



PATENTE DE INVENCION

Orden nº 11.338

Int. Cl.: G06F  
432147

## *Memoria Descriptiva*

*sobre:*

PERFECCIONAMIENTOS EN SISTEMAS DE PROCESO DE DATOS.

-----

*Solicitante:* AMDAHL CORPORATION, entidad norteamericana, residente en: East Arques Avenue, 1250, SUNNYVALE, California, EE.UU. de A., y FUJITSU LIMITED, entidad japonesa, residente en Kamokodanaka, Nakahara-Ku, Kawasaki, Japón.

-----

El presente invento se refiere al campo de ordenadores digitales controlados por instrucciones y, específicamente, a métodos y aparatos asociados con las traducciones dinámicas de direcciones en sistemas para el proceso de datos.

Recientemente se han diseñado sistemas para el proceo



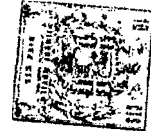
so de datos con almacenamiento virtual, que se caracterizan porque se pueden trabajar en el sistema con diferentes programas de los utilizadores. Los programas identifican lugares de almacenamiento con direcciones lógicas. Las direcciones lógicas se traducen dinámicamente a direcciones reales durante la elaboración de instrucciones. La traducción dinámica de direcciones es particularmente importante en ambientes de multi programación puesto que los programas diferentes tienen libertad para emplear las mismas direcciones lógicas.

10            Para evitar interferencias, el sistema debe traducir dichas direcciones lógicas que no son únicas a las direcciones reales que son únicas para cada programa resistente.

15            Para dar unicidad de las direcciones reales cuando se emplean direcciones lógicas que no son únicas, se utilizan tablas de traducción que son únicas para cada programa. Las tablas de traducción se almacenan normalmente, en la memoria principal. No obstante, el acceso a las tablas de traducción en la memoria principal exige una cantidad notable de tiempo que pueda degradar el funcionamiento del sistema. Para mejorar el funcionamiento cuando se hacen traducciones, es conveniente almacenar la información traducida en memorias tampón de gran velocidad para reducir el número de accesos a la memoria principal.

25            En los modernos sistemas para el proceso de datos es común el operar con una gerarquía de memoria que se caracteriza porque las memorias tampón de capacidad relativamente baja, pero de velocidad relativamente alta, funcionan en cooperación con memorias principales de capacidad relativamente grande pero de velocidad relativamente lenta. Es conveniente que la vasta mayoría de accesos bien para buscar o almacenar informa -

30



5 ción, se efectúen desde la memoria tampón con lo que se mejora el tiempo general de acceso del sistema. Para que la vasta mayoría de accesos procedan de la memoria tampón relativamente rápida, se intercambia información entre la memoria principal y la memoria tampón de acuerdo con algoritmos determinados.

10 En los sistemas de programación múltiple, de almacenamiento virtual, es también conveniente almacenar información en la memoria tampón para reducir accesos a la memoria principal. Además de las direcciones reales de datos y los propios datos, la memoria tampón almacena direcciones lógicas convenientes e identificadores de programa. Con esta información en la memoria tampón, se evitan los accesos a la memoria principal, que exige más tiempo, para la misma información.

15 La eficacia con la que una memoria tampón trabaja para reducir el tiempo de acceso del sistema general depende de un cierto número de variables. Por ejemplo, la capacidad de la memoria tampón la capacidad de la memoria principal, el régimen de transferencia de datos entre memoria, y los algoritmos de reposición que determinan cuándo se hacen transferencias entre la memoria principal y la memoria tampón.

20 Existe la necesidad de disponer de sistemas de memoria tampón perfeccionadas que sean particularmente idóneas para sistemas de proceso de datos de almacenamiento virtual y programación múltiple. De un modo específico, existe la necesidad en dichos sistemas de emplear jerarquías de memoria que tengan métodos y aparatos perfeccionados para la traducción de direcciones dinámicas.

30 El presente invento es un sistema para el proceso de datos que tiene una memoria tampón de capacidad comparativamente baja y gran velocidad y una memoria principal de velocidad



lenta y capacidad comparativamente alta. La jerarquía en la memoria se organiza como un sistema de almacenamiento virtual en el que los programas del sistema definen lugares de almacenamiento empleando direcciones lógicas. Las direcciones lógicas se traducen dinámicamente a direcciones reales durante la elaboración de instrucciones.

5

Según el presente invento, un almacenamiento identificador de programa como parte de la memoria tampón identifica qué programa del sistema tiene información de traducción dentro de la memoria tampón. El almacenamiento identificador se mantiene por lo menos con un lugar vacío. El lugar vacío facilita la introducción inmediata de información asociada con un programa del sistema que no tenga información de traducción en la memoria tampón.

10

15

El mantenimiento de por lo menos un lugar vacío para nueva información mejora el proceso de traducción puesto que un nuevo programa no tiene que esperar que se retire información antigua antes de tener acceso al almacenamiento identificador. En una modalidad el almacenamiento identificador del programa se ejecuta como una memoria localizable que tiene una gran pluralidad de lugares, 128, de los cuales solamente un pequeño número, por ejemplo 31, tienen entradas válidas en cualquier instante. El almacenamiento identificador comprende una pluralidad de campos. De un modo específico un campo válido para identificar cuál de los lugares tiene entradas válidas, un campo de traducción llamado base de segmento para almacenar información asociada únicamente con un programa particular un campo de nombre para identificar un nombre por cada entrada válida y un campo de prioridad para identificar la prioridad de las entradas válidas en el almacenamiento. El almacenamiento

20

25

30



identificador se localiza de una forma redundante en el sentido de que muchas bases de segmentos se delimitan en un sólo lugar del almacenamiento identificador. Por ejemplo, 218 bases de segmentos se identifican de una forma redundante en  $2^7$  lugares de almacenamiento identificador. La delimitación tiene lugar de acuerdo con una relación predeterminada. En el ejemplo, cada uno de los lugares especificados por las  $2^7$  direcciones representan  $2^{11}$  de las  $2^{18}$  posibles bases de segmento.

5

La memoria tampón comprende también un almacenamiento de traducción lógica que tiene una sección de índice para almacenar direcciones lógicas y nombres de programas asociados con programas que tienen información en la memoria tampón.

10

El almacenamiento de traducción lógica comprende también una parte de dato que tiene lugares los cuales mantienen una correspondencia 1 a 1 con los lugares de la parte de índice. La parte de dato almacena la dirección local asociada con la dirección lógica y el nombre del programa en el lugar de la parte de índice correspondiente.

15

La jerarquía de la memoria comprende además un almacenamiento de dato tampón que se organiza también en una parte de índice y una parte de dato asociada. La información de la parte de índice se compara con la salida de la parte de dato procedente del almacenamiento de traducción lógica.

20

En una modalidad del presente invento, el almacenamiento identificador de programa, el almacenamiento de traducción lógica y el almacenamiento de dato tampón son cada uno memorias que tienen partes de índice y partes de datos correspondientes. Además, cada parte de índice y cada parte de dato se divide en secciones primaria y alterna.

25

30

Según un aspecto del presente invento, un nombre pre



viamente utilizado se invalida y queda disponible para ser utilizado por un nuevo programa cuando el almacenamiento identificador de programa tiene todos los nombres asignados. El nombre previamente utilizado se purga del almacenamiento de traducción lógica por acceso en secuencia a todos los lugares de almacenamiento de traducción lógica buscando cualquier inscripción que sea el nombre particular previamente utilizado. La purga tiene lugar en general, bastante antes de que el nombre previamente utilizado se vuelva a asignar y, por lo tanto, al volverse a asignar, no es necesaria demora para purgar el almacenamiento de traducción lógica.

Según el resumen anterior del invento, este invento consigue el objetivo de proporcionar una jerarquía de memorias mejoradas para traducción de direcciones dinámicas en sistemas para el proceso de datos de programación múltiple.

La figura 1 representa un diagrama de conjuntos de un sistema general para el proceso de datos que incorpora los principios del presente invento.

La figura 2 ilustra una representación esquemática de una unidad de control de almacenamiento que tiene una memoria tampón primaria/alterna de nivel N empleada dentro del sistema de la figura 1.

La figura 3 ilustra una representación esquemática de una unidad de control de almacenamiento que tiene una memoria tampón primaria/alterna de dos niveles y un almacenamiento identificador de programa empleado dentro del sistema de la figura 1.

La figura 4 ilustra una representación esquemática de los detalles de un almacenamiento identificador de programa empleado dentro de la unidad de almacenamiento de la figura 3.



La figura 5 ilustra una representación esquemática de la memoria principal, la unidad de control de almacenamiento y su interfase de interconexión según se utiliza en el sistema de la figura 1.

5 La figura 6 ilustra una representación esquemática de la unidad de instrucción empleada dentro del sistema de la figura 1.

10 Las figuras 7 a 10 ilustran una representación de una parte de la máquina temporizadora en secuencias que forma una parte de la circuitería de control de la memoria tampón.

15 En la figura 1 el sistema de proceso de datos del presente invento se ilustra comprendiendo una memoria principal 2, una unidad de control de almacenamiento 4, una unidad de instrucciones 8, una unidad de ejecución 10, una unidad de canales 6 con entrada/salida y una unidad de consola 12 asociadas. El sistema de la figura 1 funciona bajo control de instrucciones del sistema donde un grupo organizado de dichas instrucciones forma un programa de sistema del utilizador. El sistema elabora más de un programa del utilizador en operaciones de programación múltiple. Las intrucciones del sistema y los datos sobre los que operan las instrucciones se introducen desde el equipo de I/O (entrada/salida) por la unidad de canal 6 a través de la unidad de control de almacenamiento 4 en la memoria principal 2. Desde la memoria principal 2, las instrucciones del sistema y los datos son buscados por la unidad de instrucciones 8 a través del control de almacenamiento 4 donde se elaboran para controlar la ejecución, por ejemplo dentro de la unidad de ejecución 10. Un programa supervisor, que es transparente a los programas de los utilizadores, funciona para supervisar la operación general del sistema.

20

25

30



La figura 2 se ilustra una modalidad de la unidad de control de almacenamiento 4 de la figura 1. La unidad de control de almacenamiento 4 es una memoria tampón que funciona para recibir, sobre la vía 362, direcciones procedentes de la unidad de instrucciones 8 de la figura 1. Las direcciones o localizaciones son normalmente localizaciones lógicas y se almacenan en el registrador de direcciones tampón (BAR) 363. El registrador 363 recibe también una entrada en la vía 354 procedente de los registradores de control dentro de la unidad de instrucciones 8 de la figura 1. La información en la vía 354 identifica únicamente el programa actual que tiene control del sistema de la figura 1 y, específicamente, la información de base de segmento que identifica las tablas de traducción empleadas para convertir direcciones lógicas en direcciones reales.

La información de base de segmento en la vía 354 comprende el tamaño de página y el tamaño de segmento, la dirección real en la memoria principal en la que se localiza la tabla de segmentos (origen de tablas de segmentos) y la longitud de la tabla de segmentos. De esta información en conjuntos se denomina información de base de segmentos. Los bits de orden inferior del origen de la tabla de segmentos forman la corriente de entrada para localizar el almacenamiento identificador del programa 155. El resto de la información de base de segmentos es la corriente de entrada al registrador 393 y, a su vez, a los comparadores 317 y 321.

El almacenamiento identificador de programa (PIS) 155 comprende una parte de índice y una parte de dato. Tanto la parte de índice como la parte de dato se dividen, a su vez, en una sección primaria y una sección alterna. Específicamen-



5 te, la parte de índice comprende la sección primaria 319 y la sección alterna 320. La parte de dato comprende una sección primaria 323 y una sección alterna 324. Las cuatro secciones del almacenamiento 155 se localizan a través de los accesos de entrada de direcciones o localizaciones (AI). Los bitios de orden superior del origen de la tabla de segmentos y la información de tamaño para una pluralidad de programas se almacena en la parte de índice del almacenamiento 155.

10 Dicha información se almacena a través del acceso de entrada de datos (DI). Cuando el programa que tiene actualmente el control del sistema de proceso de datos de la figura 1 recibe inicialmente el control, su información de base de segmento es la corriente de entrada por la vía 354 del registrador 363. Los bitios de dirección de orden inferior son la corriente de entrada para localizar el almacenamiento 155 y el resto de la información de la base de segmento en la corriente de entrada al registrador 393 y a los comparadores 317 y 321. La información que se toma lectura a través de los accesos de salida de datos (DO) de las secciones 319 y 320 es también la corriente de entrada a los comparadores 317 y 321, respectivamente. Los comparadores determinan si el programa actual en control tiene o no información dentro del almacenamiento 155.

15 En la parte de dato del almacenamiento 155, el generador de nombres 264 introduce nombres identificadores a través del acceso de entrada de datos (DI). Las secciones primaria y alterna 323 y 324 de la parte de datos son la corriente de entrada al registrador 379. Una u otra de las secciones 323 ó 324 se elige en función a una salida de señal de comparación de los comparadores 317 y 321, respectivamente. Cuando se elige, un nombre único asociado con el programa en control pasa

20

25

30



por puerta al registrador 379. El registrador 379 comprende también los bitios de dirección lógica de orden superior.

El almacenamiento de traducción lógica (LTS) 255 comprende también una sección primaria 381 y una sección alterna 382 que forman juntas una parte de índice y una sección primaria 256 y una sección alterna 357 que forman juntas una parte de dato. Los bitios de orden inferior de la dirección lógica del registrador 363 se utilizan para localizar cada una de las secciones en el almacenamiento 357 a través de los accesos de entrada de direcciones (AI). La parte de índice del almacenamiento 255 se carga con los bitios de orden superior de la dirección lógica y el nombre se introduce por puerta desde el registrador 379 hasta el acceso de entrada de datos (DI).

La parte de dato de almacenamiento 255 se carga con direcciones reales que se traducen y corresponden a las direcciones lógicas almacenadas en la parte de índice del almacenamiento 255. Las direcciones reales se cargan a través del acceso de entrada de datos (DI) cuando se reciben del almacenamiento principal a través de los registradores de datos 384 y 385. Cuando se localiza el almacenamiento de traducción lógica 255 funciona para comparar el nombre y dirección lógica del registrador 379 en los comparadores 328 y 329 con el contenido del lugar localizado en la parte de índice de la memoria.

