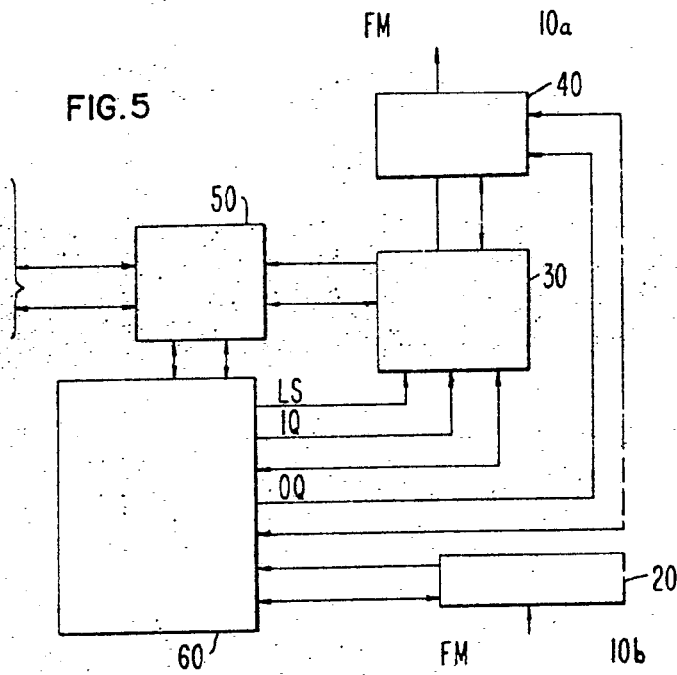
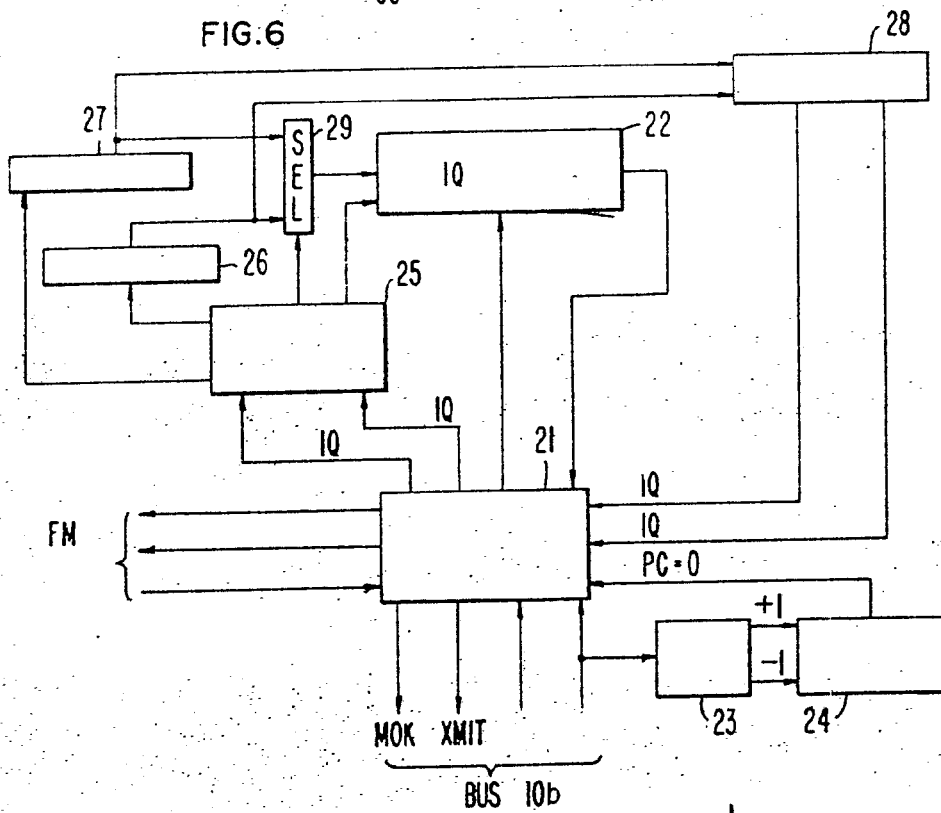