Si se halla la comparación, el comparador 328 ó 329, cualquiera que sea la comparación, hace que la dirección real procedente de la parte de dato del almacenamiento 255 se introduzca por puerta en el registrador 359. La dirección real en el registrador 359 es la corriente de entrada a los comparadores 356 y 357. Si no se halla comparación, el almacenamiento principal se localiza para tener acceso a las tablas de traduc



ción deseadas que darán por resultado la dirección real que en  
tonces se almacena en la parte de dato del almacenamiento 255.

5 El almacenamiento tampón de datos (BDS) 355 compren-  
de una parte de índice y una parte de dato. La sección prima-  
ria 365 y la sección alterna 366 comprenden la parte de índice  
del almacenamiento 355. La sección primaria 367 y la sección  
alterna 368 forman la parte de dato del almacenamiento 355.  
Las cuatro secciones del almacenamiento 355 se localizan por  
parte de los bitios de orden superior de la dirección lógica  
10 procedente del registrador 363. Los bitios de dirección de or-  
den inferior de la localización lógica son idénticos a los bi-  
tios de dirección de orden inferior de la dirección real.

15 Cuando se localiza el almacenamiento 355, las direc-  
ciones reales procedentes de la parte de índice son la corrien-  
te de entrada a los comparadores 356 y 357 y si se halla una  
comparación el dato correspondiente procedente de la parte de  
dato es la corriente de entrada al registrador 387. El dato en  
tonces en el registrador 387, es el dato originalmente locali-  
zado por la dirección lógica en el registrador 363. Si no se  
20 halla coincidencia en el almacenamiento de dato 355, parte de  
índice, entonces se localiza la memoria principal para tener  
acceso a la información deseada y se introduce en la parte de  
dato del almacenamiento 355, mientras que los bitios de direc-  
ción real de orden superior se cargan de un modo similar en la  
25 parte de índice. La información en el registrador 387 se trans  
mite a la unidad apropiada en la figura 1 (unidad I, unidad E  
o unidad C) donde es utilizada por el sistema.

30 En la figura 3, se ilustra una modalidad de unidad  
de control de almacenamiento 4 para utilizarse en el sistema  
de la figura 1. La unidad de almacenamiento de la figura 3 es



una memoria tampón y recibe direcciones procedentes de la unidad de instrucciones S por la vía 362, desde la unidad de canales por la vía 353 o desde un elaborador externo por la vía 309. La vía 309 se utiliza en una configuración de elaboración múltiple del sistema de la figura 1. Cuando se recibe una dirección lógica de entrada en una de las vías indicadas, la memoria tampón de la figura 3 funciona para tomar lectura del dato localizado en uno de los registradores de salida 387 a 391. Si la información localizada lógicamente no está disponible en la memoria tampón de la figura 3, entonces después de la traducción de dirección real de la información se lleva por la vía 809' al almacenamiento principal 2 de la figura 1 para tener acceso a la información localizada.

La información a la que se tiene acceso queda disponible en la vía de salida de datos de la memoria principal 811. Cuando la memoria tampón de la figura 3 tiene información que se ha de transferir a la memoria principal, se saca por la vía 808.

Las direcciones de entrada a la memoria tampón desde vías 353, 362 y 309 se almacenan en el registrador de direcciones tampón (BAR) 363. El registrador 363 almacena normalmente 24 bitios de dirección (BITS 8-31) y 5 bitios identificadores de programa (BITS 32-36) por un total de 29 bitios. El registrador 363 tiene su salida conectada a muchos lugares. De un modo específico, los 29 bitios se conectan al registrador de direcciones B2 (B2AR) 378, el registrador de direcciones de búsqueda de instrucciones (IFAR) 374, el registrador de direcciones de operados (OPAR) 375, y el registrador de dirección de unidades de canal (CUAR) 376, 8 bitios se introducen en el registrador de números de segmentos (SNR) 261 y 9 bitios se in



5      troducen en el registrador del número de páginas (PNR) 262. Veinticuatro bitios procedentes del registrador 363 (BITIOS 8-26 y los 5 bitios de nombre identificador) se introducen en el registrador de direcciones lógicas (LAR) 379. Siete de los bitios de direcciones procedentes del registrador 363 (BITIOS 14-20) se introducen en los accesos de direcciones (AI) del almacenamiento de traducción lógica 255.

10      El almacenamiento de traducción lógica 255 es un directorio de gran velocidad (HSD) que comprende una parte de índice que tiene una sección primaria 381 y una sección alterna 382.

15      Cada una de las secciones 381 y 382 comprenden normalmente 128 lugares de 11 bitios cada uno. Los 128 lugares en cada sección se localizan simultáneamente por parte de los bitios de localización de entrada procedentes del registrador 363 (BITIOS 14-20).

20      En el lugar localizado se almacenan 11 bitios a través del acceso de entrada de datos (DI). Los 11 bitios comprenden 6 bitios de localización de orden superior (bitios 8-13) y los 5 bitios de nombre identificador procedentes del registrador 363. La parte de índice, que comprende secciones primaria y alterna 381 y 382, funciona para almacenar estos 11 bitios para programas del sistema que tienen el control del sistema de la figura 1. Cuando un nuevo programa recibe control del sistema de la figura 1, su nombre identificador y los bitios de localización del orden superior de una dirección particular se transfieren desde el registrador 363 al registrador 379 que tiene una salida conectada a los comparadores 328 y 329.

30      Como es lógico, se puede emplear un programa y su nombre identificador con muchas direcciones diferenciales. Los



comparadores 328 y 329 comparan los bitios de dirección lógica de orden superior y los bitios del nombre identificador procedentes de las secciones primaria y alterna 381 y 382 según salen como impulsos cíclicos por los accesos de salida o datos (DO), respectivamente. De esta manera, se hace una comparación para determinar si la dirección lógica actual para el programa actual en control se encuentra en la memoria tampón en la figura 3.

La memoria de traducción lógica 255 comprende también una parte de dato que tiene una sección primaria 356 y una sección alterna 357. Cada una de las secciones 356 y 357 comprenden normalmente 128 lugares de 13 bitios cada una. Cada una de las secciones se localiza por medio de los mismos 7 bitios que la parte de índice para almacenar información por el acceso de entrada de datos (DI) o sacar datos por los accesos de salida de datos (DO). La información que sale de los almacenamientos 356 y 357 se introducen en el registrador de direcciones reales (RAR) 359. El registrador 359 recibe la información de la sección primaria 356 o de la sección alterna 357 en función de una comparación detectada por los comparadores 328 ó 329, respectivamente. Asimismo, el registrador de direcciones reales 359 puede cargarse directamente con el contenido del registrador de direcciones lógicas 379. Los 13 bitios en las secciones primaria y alterna 356 y 357 se cargan desde el registrador de traducción 387 a través del acceso de entrada de datos (DI). Los 13 bitios procedentes del registrador de traducción 387 son los 13 bitios de dirección real de orden superior (BITIOS 8-20) que corresponden a la dirección lógica en la misma dirección en la parte de índice del almacenamiento 255.

Los bitios 19 y 20 salen del registrador de dirección



nes reales 359 y entran en un decodificador normal 351 que forma cuatro líneas de selección que se activan de una en una.

Asimismo la salida del registrador 359 consiste en 19 bitios (BITIOS 9-26) que se introducen en el registrador de direcciones de la memoria principal (NSAR) 364. Cuando la información localizada no está en la memoria tampón de la figura 3, la dirección real procedente del registrador 364 se utiliza para iniciar un acceso a la memoria principal 2 de la figura 1 de donde se obtiene finalmente la información deseada.

El registrador de direcciones tampón 363 conecta también 8 bitios como entrada al almacenamiento de datos tampón 355. El almacenamiento 355 comprende una parte de índice que tiene una sección primaria 365 y una sección alterna 366. Cada una de las secciones de índice 365 y 366 comprenden normalmente 256 lugares con 15 bitios por lugar. Cada uno de los lugares es localizado por 6 bitios de dirección lógica de orden inferior (BITIOS 21-26). Cada uno de los lugares en las secciones primaria y alterna 365 y 366 queda disponible para almacenamiento en una clave de almacenamiento principal de 5 bitios (NSKEY) y 11 de los 19 en el registrador 359. Los bitios 19 y 20 procedentes del registrador 359 se introducen en el decodificador 331 y los 11 bitios de orden superior quedan disponibles a través del acceso de entrada de datos (DI) para almacenamiento en las secciones 365 y 366. Cada una de las secciones 365 y 366, cuando se localizan, sacan cuatro grupos de 16 bitios que cada uno comprende 11 bitios de dirección y 5 bitios clave. Las cuatro salidas pasan por puerta a través de los accesos de salida de datos (DO) a los circuitos de selección 396 y 397, respectivamente. Los circuitos de selección 396 y 397 funcionan para elegir una de las cuatro salidas por cada una



de las secciones 365 y 366 en respuesta a una de las cuatro sa-  
lidas del decodificador 331. Las salidas elegidas procedentes  
de los circuitos de selección 396 y 397 entran en los compara-  
dores 336 y 337, respectivamente. Los comparadores 336 y 337  
5 actúan para comparar la dirección real actual en el registra-  
dor 359 con una dirección real utilizada previamente en las  
secciones 365 y 366. Si se halla una comparación, los compara-  
dores 336 y 337 funcionan para elegir la sección primaria 367  
o la sección alterna 368, respectivamente.

10 La sección primaria 367 y la sección alterna 368 com-  
prende cada una normalmente 256 lugares de 256 bitios cada  
uno. Los 256 lugares para cada una de las secciones de datos  
367 y 368 se localizan cada una por 8 bitios de dirección lógi-  
ca de orden inferior (BITIOS 21-28) y cada una proporciona en  
15 respuesta en los accesos de salida de datos (DO) cuatro luga-  
res parciales de dato de una vez (64 bitios por lugares parcia-  
les). Un lugar completo comprende 256 bitios de datos y se lo-  
caliza por los bitios 21-26. Desde estos 256 bitios de dato,  
los bitios 27-28 eligen 64 bitios de dato que salen a las puer-  
20 tas de selección 398 y 399. Las cuatro salidas de las seccio-  
nes 367 y 368 se introducen en los circuitos de selección 398  
y 399, respectivamente. Los circuitos de selección 398 y 399  
se eligen para proporcionar una salida en respuesta a la sali-  
da elegida de la salida del decodificador 331. La salida de  
25 los circuitos de selección 398 y 399 se introducen, por circui-  
tos de alineación de búsqueda 372 y 373, respectivamente, a  
uno o más de los cinco registradores de salida 387 a 391. La  
determinación de si se elige la salida primaria o alterna se  
controla mediante los comparadores 336 y 337 cada uno de los  
30 cuales tiene una entrada a los registradores 387 a 391.

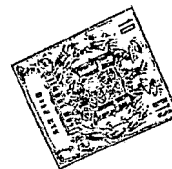


5 El almacenamiento 355 se emplea para determinar si el lugar actualmente localizado, especificado por la localización del registrador 359 tiene o no la misma dirección en la sección de índice 365 ó 366. Si es así, el comparador 336 o el comparador 337 detecta la correspondencia y el dato correspondiente en las secciones de datos 367 ó 368, respectivamente se introduce en uno de los registradores apropiados 387 a 391. El dato se carga en las secciones de datos 367 y 368 por la vía de 64 bitios desde el registrador 385 a través del acceso de entrada de datos (DI).

10 El acceso de entrada de datos se recibe 64 bitios de datos desde la vía de 64 bitios conectada al registrador de dato tampón de gran velocidad (HSBDR) 385. Las puertas de selección 369 se eligen para derivar las secciones de datos 367 y 15 368 que pasan informaciones en el registrador 385 directamente a través de una circuitería de alineación de búsqueda (FETAL) 372 a uno o más de los registradores 387 a 391.

20 El registrador 385 se conecta para recibir información procedente de la memoria principal por el registrador de datos de la memoria principal (MSDR) 384 y desde una circuitería de alineación de almacenamiento (ETOAL) 383 desde la puerta de selección 386. Las puertas de selección 386 se eligen in 25 formación principalmente de la unidad de instrucciones 8 por la vía 352 o desde la unidad de canales por la vía 358. La salida de información de los circuitos de selección 398 y 399 se introducen en el registrador 384. La salida procedente del registrador 384 vuelve también a la memoria principal por la vía 808 para transmitir de nuevo información a la unidad de memoria principal 2 de la figura 1.

30 Los registradores de salida 387 a 391 reciben infor-



mación desde los almacenamientos 367 y 368. El registrador de traducción 387 es un registrador normal de 32 bitios y se emplea junto con la traducción de dirección dinámica requerida para convertir una dirección lógica en una dirección real.

5

El registrador de palabras de instrucciones (IWR) 358 se utiliza junto con la transmisión de palabras de instrucción a la unidad de instrucciones 8 por la vía 396. El registrador de palabras de operandos (OWR) 399 se utiliza junto con la transmisión de operandos a la unidad de ejecución 10 por la vía 395. El registrador de palabras de canal (OWR) 390 se utiliza junto con la transmisión de información por la vía 394 a la unidad de canales 6. El registrador de errores (ERR) 391 se utiliza junto con la circuitería de detección y corrección de errores.

10

15

Los nombres para los identificadores empleados para distinguir programas diferentes capaces de funcionar en el sistema de la figura 1 se obtienen gracias al almacenamiento de identificadores de programa 340. El almacenamiento 340 recibe información de base de segmento desde los registradores de control en la unidad de instrucciones 8 por la vía de bitios 354. El almacenamiento 340 comprende normalmente 128 lugares para almacenar información asociados con hasta 31 programas simultáneamente. Un nombre identificador del programa 32 se mantiene sin utilizar para estar siempre disponible para asignación a un nuevo programa que no se encuentre simultáneamente en almacenamiento 340. El almacenamiento 340 proporciona el nombre identificador de programa en la vía de 3 bitios 392 que se introduce al registrador de dirección tampón 363. Más detalle del almacenamiento 340 se describirá con relación a la figura 4 más adelante.

20

25

30



Los registradores 374 a 377 junto con el registrador de traducción (TRR) 387 se eligen a través de puertas de selección 380 como entradas a la sumadora de líneas 360 o la sumadora de bytes 361. El registrador de dirección B2 (B2AR) 378 se  
5 emplea como entrada a registrador de direcciones de búsqueda previa (PFAR) 377 para selección por parte de las puertas 380. Asimismo, el registrador 378 se puede elegir por medio de la puerta 333 para introducción cíclica en el registrador de palabras de operando (OWR) 389 o el registrador de palabras de canales (CWR) 390. La sumadora 360 funciona para aumentar la dirección completa en incremento de 0,32 a 2048, y la sumadora  
10 361 funciona para incrementar la localización de "bytes" en incrementos de 0,+4 ó +8, respectivamente. La dirección de la ventana se saca por una vía de dirección de 29 bitios como entrada al registrador de dirección tampón 363.  
15

La palabra de traducción de almacenamiento en la vía 358 se puede elegir también por medio de las puertas 333 para introducirse en el registrador TRR 387.

Las puertas de selección 333, cuando se eligen, conectan también la información de base de segmentos CPU por la vía 354, la clave de 3 bitios procedentes del registrador 384,  
20 y la información de base de segmento de la unidad de canales por la vía 358 a los registradores 387, 389 y 390.

En la figura 4 el almacenamiento de identificadores de programa 340 empleado en la memoria tampón de la figura 3,  
25 se ilustra con detalle. El almacenamiento de la figura 4 recibe la información de base de segmentos por la vía 354 desde los registradores de control CRO y CR1 de la unidad de instrucciones. Los 30 bitios en la vía 354 comprenden 4 bitios procedentes del registrador de control (CRO). Dichos cuatro bitios  
30



son los bitios 8 y 9 que definen el tamaño de la página como de 2Kbytes ó 4Kbytes. Adicionalmente, los bitios 11 y 12 definen el tamaño de segmentos como de 64Kbytes ó 1Mbytes. La vía 354 comprende también 26 bitios del registrador de control

5 CRI. De un modo específico, los bitios 0 a 7 definen la longitud de la tabla de segmentos y los bitios 8 a 23 definen la dirección de la tabla de segmentos, que es el origen de la tabla de segmentos.

10 Cada programa en el sistema de proceso de datos tiene normalmente un origen de tablas de segmentos diferentes por lo que el origen define de una forma única el programa que tiene control del sistema de proceso de datos. Los bitios 8 a 25 entran en una circuitería de recodificación o de copia 287 que delimita los 18 bitios de entrada en 7 bitios de salida. Es

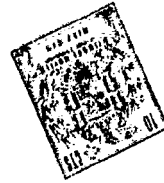
15 preferible que la delimitación sea casual aunque una vez elegida no cambie.

Cada una de las direcciones especificadas por los 7 bitios de salida representa  $2^{11}$  lugares de dirección especificados por los 18 bitios de entrada. Los 7 bitios de dirección de salida de la circuitería 287 son elegidos por puertas 288 y se almacenan en un registrador de direcciones 289. El registrador de direcciones 289 emplea los 7 bitios de dirección para localizar la memoria de la base de segmento (SB) 291. Asimismo, la dirección en el registrador 289 puede aumentar en uno

20 en el incrementador 290 y ser elegida por puertas 288 para proporcionar una nueva dirección incrementada en el registrador 289.

La memoria SB 291 comprende 128 lugares de 41 bitios por lugar. Treinta de los bitios se derivan de la vía 354 y

30 contienen el tamaño de segmento, tamaño de página, dirección



de origen de la tabla de segmentos y longitud de la tabla de  
segmentos. Además, cada lugar comprende un identificador de 5  
bitios para dar nombre a cada anotación o introducción en la  
memoria 291. Asimismo, cada lugar comprende un campo de priori-  
5 dad de 5 bitios para establecer la prioridad de introducciones  
en la memoria 291. Adicionalmente, en cada lugar hay presente  
un campo válido de un bitio. La información se lee de la memo-  
ria 291 en un número de registradores cuando la memoria es lo-  
calizada por la dirección en el registrador 289. El bitio váli-  
10 do se lee en registradores VA 210 y VA 209 para utilizarse en  
el establecimiento de la validez de la información leída.

El campo de validez en la memoria 291 se emplea para  
identificar hasta 31 anotaciones válidas del total de 128 luga-  
res. El registrador de VA 210 tiene un cronometrador controla-  
15 do y solamente se carga al ordenárselo. El registrador VA 209  
es de marcha libre y se carga cada vez que se localiza la memo-  
ria 291. Los registradores 210 y 209 se utilizan junto con el  
incrementador localizador alterno 290. Siempre que la memoria  
291 tiene inicialmente acceso por parte de una dirección proce-  
20 dente del registrador 289 y la información localizada no se si-  
túa en la memoria 291. La memoria vuelve a localizarse por par-  
te de la siguiente dirección en la secuencia.

De un modo específico, el que la memoria 291 no haya  
tenido acceso respecto a la información localizada queda indi-  
25 cado por la ausencia de una señal FND de la puerta Y 282. La  
ausencia de una señal desde la puerta 282 hace que el circuito  
1 indicado por la referencia 290 incremente la dirección en el  
registrador 289 a la dirección siguiente. Si se halla la infor-  
mación deseada en este lugar entonces el nombre apropiado pasa  
30 por puerta a través de los registradores 294 y 295, y es elegi



do por la puerta 206 como salida al registrador de direcciones  
tampón. Si la información de base de segmento no se sitúa en  
la memoria en el segundo lugar, entonces dicha información, un  
nombre y una prioridad deben cargarse en la memoria 291 en uno  
5 de los dos lugares a los que se ha llegado previamente. Los bi-  
tios de validez, según se almacenan en los registradores 210 y  
209 en combinación con la información de prioridad en los cam-  
pos de prioridad por cada lugar de la memoria, se emplean para  
determinar cuándo se almacena una nueva entrada o anotación en  
10 la memoria 291.

Cuando la memoria 291 se localiza, se toma lectura  
de la información de base de segmento del registrador de datos  
292 de 30 bitios. La información de bases de segmentos en el  
registrador 292 se compara con la información de bases de seg-  
15 mentos actual en la vía 354 en un comparador 293 de 30 bitios.  
Si la información SB de la memoria y la información de la vía  
354 se comparan y se establece un bitio válido en el registra-  
dor VB 209, entonces la puerta Y 282 proporciona una señal FND  
indicativa de que la información de base de segmentos actual  
20 se ha introducido previamente en la memoria 291.

Siempre que la memoria 291 se localiza el nombre de  
anotación a la que se ha tenido acceso se introduce en el re-  
gistrador IA de marcha libre 294.

Si el acceso a la memoria da por resultado una indi-  
25 cación FND de la puerta 282 o se exige la escritura de una nue-  
va anotación, el nombre "hallado" o el nuevo nombre sale del  
registrador IA 294 y se guarda en el registrador ID 205.

Desde este punto el nombre pasa por puerta por el se-  
lector 206 a la vía 392 para entrada al registrador de direc-  
30 ciones tampón 363 en la figura 3.



5 Cuando la escritura de una nueva anotación en la memoria 291 produce el desplazamiento de una anotación antigua, se debe conseguir primero acceso a la memoria y moverse la anotación desplazada desde el registrador IA 294 y guardarse en el registrador IAB 204 y el registrador IB 205. Este desplazamiento se produce cuando una nueva anotación debe escribirse en un lugar previamente lleno, o cuando la memoria 291 ya contiene 31 anotación de programa. Desde el registrador IE 205 el nombre desplazado pasa temporalmente por acción del selector 10 206 a la vía 392 para entrar en el registrador de localización tampón 363 de la figura 3. Después se utilizará para invalidar todas las anotaciones de almacenamiento de traducción lógica que tienen el mismo nombre. El nombre desplazado conservado en el registrador de IAB 284 es también el nombre siguiente asignado a una nueva anotación de programa. El registrador CTB 207 15 contiene el nombre asignado a una nueva anotación de programa cuando el registrador IAB 204 está vacío. Inicialmente el registrador CTB 207 se coloca en la primera de una secuencia de 32 nombre (nombre "cero"). Si el nombre en el registrador CTB 20 207 se utiliza cuando se hace una nueva anotación de programa en la memoria 291, el nombre siguiente de la frecuencia se genera en el incrementador 208 y el registrador CTB 207 se pone al corriente.

25 Cuando un programa está a punto de conseguir el control del sistema, la prioridad actual de su anotación de programa si existe una, o la prioridad 31 (la inferior) si debe hacerse una nueva asociación de programa, se guarda en el registrador PAA 298. Cuando se devuelve el control del sistema al programa, la prioridad de su anotación de programa se establece en cero -la superior- y su nombre se conserva en el re- 30



gistrador IAA 295. La prioridad debe ahora ponerse al corriente en la memoria 291. La memoria 291 se localiza en cada lugar por ordenación en secuencia a través de todas las combinaciones de dirección especificables por parte del registrador CTA

5 289. Cada vez que se tiene acceso a una anotación válida de programa su nombre se fija en el registrador IA 294 y su prioridad en el registrador PA 297. El registrador IA 294 se compara con el registrador IAA 295 en el comparador 296. Si son iguales, o sea, se ha conseguido acceso a la anotación de programa hecha con inmediata anterioridad, no es necesario poner

10 a corriente la prioridad. Si el registrador IA 294 y el registrador IAA 295 son diferentes, el registrador PA 297 se compara con el registrador PAA 298 en el comparador 299. Si el registrador PA 297 es menor que el registrador PAA 298, la anotación del programa a la que se ha tenido acceso era anteriormente

15 de prioridad mayor que el programa actualmente con control del sistema (y ahora a la máxima prioridad). El registrador PA 297 se incrementa entonces en la circuitería de prioridad 202 y se coloca en el registrador PB 203. Desde este punto se vuelve a escribir en la memoria 291 para reducir la prioridad de entrada del programa conseguido. Si el contaje de prioridad en el registrador PA 297 es mayor que el contaje en el registrador PA 298, la circuitería 202 no cambia el valor del registrador PA 297 colocado en el registrador PB 203, y la prioridad

20 de entrada del programa conseguido permanece igual.

En la figura 5 se ilustra un diagrama de conjuntos general de la memoria principal 2 de la figura 1. La unidad de control de almacenamiento 4 comunica 81 bitios de dato por la vía 808, 19 bitios de localización por la vía 809, 30 bitios

30 de control por la vía 810 a la unidad de tráfico de vías 805



5 en la memoria principal 2. La unidad de tráfico de vías 805 de  
vuelve 81 bitios de dato por la vía 811 a la unidad de control  
de almacenamiento 4. Las vías de datos 808 y 811 comprenden en  
general 64 bitios de dato, 9 bitios de código de corrección de  
error asociado, 5 bitios clave, un bitio de paridad asociado y  
2 bitios de control adicionales. La vía de dirección 809 se ha  
indicado como si comprendiera 19 bitios de dirección. No obs-  
tante, el número de bitios de localización cambia dependiendo  
del tamaño de la memoria principal 2. En una configuración me-  
10 nor son normales 16 bitios de dirección.

15 En la figura 6 la unidad de instrucciones 8 de la fi-  
gura 1 se ilustra con detalle. Una pluralidad de 14 registrado-  
res 310 a 316 funcionan para introducir información en una su-  
madora de direcciones efectivas 318 que tienen salida a un re-  
registrador efectivo de direcciones 322. El registrador 322 ali-  
menta al registrador de direcciones tampón de la unidad de al-  
macenamiento por la vía 362. La unidad de instrucciones 8 com-  
prende también paquetes de registradores pares e impares 338 y  
339 que se alimentan por registradores 334 y 335 con salida a  
20 los registradores 341 y 342. Una pluralidad de registradores  
de control 344 a 348 se emplea para almacenar diversa informa-  
ción de control. De un modo específico, el registrador CR-0  
344 y el registrador CR-1 345 se utilizan junto con traducción  
localizada dinámica y alimentan su información en una vía de  
25 30 bitios 354 a la unidad de almacenamiento de las figuras 2 y  
3.

#### FUNCIONAMIENTO

30 La unidad de almacenamiento de la figura 3 funciona  
para conseguir traducción de dirección dinámica cada vez que  
se hace una referencia a almacenamiento. Las traducciones tie-



nen lugar en partes de direcciones llamadas "segmentos" donde los segmentos se dividen además en bloques llamados "páginas".

Un segmento es normalmente un bloque de direcciones lógicas de secuencias que abarcan 65.536 ó 1.048.566 bytes.

5 El segmento comienza en una dirección que es un múltiplo de su tamaño. El tamaño del segmento se controla por bitios 11 y 12 del registrador de control cero que está situado en la unidad de instrucciones 8 del sistema de la figura 1.

10 Una página es normalmente un bloque de almacenamiento continuo que contiene 2048 ó 4096 bytes. Una página comienza en una dirección que es un múltiplo de su tamaño. El tamaño de la página está determinado por los bitios 8 y 9 en el registrador de control cero (CRO). Cada dirección lógica se divide en un campo de índice de segmento, un campo de índice de páginas y un campo de índice de bytes.

15 Para hacer la traducción de dirección lógica a dirección real, se emplean dos tablas de traducciones. Las tablas de traducciones se almacenan normalmente en la memoria principal 2. La parte de índice de segmentos de cada dirección lógica se utiliza para elegir una entrada desde una tabla de segmentos donde se especifican la dirección inicial y la longitud de la tabla de segmentos por el contenido del registrador de control 1 (CR1) en la unidad de instrucciones. La introducción en la tabla de segmentos designa la tabla de páginas que se ha de utilizar. La parte de índice de páginas de la dirección lógica se utiliza para elegir una introducción o anotación de la tabla de página. Esta anotación en la tabla de páginas contiene los bitios de orden superior de la localización real que corresponde a la dirección lógica que se traduce. El campo de índice de bitios de la dirección lógica se utiliza sin cambiar

20

25

30



para las posiciones de orden inferior de la dirección real.

Cuando se traducen direcciones lógicas a direcciones reales, se debe seguir el procedimiento de búsqueda en las tablas indicado. Como las traducciones de direcciones dinámicas se hacen a base de páginas, es muy probable que un programa exija la misma traducción más de una vez para controlar el sistema de proceso de datos. De este modo, se hace una traducción de dirección lógica a real empleando el proceso de búsqueda en tablas, y la dirección real y la dirección lógica de la que se traduce se mantienen en un directorio de gran velocidad llamado almacenamiento de traducciones lógicas. Una petición ulterior de la misma dirección lógica, en el supuesto que el mismo programa mantenga el control, se puede extraer inmediatamente de almacenamiento de traducciones sin tener acceso a la memoria principal.