FIG. 6



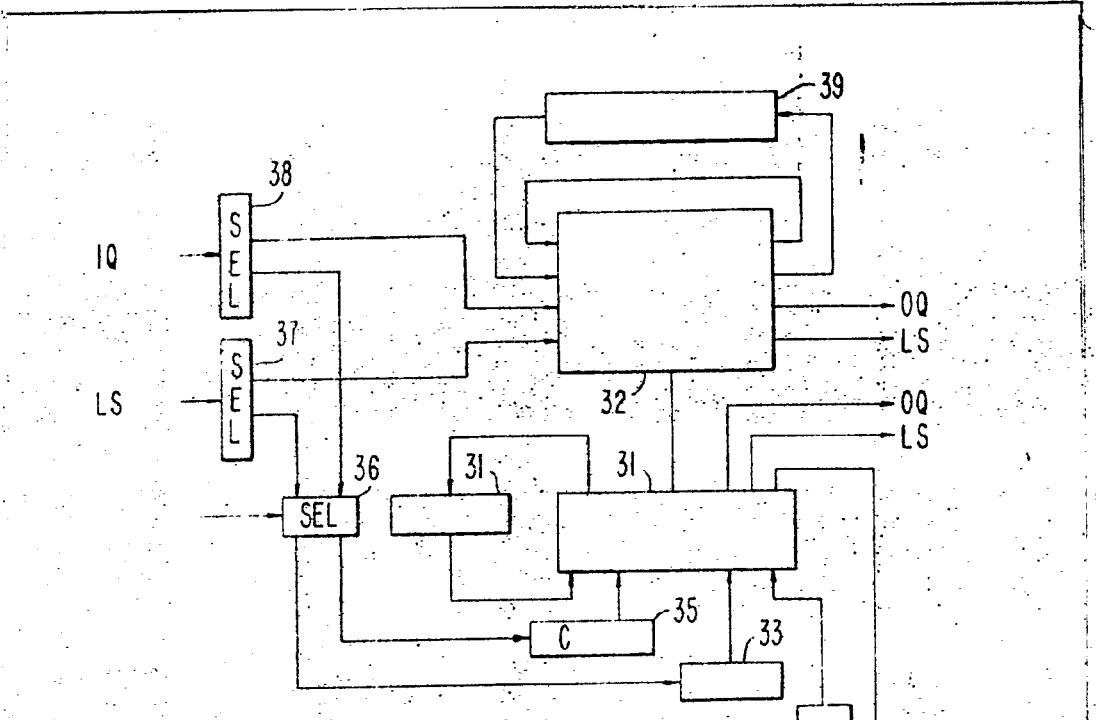


FIG. 7

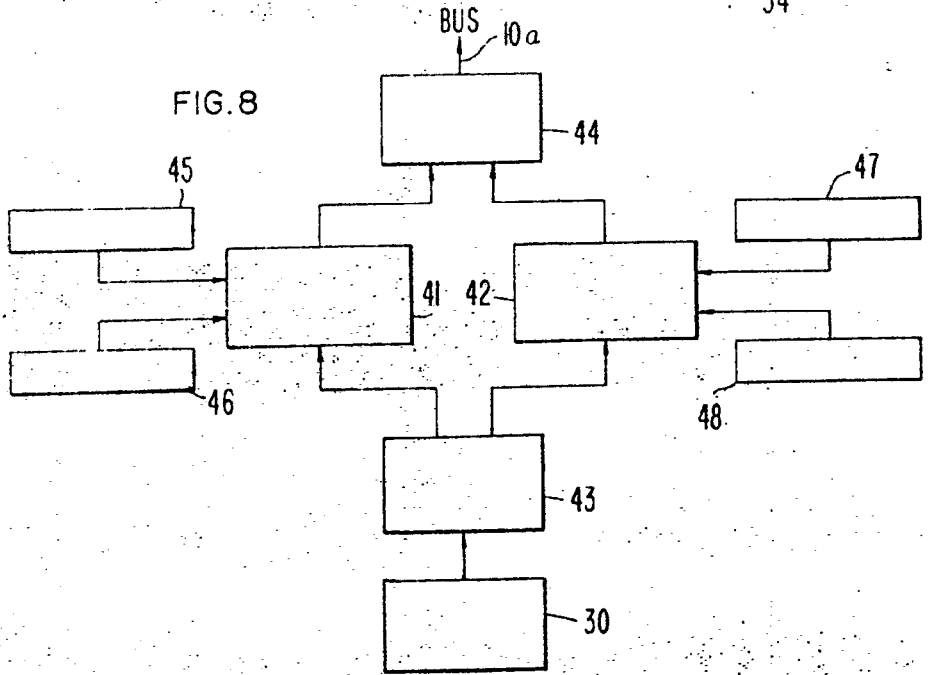