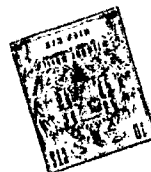
Cada relación de dirección lógica a dirección real (LA/RA), introducida en el directorio de gran velocidad, es válida solamente para el programa particular que mantenía control el sistema de la figura 1 en el instante de la traducción de búsqueda en tabla original. Para evitar tener que eliminar esta información de LA/RA del almacenamiento de traducciones lógicas cuando un nuevo programa toma el control del sistema, se añade como apéndice un identificador o nombre a cada anotación de LA/RA en almacenamiento de traducciones. El nombre sirve para que la anotación quede disponible solamente al programa que causó su introducción y, por lo tanto, no queda disponible para todos los demás programas. Una dirección real válida se puede alcanzar desde el almacenamiento de traducción lógica solamente si el nombre y la dirección asociados con el programa en control son idénticas al nombre y dirección en almacena-



miento de traducción lógica. La correspondencia entre programas y nombres de programas se mantiene en un almacenamiento de identificadores de programa.

5 Cada programa que tiene control del sistema tiene un número de parámetros que definen la forma en que se hacen las traducciones. Estos parámetros son el origen de tablas de segmentos (STO), la longitud de la tabla de segmentos, el tamaño de los segmentos, y el tamaño de las páginas. De una forma selectiva estos parámetros se llaman la información de bases de  
10 segmentos (SB). Cada programa o tarea tendrá su propia tabla de segmentos que comienzan en el origen de la tabla de segmentos.

El almacenamiento de identificadores de programa funciona para almacenar la información de bases de segmentos para  
15 muchos programas diferentes. La información de bases de segmentos de cada programa se asocia con un nombre de 5 bits en cualquier instante hasta 31 nombres que se pueden asignar simultáneamente a 31 programas diferentes. Siempre que un programa trigésimosegundo, que no se encuentra simultáneamente en el  
20 almacenamiento de identificadores de programa, adopta control del sistema, se asigna inmediatamente el trigésimosegundo nombre y se elimina una anotación de programa previamente disponible en el conjunto. Un campo de prioridad en el conjunto 340 determina qué programa se ha de eliminar siempre que el conjunto  
25 de almacenamiento contenga 32 anotaciones de programas proporcionando un nombre sin utilizar, el almacenamiento queda siempre disponible y dispuesto para una asignación inmediata de un nuevo programa lo cual permite que el sistema comience inmediatamente la elaboración o proceso de un nuevo programa  
30 sin tener que esperar hasta que haya disponible un nombre.



Cuando se elimina una introducción de un programa antiguo del almacenamiento de identificadores de programas, todas las anotaciones de que tienen el nombre de este programa antiguo deben eliminarse también de la memoria tampón. Este proceso de invalidez de anotaciones en almacenamiento de traducciones lógicas antiguas asociadas con un solo programa se realiza en último término mientras se elabora la nueva tarea.

La capacidad de poner al corriente el almacenamiento de traducciones lógicas sin estorbar el proceso del programa que mantiene en ese momento el control del sistema se puede realizar porque solamente las anotaciones en almacenamiento de traducciones lógicas tienen el nombre del programa en control quedan disponibles para fines de traducción. Por lo tanto, cualquier anotación de traducción lógica con un nombre diferente, y específicamente aquéllos que tienen el nombre del programa que se elimina no se pueden utilizar.

Cuando el sistema para el proceso de datos de la figura 1 comienza a funcionar, las primeras fases realizadas con la carga o alimentación del programa inicial (IPL). Cuando se ha completado la carga del programa inicial, el programa supervisor hace que se carguen los registradores de control, específicamente el registrador CR-0 344 y el registrador CR-1 345. El registrador CR-0 344 tiene 4 bitios que determinan el tamaño de página y el tamaño de segmentos. El registrador CR-1 345 define la longitud de la tabla de segmentos en bitios 0-7, cuando la longitud se especifica en unidades de 64 bytes.

Los bitios 8-25 del registrador CR-1 345 indican una localización real de 24 bitios (con 6 ceros de orden inferior añadidos como apéndices) que indica el comienzo de la tabla de segmentos para el primer programa. La información de los regis



tradores 344 y 345, cuando es alimentada por el programa supervisor, hace que el sistema comience a elaborar instrucciones del primer programa. Cada dirección lógica en una instrucción se traduce a una dirección real mediante el empleo de tablas de traducción que se sitúan en la memoria principal.

5

Los detalles del funcionamiento del almacenamiento de identificadores de programas se dan en las figuras 7-9. Las fases de traducción descritas más adelante pueden trazarse siguiendo la línea sólida en dichas figuras comenzando en START. El proceso de traducción comienza por la inspección de los registradores de control 344 y 345 (CR CHANGE SIG de la figura 7). La información se introduce por la vía 354 a almacenamiento de identificadores de programas de la figura 4. Como se supone en el ejemplo presente que el primer programa mantiene control y comienza a funcionar (sin expansión de memoria) el almacenamiento de identificadores de programa de la figura 4 no tiene anotaciones. Los 18 bitios de direcciones de la tabla de segmentos se introducen a través de la circuitería de transformación 287 y se delimitan en los 7 bitios de dirección en el registrador 289. El registrador 289 localiza entonces un lugar único en almacenamiento 291. El almacenamiento 291 envía como respuesta información a los registradores 210 y 209, el registrador 292 y los registradores 294, 295, 297 y 298. Como no existe anotación alguna en almacenamiento, el bitio de validez no está y el registrador 209 evita que se satisfaga la puerta 282 con lo que no se envía señal de "no hallado" desde la puerta 282 (ENTRY FND, tómese N).

10

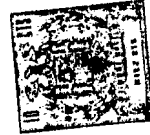
15

20

25

El control 272 detecta el hecho de que no ha salido señal de hallazgo desde la puerta 282 y hace que la dirección en el registrador 289 pase a través del incrementador de  $\pm 1$

30



indicado por la referencia 290 (LOCATION, tómesese 1). La dirección incrementada se introduce a través del selector 282 al registrador 289 (CTA + 1 ----- CTA). El almacenamiento 291 vuelve a localizarse con la dirección incrementada en + 1. Como el ejemplo en cuestión es la primera localización del programa 1, el bitio válido no se encuentra tampoco y de nuevo no se produce una señal de hallazgo (ENTRY FND, tómesese N).

El control 272 al no detectar tampoco una señal de hallazgo de la puerta 282, hace que la información de bases de segmentos en la vía 354 se alimente al lugar localizado por el registrador 289 y hace que se establezca el bitio válido. Al mismo tiempo, el registrador CTB 207 se activa al campo de nombre para asignar el primero de los 31 números y el campo de prioridad se activa a todos los ceros para asignar la mayor prioridad al primer programa. De esta manera, el almacenamiento de identificadores de programas de la figura 4 queda cargado con la información de bases de segmentos para el primer programa, mientras que se asignan un nombre y una pluralidad a dicho programa.

Después se vuelve a tener acceso al almacenamiento de identificadores de programa alimentando el nombre al registrador IA 294 y al registrador IAA 203, con lo que queda disponible cuando se elige a través de las puertas 206 al registrador de direcciones tampón en la figura 3.

En la figura 3, la dirección en la vía 362 procedente del registrador de direcciones efectivas 322 en la unidad de instrucciones de la figura 6 se alimenta en el registrador de direcciones tampón junto con el nombre por la vía 392 procedente de almacenamiento de identificadores de programa 340. El almacenamiento de traducciones lógicas 255 se localiza enton-



ces para determinar si una dirección lógica con el mismo nombre que el conseguido por el almacenamiento de identificadores de programa se encuentra en el almacenamiento 255. La dirección lógica y el nombre procedentes del registrador de direcciones tampón 363 se ponen en el registrador de direcciones lógicas 379 y desde este punto se comparan con las corrientes de salida procedentes de la parte de índice de traducciones lógicas del almacenamiento 255 en los comparadores 328 y 329. Como el ejemplo en cuestión es el primer acceso, no se halla anotación alguna en almacenamiento de traducciones lógicas 255. Para obtener la información deseada, se debe llegar a la tabla de segmentos en la memoria principal.

La dirección real del origen de la tabla de segmentos se introduce por la vía CPU SBR 354 desde la unidad de instrucciones al registrador de traducciones 387 a través de las puertas de selección 333. Asimismo, el número de segmentos procedentes del registrador de direcciones tampón 363 se introduce en el registrador de número de segmento 261. El número de segmento consiste en los bitios de orden superior de la dirección lógica y puede comprender los bitios 8-13 dependiendo del tamaño del segmento. Después el contenido del registrador de traducciones 387 es elegido por las puertas 380 enviándose a la sumadora de líneas 360 y el número de registrador de segmentos procedentes del registrador 261 se elige también a los sumadores de líneas 360 y se añade formando un resultado que se almacena en el registrador de direcciones tampón 363. El contenido del registrador 363 es ahora la dirección de anotación en la tabla de segmento que se utiliza a través del registrador 379, registrador 359 y registrador de dirección de la memoria principal 364 para localizar la memoria principal por la vía



809. La información conseguida de este modo de la memoria principal es la anotación en la tabla de segmentos que contiene la dirección del origen de la tabla de páginas. La dirección del origen de la tabla de páginas se introduce a través del registrador de datos de la memoria principal 384 al registrador de datos tampón de gran velocidad 385 a través de puertas de selección 369 al registrador de traducciones 387. La función ahora consiste en tener acceso a la tabla de páginas en la memoria principal.

La dirección de origen de la tabla de páginas se encuentra en el registrador de traducciones 387 y el número de páginas se encuentra en el registrador de la tabla de páginas 362. El número de la tabla de página consiste en los bitios de orden inferior de la dirección lógica y puede comprender los bitios 12-20 dependiendo del tamaño del segmento y del tamaño de la página. Si el número de segmentos comprende los bitios 8-14 entonces el número de página comprendería el resto de los bitios de la localización, o sea, los bitios 13-20. El número de páginas se elige del registrador 262 y su entrada a la sumadora de líneas 360 y el origen de la tabla de páginas del registrador de traducciones 387 se introducen a través del circuito de selección 380 en la sumadora de líneas 360.

El origen y número de página se suman y el resultado se pone en el registrador de dirección tampón 363 para formar la dirección real de la anotación de tabla de página. El contenido del registrador de dirección tampón 363 se introduce a través del registrador de dirección lógica 379, el registrador de direcciones reales 359 y el registrador de direcciones de la memoria principal 364 para localizar la memoria principal en la vía 809. El resultado del acceso de la anotación en la



5 tabla de páginas, aparece en el registrador de datos de la memoria principal 384 cuando se transfiere al registrador de datos de la memoria tampón de gran velocidad 385 y se elige mediante el circuito 369 y se introduce en el registrador de traducciones 387.

La información queda ahora disponible para alimentarse a almacenamiento de traducciones lógicas de la figura 3.

10 La dirección lógica se elige del registrador 374 por medio del circuito 380 introduciéndose en la sumadora 360 donde nada se suma a la misma y pasa a través del registrador de direcciones tampón 363. La dirección lógica y el nombre en el registrador 363 localiza el almacenamiento de traducciones lógicas 355. La anotación de página almacenada en el registrador de traducciones 387 constituye los bitios 8-20 de la dirección  
15 real del dato y dicha dirección real se introduce a través del acceso de entrada de datos (DI) de la parte de dato del almacenamiento de traducciones lógicas 255.

20 Los bitios 8-13 de dirección lógica y los cinco bitios de nombre procedentes del registrador de direcciones tampón 363 se introducen a través del acceso de entrada de dato (DI) de la parte de índice del almacenamiento de traducciones lógicas 255.

25 El almacenamiento de traducciones lógicas 255 vuelve a localizarse ahora con la información en el registrador tampón 363. Como se ha cargado ahora el almacenamiento 255, los bitios de la dirección real de orden superior se introducen en el registrador de direcciones reales desde la parte de dato del almacenamiento de direcciones reales desde la parte de dato del almacenamiento 255. La memoria tampón de datos 355 se  
30 localiza también empleando información procedente del registra



5      dor de direcciones tampón 363. El contenido de la parte de almacenamiento 355 se lee desde los lugares localizados y es elegido por los circuitos de selección 396 y 397 para proporcionar una entrada a los comparadores 356 y 357. En los comparadores 336 y 337 se hace una comparación con el contenido del registrador de direcciones reales 359. Como en el ejemplo en cuestión es el primer acceso de la memoria tampón de datos 355, no se produce comparación alguna.

10      Por consiguiente, la dirección real procedente del registrador 359 se introduce en el registrador de dirección de la memoria principal 364 de donde sale a la unidad de almacenamiento principal por la vía de dirección 809. La dirección en la vía 809 hace que se llegue a la memoria principal y que el dato localizado se enclave en el registrador de datos de la memoria principal 384 por la vía de datos 811. La información  
15      procedente del registrador 384 se transfiere al registrador de datos de la memoria tampón de gran velocidad 385 donde es elegida por el circuito 369 y almacenado en el registrador de palabras de introducción 388. Al mismo tiempo, el dato se introduce en la parte de dato de la memoria tampón de datos 355 a  
20      través del acceso de entrada de datos (DI). Asimismo, los bits de dirección real del registrador 359 se introducen en la parte de índice del almacenamiento tampón de datos 355 a través del acceso de entrada de datos (DI). La palabra de instrucción en el registrador 388 era la información deseada originalmente y localizada por la dirección lógica y desde el registrador 388 se transmite a la unidad de instrucciones.

25      Si el acceso siguiente a almacenamiento comprende al mismo programa, se saca entonces el mismo nombre del almacenamiento identificador de programa 340 al registrador de dirección  
30



5 ciones tampón 363. Si la dirección lógica es también la misma, entonces se halla la información en el almacenamiento de traducciones lógicas 255 y en la memoria tampón de datos 355, por lo que la información deseada se almacena inmediatamente en uno de los registradores de salida 387 a 391 sin necesidad de acceso a la memoria principal. Si la localización lógica es diferente pero el programa es el mismo, entonces se llega de nuevo a la memoria principal para cargar los almacenamientos 255 y 355 según se ha indicado anteriormente.

10 Cuando un nuevo programa obtiene control del sistema de la figura 1, los registradores de control en la unidad de instrucciones de la figura 6 se alimentan con la nueva información de base de segmentos. La información de base de segmentos se introduce en el almacenamiento de identificadores de programas de la figura 4 donde se localiza un primer lugar y no se halla comparación.

15 La dirección incrementa en uno y de nuevo no se halla dirección. Un nuevo nombre se asigna por parte del registrador CTB 207 puesto que en el ejemplo presente se supone que es el segundo programa que tiene control del sistema. La prioridad se pone al corriente y el bitio válido se establece y se toma lectura de la información del almacenamiento de identificadores de programa de la figura 4. Colocando un nombre en el registrador de direcciones tampón 363 de la figura 3.

20 El funcionamiento de la memoria tampón de la figura 3 continúa después en la forma descrita anteriormente. Siempre que el número de programas con control alcanza el número 31 y entra en control un nuevo programa, se asigna inmediatamente un trigésimosegundo nombre al nuevo programa. No obstante, uno de los nombres anteriores en el almacenamiento de identificado

30



res de programas se elimina inmediatamente. La anotación eliminada se mantiene en el registrador IAB 204 del almacenamiento de identificadores de programa 272. Los 236 lugares en almacenamiento de traducciones lógicas 255 se localizan entonces de  
5 dos en dos para una comparación con el campo de nombre de 5 bits que se saca del registrador 204 a través del registrador de direcciones tampón 363 al registrador de direcciones lógicas 379. Solamente se compara el campo de nombre y por cada lugar que tiene una comparación se elimina la información en la  
10 memoria. No obstante, la eliminación de información no estorba el proceso de una petición de almacenamiento del nuevo trigésimo segundo programa, puesto que el nuevo programa tiene asignado su propio nombre que es diferente al que se purga.

La purga de información del almacenamiento de traducciones lógicas se realiza en último término sin degradar, en  
15 general, el funcionamiento del sistema de proceso de datos.

En la figura 2, se describe una modalidad de un almacenamiento de identificadores de programa donde el almacenamiento comprende una parte de índice y una parte de datos.

Además, la parte de índice y la parte de datos están  
20 cada una comprendidas por una sección primaria y una sección alterna. El almacenamiento de identificadores del programa en la figura 2 funciona para localizar un almacenamiento de traducciones lógicas. En la figura 3 se ha descrito otra modalidad de almacenamiento de identificadores de programa desde el  
25 almacenamiento se localiza de una forma redundante por una traducción de tabla de copia de la información de base de segmentos. El almacenamiento de identificadores de programas en la figura 3 funciona también para localizar el almacenamiento de  
30 traducciones lógicas. Tanto en la modalidad de la figura 2 co-



mo en la modalidad de la figura 3, el almacenamiento de traducciones lógicas localiza, a su vez, una memoria tampón de datos. A pesar de que se han descrito dos modalidades de un almacenamiento de identificadores de programas, se pueden emplear otras modalidades donde el almacenamiento de identificadores de programa localice el almacenamiento de traducciones lógicas que, a su vez, localiza la memoria tampón de datos.

Con relación a la dirección del almacenamiento de traducción lógica 255 en la figura 3 se han utilizado 7 bitios de orden inferior para localizar la parte de dato y la parte de índice de la memoria. La parte de índice de la memoria, a su vez, almacena los 6 bitios de dirección de orden superior a 5 bitios de nombre. Otra modalidad para el almacenamiento de traducciones lógicas es emplear un esquema de dirección de tabla de copia. En dicha otra modalidad, por ejemplo, los 5 bitios de nombre más los 13 bitios de dirección de orden superior (bitios 8-20) se introducen en una tabla de copia y se delimitan en 7 bitios de dirección. Dichos 7 bitios de dirección delimitados se utilizan entonces para localizar tanto la parte de índice como la parte de dato de traducciones lógicas 255. En la variante de la modalidad la parte de índice de la memoria 255 comprende todavía los 6 bitios de localización de orden superior más los 5 bitios de nombre que se sacan, después de la dirección, a los comparadores 328 y 329.

A pesar de que el invento se ha descrito e ilustrado de una forma particular con relación a modalidades de preferencia del mismo, los expertos en la materia comprenderán que pueden efectuarse diversos cambios en la forma y detalles sin desviarse del espíritu y alcance del invento.



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 EE.UU. de A., bajo el número 418.050, de fecha de 21 de noviembre de 1.973, 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 en España, sobre: PERFECCIONAMIENTOS EN SISTEMAS DE PROCESO DE DATOS; caracterizándose por lo siguiente:

1.- Perfeccionamientos en sistemas de procesc de datos del tipo que tienen almacenamiento localizado por direcciones reales y una unidad de elaboración para localizar el almacenamiento con el fin de buscar y almacenar información junto con la ejecución de instrucciones, donde las instrucciones se reciben desde una pluralidad de programas de sistema, donde los programas identifican lugares en el almacenamiento empleando direcciones lógicas y donde cada programa se asocia con una tabla única en almacenamiento para traducir direcciones lógicas a direcciones reales, caracterizados porque el sistema comprende medios de almacenamiento de traducción lógica para almacenar un nombre identificador de programa y direcciones lógicas asociadas y para almacenar direcciones reales correspondientes traducidas; medios de almacenamiento de datos también para almacenar información a la que han tenido acceso dichas direcciones reales; medios de almacenamiento de identificad-





res de programa que tienen una pluralidad de direcciones de nombres cada uno para almacenar un nombre que identifica un programa diferente que tiene información en dichos medios de almacenamiento de traducción lógica, comprendiendo dichos medios de almacenamiento de identificadores de programas medios para mantener por lo menos uno de dichos lugares vacío y disponible para ser utilizado por un nuevo programa que no tenga información en dicho almacenamiento de traducción lógica.

5

2.- Perfeccionamientos según la reivindicación 1, caracterizados porque dichos medios de almacenamiento de identificadores de programa comprenden además un paquete de memoria localizada de una forma redundante que tiene una pluralidad de lugares cada uno de los cuales comprende un campo de bases de segmentos para identificar de una forma única un programa, un campo de nombres para almacenar un nombre asignado al programa identificado en el campo de base de segmento, y un campo de prioridad para establecer una pluralidad de reposición para el programa identificado en el campo de base de segmento; y medios de dirección para localizar los lugares en dicho paquete de memoria; medios de salida de datos para almacenar información que se ha leído de dicho paquete de memoria; medios de entrada de datos para cargar información en dicho paquete de memoria; medios de control de paquete de memoria para controlar dichos medios de localización, dichos medios de entrada de datos y dichos medios de salida de datos.

10

15

20

25

3.- Perfeccionamientos según la reivindicación 2, caracterizados porque dichos medios de localización comprenden una tabla de copia para delimitar un primer campo de bitios de localización en un segundo campo de bitios de localización cuando dicho segundo campo es menor que dicho primer campo.

30





4.- Perfeccionamientos según la reivindicación 1, ca  
racterizados porque dichos medios de almacenamiento de identi-  
ficadores de programas, dichos medios de almacenamiento de tra-  
ducción lógica y dichos medios de almacenamiento de datos tam-  
pón comprenden cada uno una parte de índice y una parte de da-  
to asociada donde la parte de índice y la parte de dato en ca-  
da almacenamiento o memoria se localizan por los mismos bitios  
de dirección o localización.

5.- Perfeccionamientos según la reivindicación 4, ca  
racterizados porque cada una de dichas partes de índice y cada  
una de dichas partes de dato comprenden una sección primaria y  
una sección alterna y porque dicho sistema comprende además me-  
dios de localización por cada una de dichas memorias para loca-  
lizar dichas memorias, comprendiendo dichos medios de localiza-  
ción por cada almacenamiento o memoria medios comparadores pri-  
mario y alterno para recibir el contenido de un lugar localiza-  
do en dichas secciones primaria y alterna de la parte de índi-  
ce para comparar con una dirección de entrada y medios para  
elegir la sección primaria o alterna de la parte de dato en  
función a una comparación en los medios comparadores primario  
o alterno, respectivamente, comprendiendo dicho sistema medios  
para conectar la salida elegida desde la parte de dato del al-  
macenamiento o memoria identificadora de programa a los medios  
comparadores para el almacenamiento de traducción lógica y me-  
dios para conectar la salida elegida de la parte de dato del  
almacenamiento o memoria o traducción lógica a los medios com-  
paradores para el almacenamiento o memoria de datos tampón.

6.- Perfeccionamientos según la reivindicación 2, ca  
racterizados porque dicho paquete de memoria comprende además  
un campo de validez en cada lugar para identificar anotaciones



válidas dentro de dicho paquete de memoria.

5 7.- Perfeccionamientos según la reivindicación 2, ca-  
racterizados porque dicho paquete de memoria comprende un pri-  
mer número de lugares y porque dicho campo de nombres compren-  
de un número de bits que especifican de una forma única un  
segundo número de nombres, donde dicho segundo número es menor  
que dicho primer número, por lo que el número de anotaciones  
válidas en dicho paquete de memoria es menor que el número de  
lugares en dicho paquete de memoria.

10 8.- Perfeccionamientos según la reivindicación 7, ca-  
racterizados porque dicho paquete de memoria comprende lugares  
y el tamaño del campo de prioridad es igual al tamaño del cam-  
po de nombres.

15 9.- Perfeccionamientos según la reivindicación 6, ca-  
racterizados porque dichos medios de salida de datos compren-  
den por cada uno de dichos campos un registrador de marcha li-  
bre para almacenar la salida de dicho paquete de memoria cada  
vez que dicho paquete de memoria es localizado y comprende un  
registrador controlado para almacenar salidas elegidas de di-  
cho paquete de memoria bajo control de dichos medios de con-  
20 trol.

25 10.- Perfeccionamientos según la reivindicación 1, ca-  
racterizados porque dichos medios de almacenamiento de identi-  
ficadores de programa comprenden además un paquete de memoria  
localizada de una forma redundante que tiene una pluralidad de  
lugares, cada uno de los cuales comprende un campo de bases  
de segmentos para identificar de una forma única un programa,  
incluyendo un campo de nombres para almacenar un nombre asigna-  
do al programa identificado en el campo de base de segmentos,  
30 incluyendo un campo de prioridad para establecer una prioridad





de reemplazamiento de reposición para el programa identificado en el campo de bases de segmentos, e incluyendo un campo de validez para identificar si dicho campo de nombres es válido, medios de dirección para localizar los lugares en dicho paquete de memoria con información de base de segmento asociada con programas del sistema; medios de salida de datos para almacenar información a la que se ha llegado desde lugares localizados en dicho paquete de memoria; medios de entrada de datos para cargar información en lugares localizados en dicho paquete de memoria; medios de control del paquete de memoria para controlar dichos medios de localización, dichos medios de entrada de datos y dichos medios de salida de datos para tener acceso a la información de validez, de base de segmento, de nombre y de prioridad.

11.- Perfeccionamientos según la reivindicación 10, caracterizados porque dichos medios de control del paquete de memoria comprenden además medios para asignar nombres, hasta un número máximo de nombres, para introducción en dichos campos de nombre y comprende además medios para invalidar un nombre previamente utilizado, cuando dicho número máximo de nombres se ha asignado, con el fin de dejar disponible dicho nombre previamente utilizado para asociación con un nuevo programa.

12.- Perfeccionamientos según la reivindicación 11, caracterizados porque dichos medios de salida de datos comprenden medios para almacenar dicho nombre previamente utilizado, por lo que dicho nombre previamente utilizado queda disponible para localizar dicho dispositivo de almacenamiento de traducción lógica y para purgar dicho nombre previamente utilizado de dicho dispositivo de almacenamiento de traducción lógica.



13.- Perfeccionamientos según la reivindicación 10, caracterizados porque dichos medios de control del paquete de memoria comprenden además medios para establecer la validez de un número menor que dicho número máximo de dichos campos válidos, por lo que por lo menos uno de dichos campos válidos no quedan establecidos como válido y de este modo el campo de nombre asociado queda disponible para recibir un nombre asociado con una nueva base de segmento y un nuevo programa asociado.

14.- Perfeccionamientos según las reivindicaciones anteriores, caracterizados porque en un sistema de proceso de datos que tiene almacenamiento localizado por direcciones o localizaciones reales y una unidad de elaboración para localizar almacenamiento con el fin de buscar y almacenar información junto con la ejecución de instrucciones, donde las instrucciones se reciben de una pluralidad de programas del sistema, donde los programas identifican lugares en almacenamiento utilizando localizaciones lógicas y donde cada programa se asocia con una tabla única en almacenamiento para traducir localizaciones lógicas a localizaciones reales se incluye un aparato que comprende medios de almacenamiento de traducción lógica para almacenar un nombre identificador de un programa y localizaciones lógicas asociadas y para almacenar localizaciones reales traducidas correspondientes; un dispositivo de almacenamiento de datos tampón para almacenar información a la que se ha tenido acceso desde dichas localizaciones reales; medios de almacenamiento de identificadores de programa que tienen una pluralidad de lugares de nombres cada uno para almacenar un nombre que identifica un programa diferente que tiene información en dicho dispositivo de almacenamiento de traducción lógica, comprendiendo dicho dispositivo de almacenamiento de iden-





5      tificadores de programa medios para mantener por lo menos uno  
de dichos lugares vacío y disponible para ser utilizado por un  
nuevo programa que no tenga información en dicho almacenamien-  
to de traducción lógica; medios para localizar dichos medios  
10      de almacenamiento de traducción lógica con el fin de obtener  
una de dichas localizaciones reales, con una localización lógi-  
ca prevista por un programa del sistema particular y con un  
nombre procedente de dicho almacenamiento de identificadores  
de programas asociado con dicho programa de sistema del siste-  
15      ma particular; y medios para localizar dicho dispositivo de al-  
macenamiento de datos de tampón con una de dichas localizacio-  
nes reales con el fin de tener acceso a información asociada  
con dicha localización de las citadas localizaciones reales.

15      15.- Perfeccionamientos según la reivindicación 14,  
caracterizados porque dicho dispositivo de control del paquete  
de memoria comprende además medios para asignar nombre, hasta  
un número máximo de nombres, para introducirse en dichos cam-  
pos de nombres y comprende además medios para invalidar un nom-  
bre previamente utilizado, cuando dicho número máximo de nom-  
20      bres se ha asignado con el fin de que dicho nombre previamente  
utilizado quede disponible para asociación con un nuevo progra-  
ma.