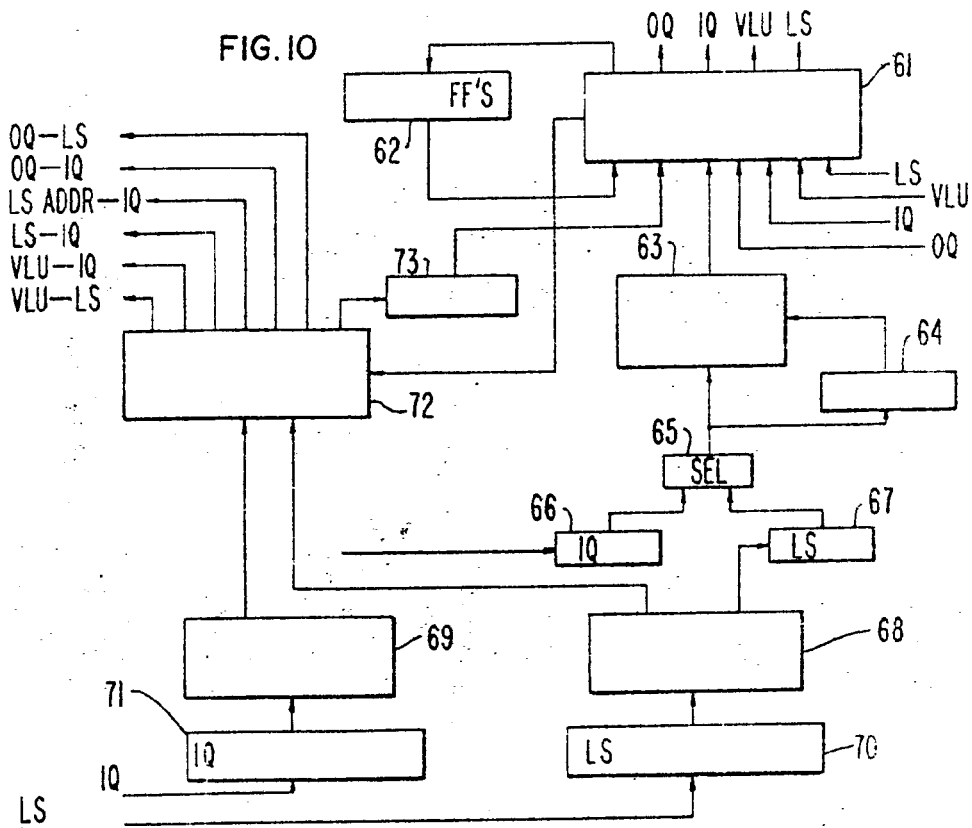
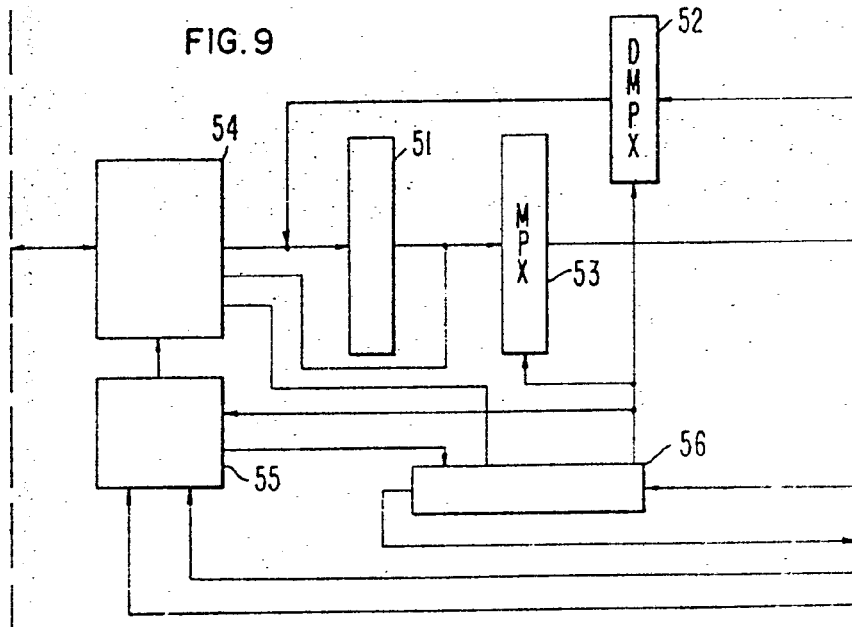