25      16.- Perfeccionamientos según la reivindicación 15,  
caracterizados porque dichos medios de salida de datos compren-  
de medios para almacenar dicho nombre previamente utilizado,  
por lo que dicho nombre previamente utilizado queda disponible  
a dicho medio de localización para localizar dicho dispositivo  
de almacenamiento de traducción lógica, y porque dichos medios  
empleados para localizar dicho dispositivo y de almacenamiento  
30      de traducción lógica comprenden medios para localizar en se-





cuencia todos los lugares de almacenamiento de traducción lógica, comprende medios para detectar dicho nombre previamente utilizado en cualquier lugar localizado, y comprende medios para purgar dicho nombre previamente utilizado cuando es detectado en cualquier lugar localizado.

5

17.- Perfeccionamientos según las reivindicaciones anteriores, caracterizados porque en un sistema para el proceso de datos que tiene almacenamiento de sistema localizado por direcciones reales y que tiene una unidad de elaboración para localizar el almacenamiento de sistema con el fin de buscar y almacenar información junto con la ejecución de instrucciones y llevar a cabo traducciones de localizaciones dinámicas cuando las instrucciones se reciben desde una pluralidad de programas del sistema, cuando cada programa identifica lugares en almacenamiento del sistema utilizados localizaciones lógicas que no son únicas a un programa, y cuando cada programa se asocia con una tabla única en almacenamiento para traducir dinámicamente direcciones lógicas que no son únicas a direcciones reales únicas, se incluye un aparato que comprende un dispositivo de almacenamiento de traducción lógica que tiene una parte de índice con una pluralidad de lugares para almacenar nombres de identificadores de programas y para almacenar nombres de identificadores de programas y para almacenar localizaciones lógicas asociadas y que tiene una parte de dato que tiene una pluralidad de lugares para almacenar localizaciones reales correspondientes obtenidas traduciendo localizaciones lógicas a localizaciones reales empleando una tabla única; medios de almacenamiento de datos tampón que tiene una parte de índice para almacenar localizaciones reales y que tiene una parte de dato para almacenar información a la que se ha tenido acceso de alma-

10

15

20

25

30





5 cenamiento o memoria del sistema de dichas localizaciones rea-  
les; medios de almacenamiento de identificadores de programas  
que comprenden un paquete de memoria localizada de una forma  
redundante que tiene una pluralidad de lugares localizables ca  
10 da uno de los cuales comprende un campo de base de segmentos  
para identificar de una forma única un programa incluyendo un  
campo de nombre para almacenar un nombre asignado al programa  
identificado en el campo de bases de segmentos, incluyendo un  
campo de prioridad para establecer una prioridad de reposición  
15 para el programa identificado en el campo de bases de segmen-  
tos, e incluyendo un campo de validez para identificar si la  
anotación del campo de nombre es válida, comprendiendo dichos  
medios de almacenamiento de identificadores de programas me-  
dios de salida de datos para almacenar información a la que se  
20 ha tenido acceso de los lugares localizados en dicho paquete  
de memoria, comprendiendo dichos medios de almacenamiento de  
identificadores de programas medios de control del paquete de  
memoria para controlar dichos medios de dirección, dichos me-  
dios de entrada de datos y dichos medios de salida de datos pa  
25 ra tener acceso a información de validez, bases de segmentos,  
nombres y prioridad comprendiendo dichos medios de control del  
paquete de memoria medios para asignar nombres, hasta un núme-  
ro máximo de nombres, para introducción en dichos campos de  
nombres e incluyendo medios para invalidar un nombre previamen  
30 te utilizado, cuando se ha asignado dicho número máximo de nom-  
bres y para almacenar dicho nombre previamente previamente uti-  
lizado en dichos medios de salida de datos, medios para locali-  
zar dichos medios de almacenamiento de identificación de pro-  
gramas con información de base de segmento asociada con un pro-  
grama particular actualmente en control de dichos sistemas de



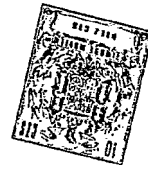


proceso de datos para obtener un nombre que identifique dicho programa particular; medios para direccionar dichos medios de almacenamiento de traducción lógica y obtener una de dichas direcciones reales empleando una localización lógica desde dicho programa particular y empleando un nombre de dichos medios de almacenamiento de identificadores de programa que identifica dichos programas particulares, comprendiendo también dichos medios de localización del dispositivo de almacenamiento de traducción lógica también medios para localizar en secuencia todos los lugares de almacenamiento de traducción lógica cuando no han sido localizados por una dirección lógica con el fin de detectar y purgar dicho nombre previamente utilizado del dispositivo de almacenamiento de traducción lógica; medios para localizar dicho dispositivo de almacenamiento de datos también con una de dichas localizaciones reales con el fin de tener acceso a la información asociada con dicha localización real de las citadas localizaciones reales.

18.- Perfeccionamientos según las reivindicaciones anteriores, caracterizados porque en un sistema para el proceso de datos que tiene almacenamiento direccionado por direcciones reales y una unidad de elaboración para direccionar almacenamiento con el fin de buscar y almacenar información junto con la ejecución de instrucciones, cuando las instrucciones se reciben desde una pluralidad de programas del sistema, y donde los programas identifican lugares en almacenamiento utilizando direcciones lógicas y donde cada programa se asocia con una tabla única en almacenamiento para traducir localizaciones lógicas a localizaciones reales, se incluye un aparato que incluye un dispositivo de almacenamiento de traducción lógica que tiene una parte de índice dividida en secciones primaria y alter-



na para almacenar nombres identificadores de programas y localizaciones lógicas asociadas y que tiene una parte de datos dividida en secciones primaria y alterna para almacenar direcciones reales traducidas correspondientes, medios de almacenamiento de datos tampón que tienen una parte de índice dividida en secciones elemental y eventual para almacenar direcciones reales y que tiene una parte de dato dividida en secciones primaria y alterna para almacenar información a la que se ha tenido acceso de dichas direcciones reales; medios de almacenamiento de identificadores de programa que tienen una pluralidad de lugares de nombres cada uno, para almacenar un nombre que identifica un programa diferente, que tiene información en dicho dispositivo de almacenamiento de traducción lógica, comprendiendo dichos medios de almacenamiento identificadores de programas, medios para mantener al menos uno de dichos lugares vacío y disponible para ser utilizado por otro programa que no tenga información en dicho dispositivo de almacenamiento de traducción lógica, medios para localizar simultáneamente dichas secciones primaria y alterna de dicha parte de índice del dispositivo de almacenamiento de traducción lógica con una localización lógica prevista, un programa de sistema particular y con un nombre de dicho almacenamiento de identificadores de programas asociado con dicho programa de sistema particular, para obtener una de dichas localizaciones reales de dicha sección primaria o alterna, respectivamente, de la citada parte de datos de dicho dispositivo de almacenamiento de traducción lógica, medios para localizar simultáneamente dichas secciones primaria y alterna de dicha parte de índice del citado dispositivo de almacenamiento de datos tampón con dicha dirección real de las citadas direcciones reales con el fin de tener ac-



ceso a la información asociada con dicha localización real de las citadas localizaciones reales desde dicha sección primaria o alterna, respectivamente, de la citada parte de datos de dicho almacenamiento de datos tampón.

5                   19.- Perfeccionamientos según las reivindicaciones anteriores, caracterizados porque un sistema para el proceso de datos que tiene almacenamiento direccionado por direcciones reales y una unidad de elaboración para direccionar almacenamiento con el fin de buscar y almacenar información junto  
10 con la ejecución de instrucciones, donde las instrucciones se reciben de una pluralidad de programas del sistema, donde los programas identifican lugares en almacenamiento utilizando localizaciones lógicas y donde cada programa se asocia con una tabla única en almacenamiento para producir direcciones lógicas a direcciones reales, dicho sistema que comprende un dispositivo de almacenamiento de traducción lógica para almacenar un nombre identificador de programa y direcciones lógicas asociadas y para almacenar direcciones reales traducidas correspondientes; medios de almacenamiento de datos tampón para almacenar información a la que se ha tenido acceso de dichas direcciones reales; medios de almacenamiento de identificadores de programas que tienen una pluralidad de lugares de nombres cada uno para almacenar un nombre que identifica un programa diferente, que tiene información en dicho dispositivo de almacenamiento de traducción lógica, comprendiendo dichos medios de almacenamiento de identificadores de programa: un paquete de memoria localizada con una forma redundante que tiene una pluralidad de lugares, cada uno de los cuales comprende un campo de bases de segmentos para identificar de una forma  
20  
25  
30 única un programa, incluyendo un campo de nombre para almace-





nar un nombre asignado al programa identificado en el campo de bases de segmentos, incluyendo un campo de prioridad para establecer una prioridad de desplazamiento, reposición para el programa identificado en el campo de bases de segmentos, e incluyendo un campo de validez para identificar si dicho campo de nombres es válido; medios de direccionamiento para localizar los lugares en dicho paquete de memoria con información de bases para almacenar información a la que se ha tenido acceso desde lugares direccionados en dicho paquete de memoria; medios de entrada de datos para cargar información en lugares direccionados en dicho paquete de memoria; medios de control del paquete de memoria para controlar dichos medios de direccionamiento, dichos medios de entrada de datos y dichos medios de salida de datos, con el fin de tener acceso a información de validez, bases de segmentos, nombre y prioridad, comprendiendo dichos medios de control de paquete de memoria, medios para incrementar la información de bases de segmentos en uno, para hacer que dichos medios de direccionamiento vuelvan a localizar dicho paquete de memoria si dichos medios de direccionamiento no hacen que un nombre válido se fije en dichos medios de salida de datos.

20.- Perfeccionamientos en sistemas de proceso de datos, tal y como queda sustancialmente descrito en la presente memoria e ilustrado en los dibujos adjuntos.

Esta Memoria consta de 51 hojas escritas a máquina por una sóla cara.

Madrid, 28 FEB. 1975  
AMDAHL CORPORATION,  
y FUJITSU LIMITED.

J. ROMEZ AGUDO Y MOJER  
p. Firmado: L. Gueta Fernández



28 FEB. 1975

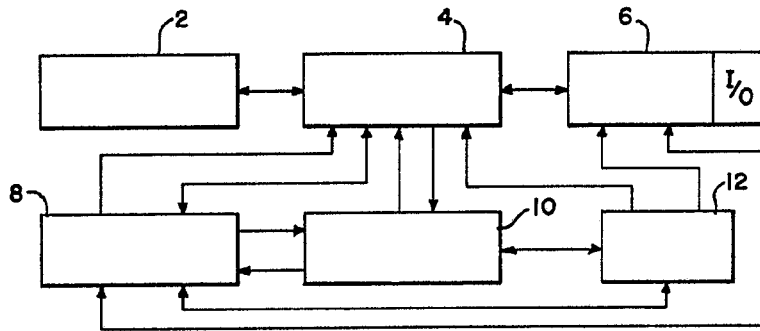


FIG.-1

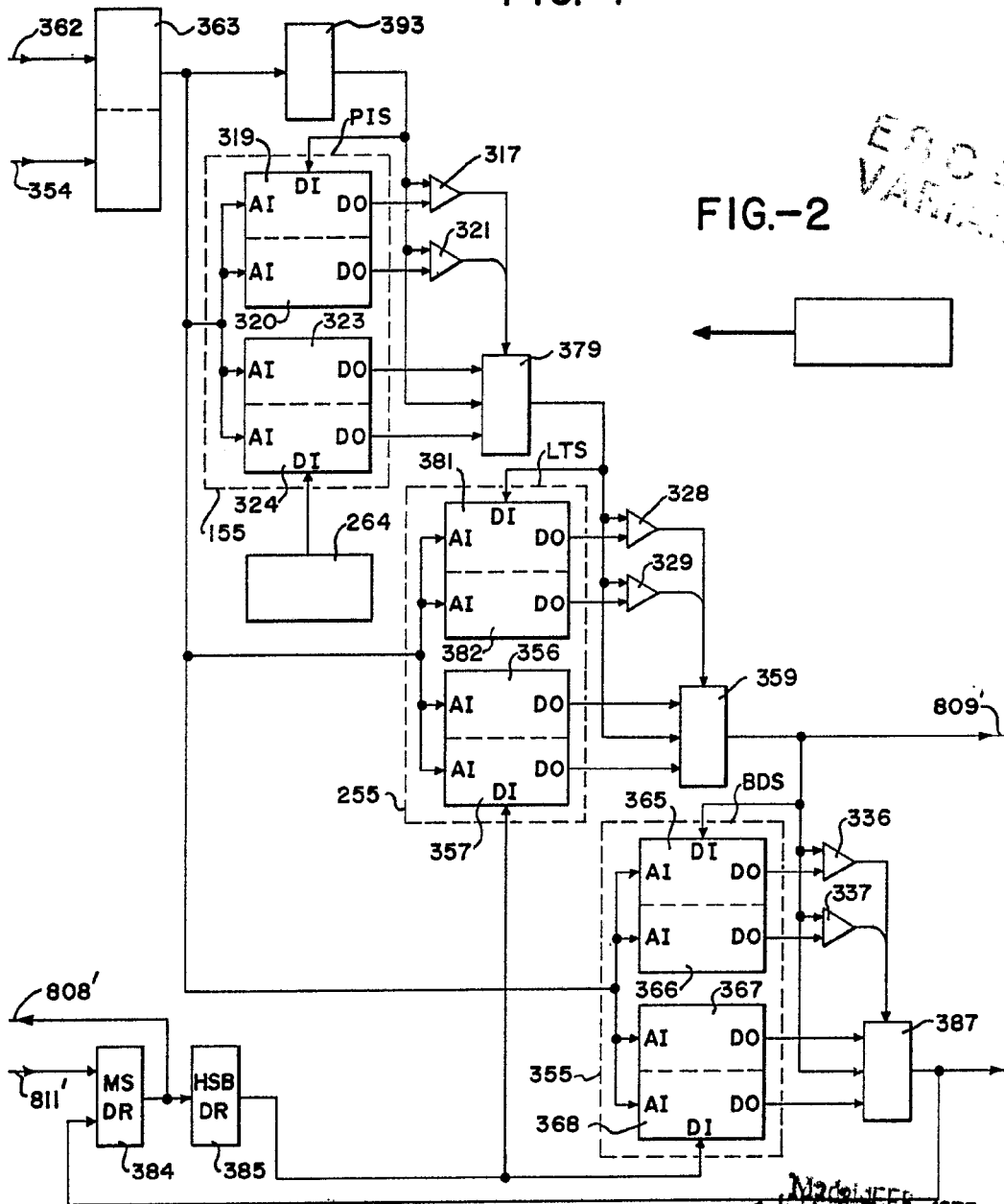


FIG.-2

ESCALA  
VARIANTE

Madrid, Feb. 1975  
A. GONZALEZ, INGENIERO  
p. p. Firmado: L. Garcia Fernandez

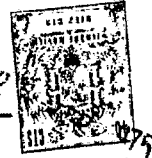
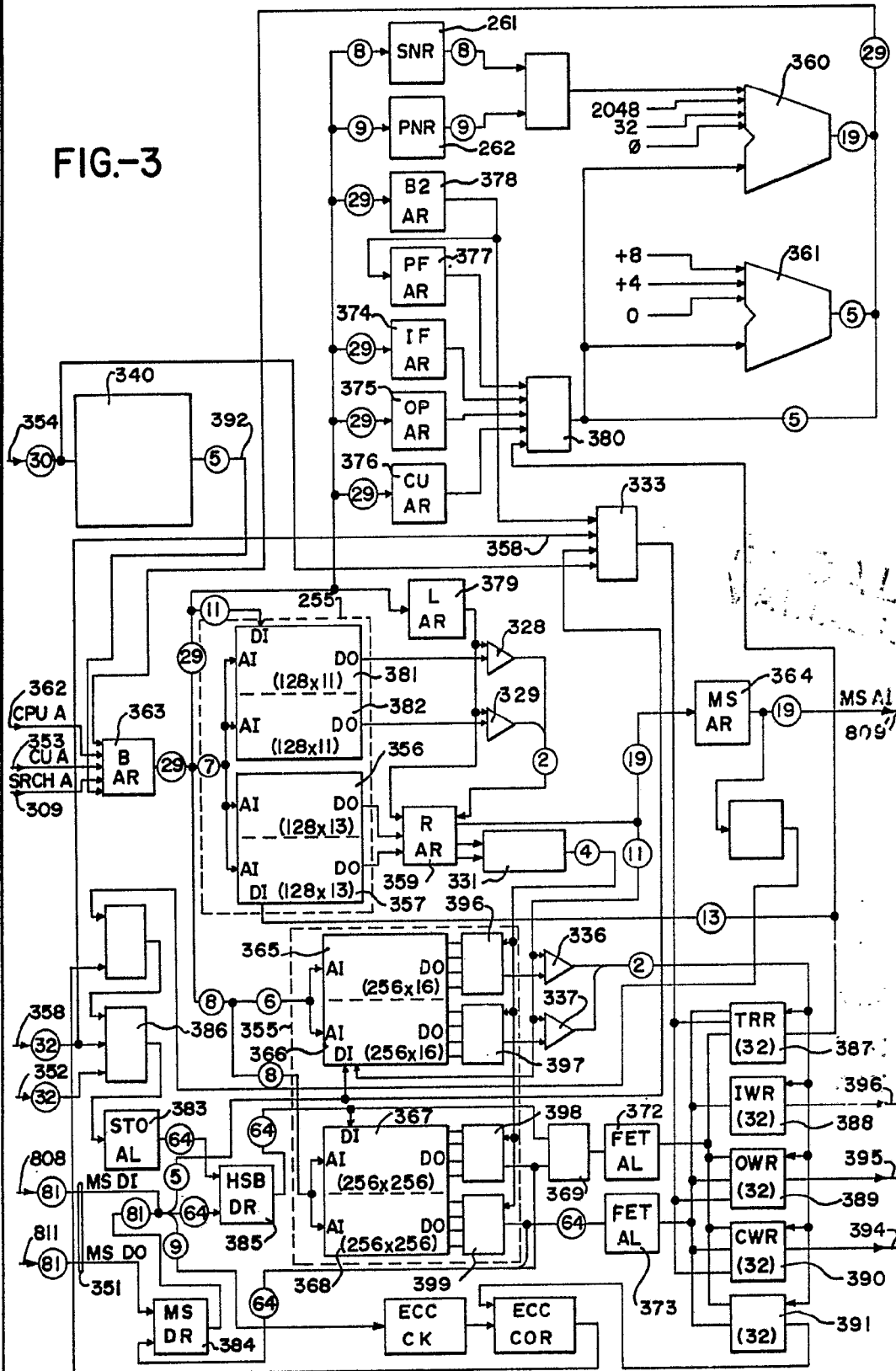


FIG.-3



12-8 FEB. 1975

J. GOMEZ GONZALEZ Y IZQUIERDO  
Ingeniero en Ciencias Físicas

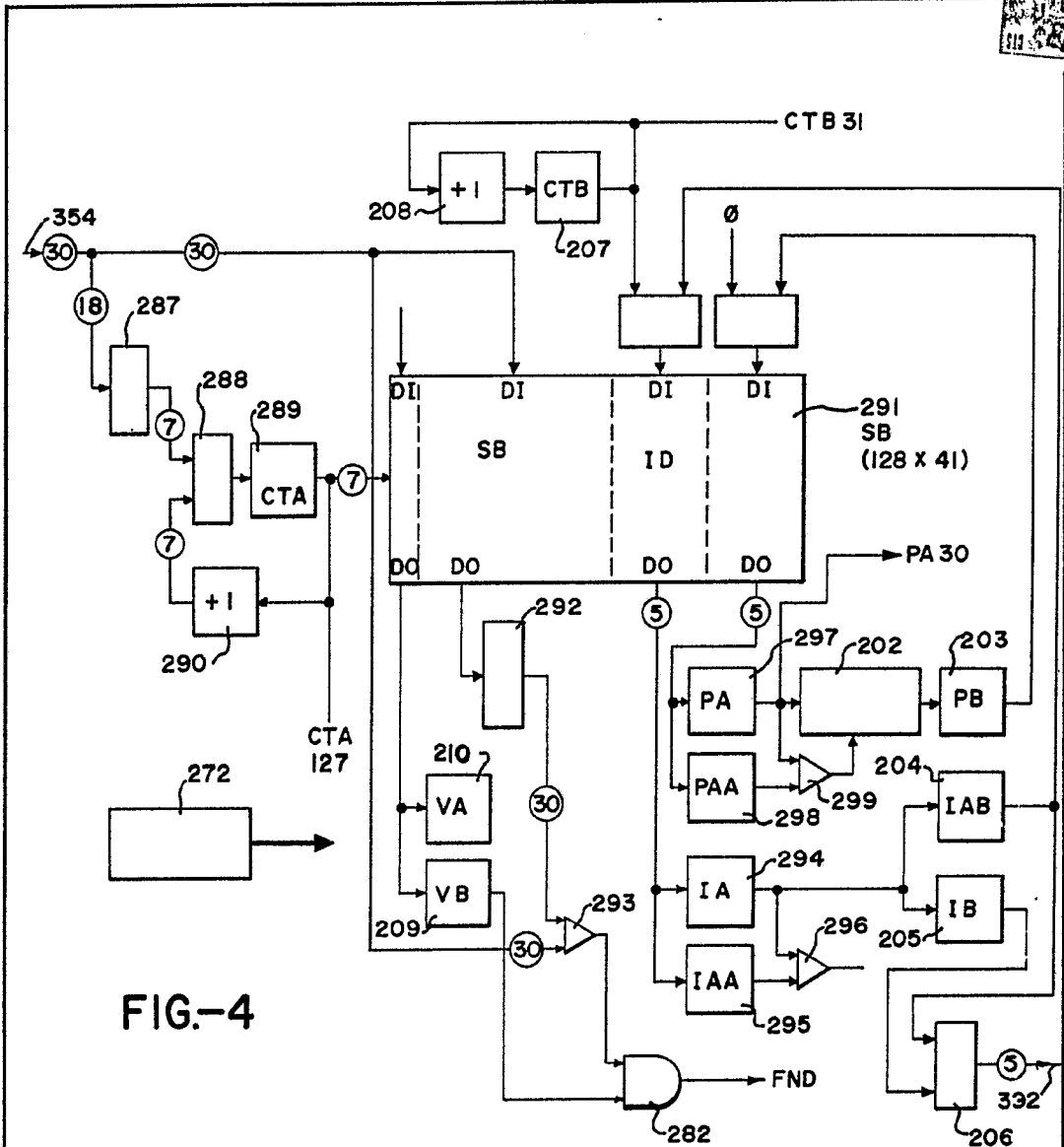


FIG.-4

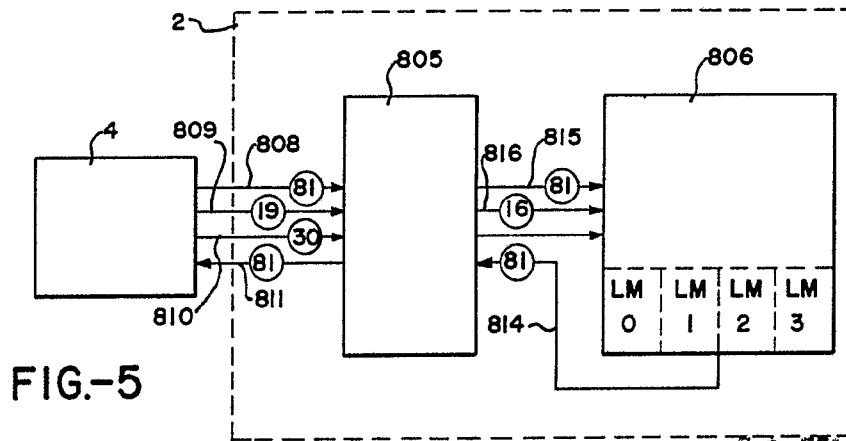
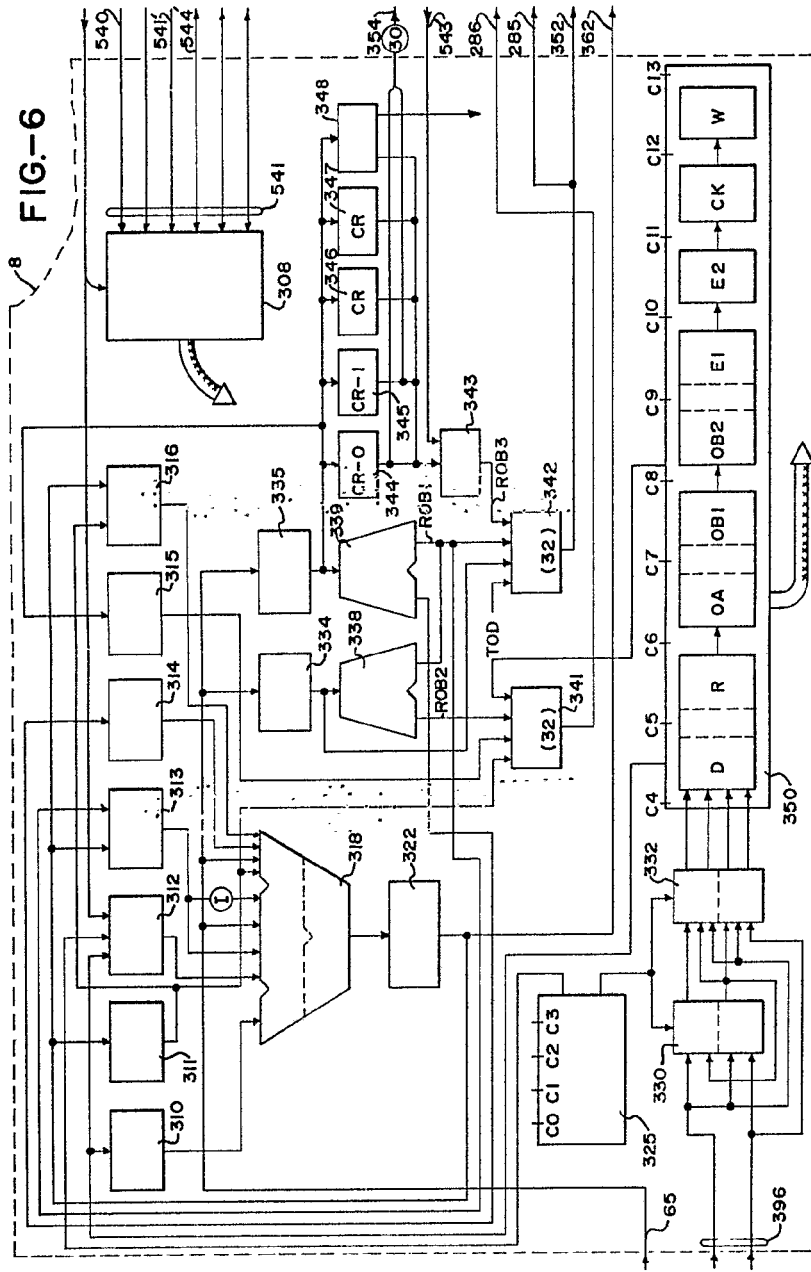