FIG. 8

POOR QUALITY



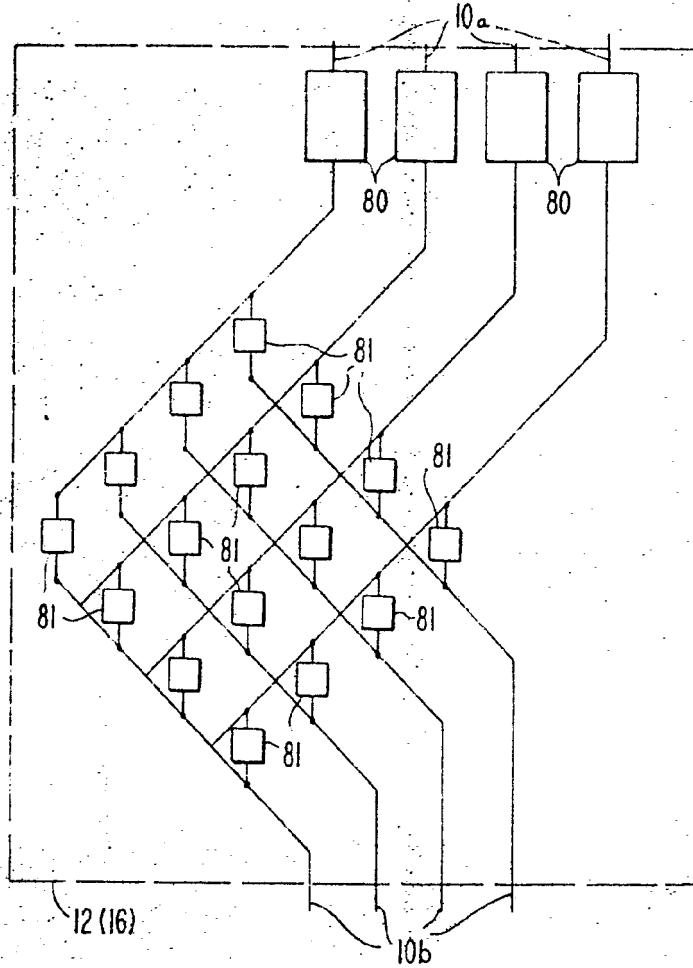


FIG. II