FIG.-5

28 FEB. 1975

*[Handwritten signature]*

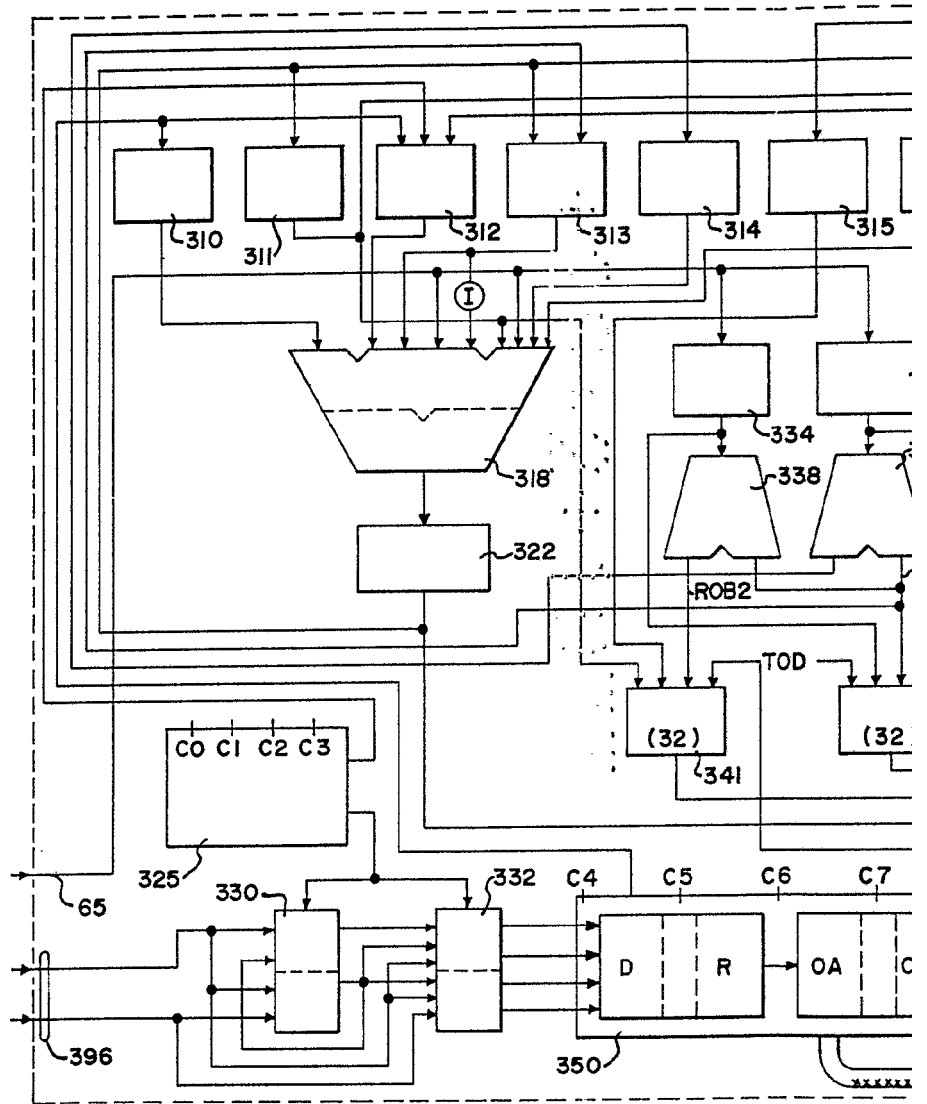
6 HOJAS NO 4



2 FEB. 1974

AMDAHL CORPORATION

FUJITSU LIMITED



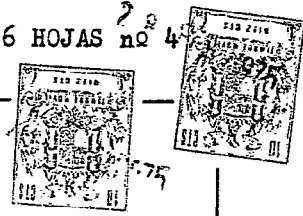
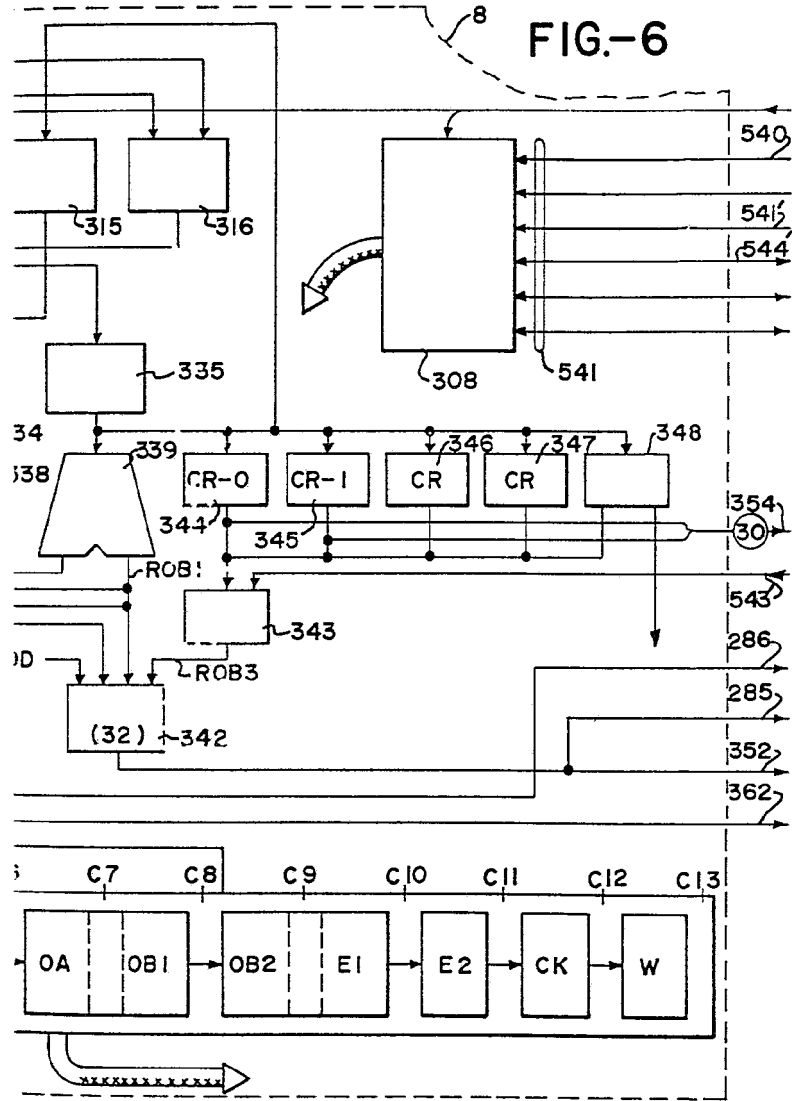


FIG.-6



20 FEB. 1975

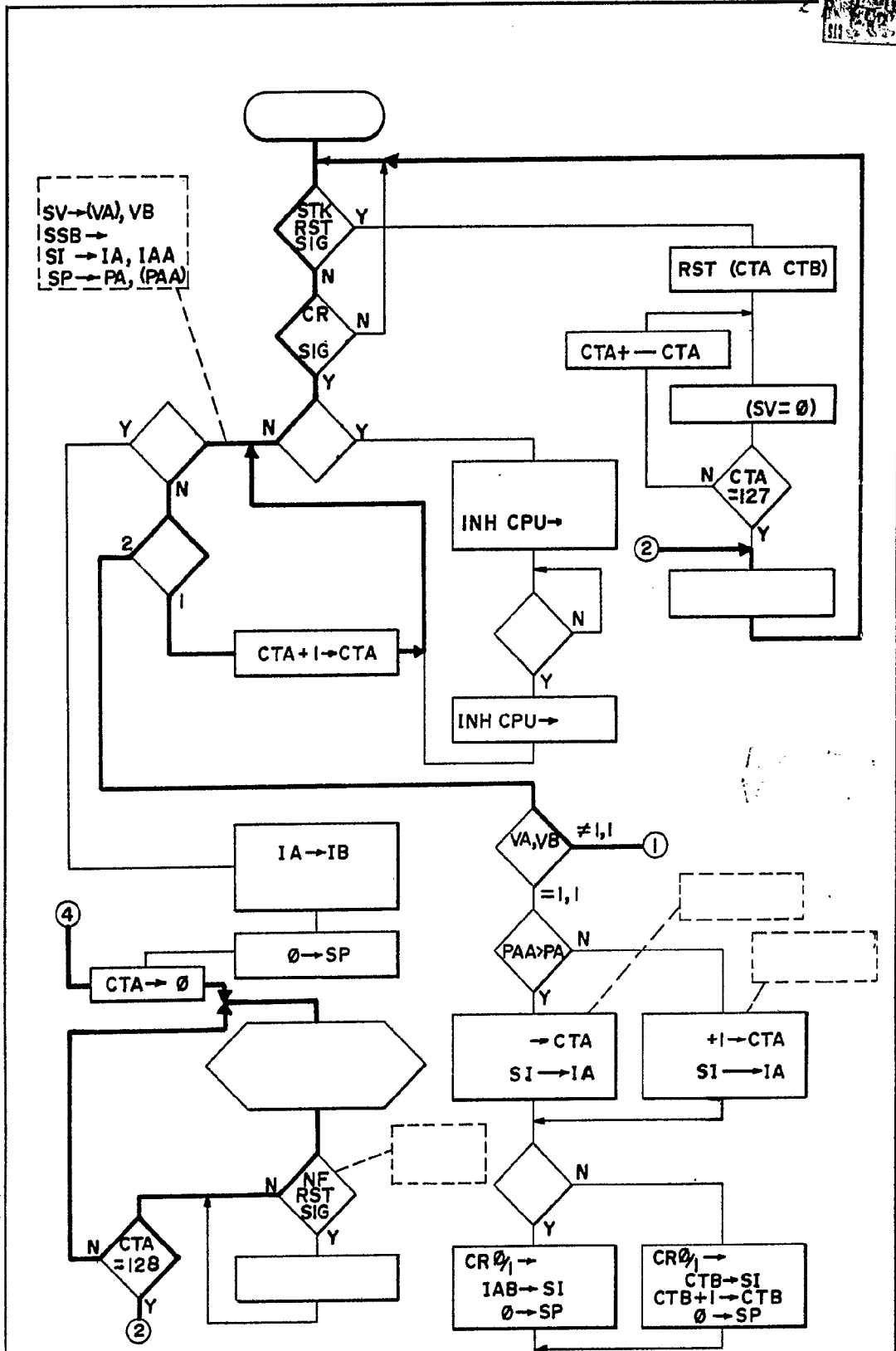


FIG.-7

28 FEB. 1975

*Handwritten signature or initials at the bottom right.*

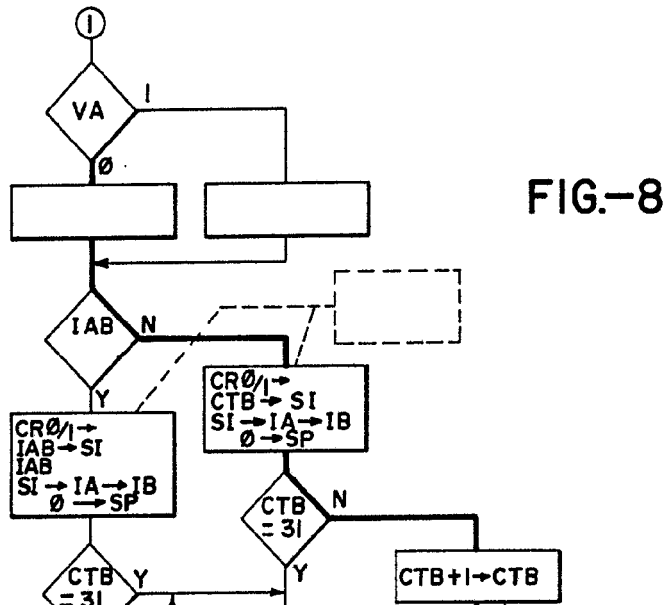


FIG.-8

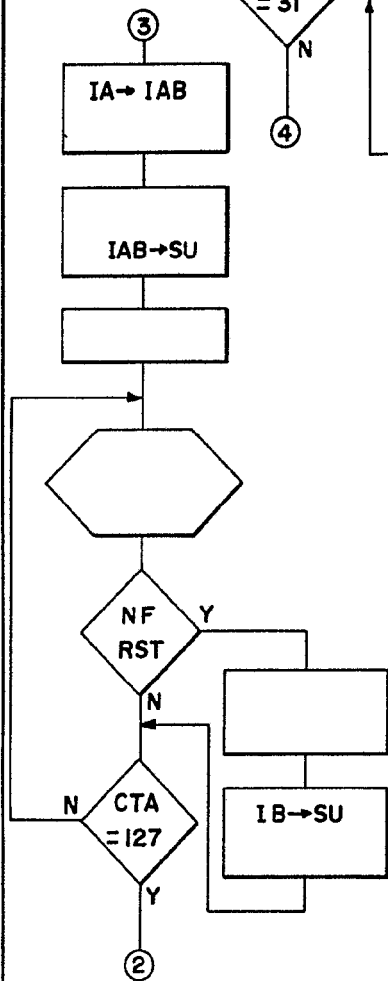


FIG.-9

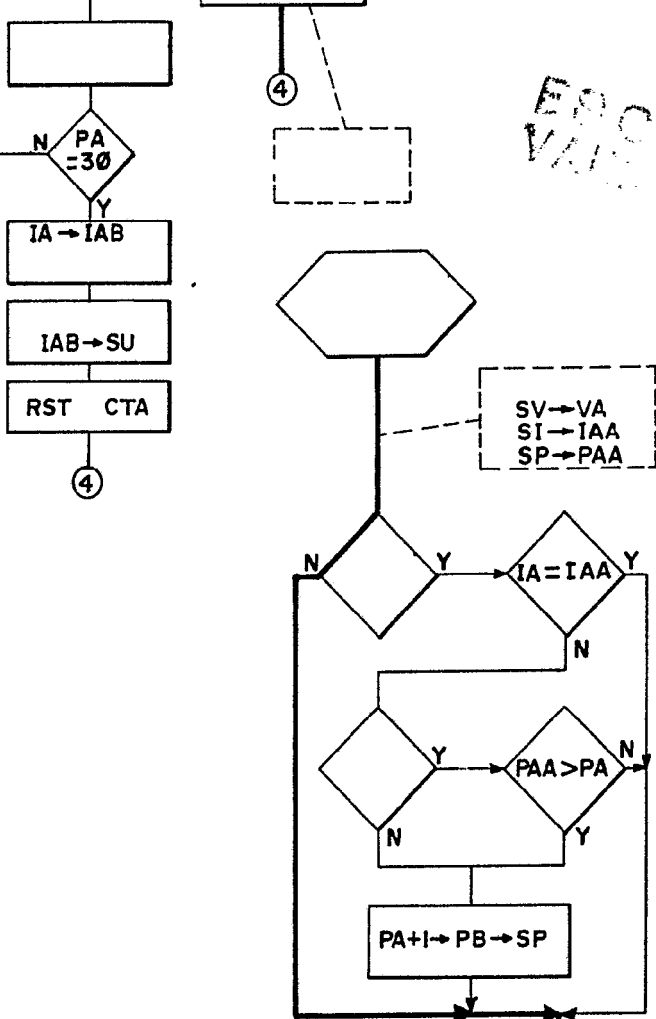


FIG.-10

CTA -> CTA + 1  
28 FEB. 1975

*Handwritten signature*