



ESPAÑA

10 ES	11 21	NUMERO 458.210	10 A1
	22	FECHA DE PRESENTACION 27-4-77	

PATENTE DE INVENCION

30 PRIORIDADES:		
31 NUMERO	32 FECHA	33 PAIS
17121/76	27 de Abril de 1976	Inglaterra
47 FECHA DE PUBLICIDAD	51 CLASIFICACION INTERNACIONAL G 0 6 F	62 PATENTE DE LA QUE ES DIVISIONARIA
64 TITULO DE LA INVENCION Perfeccionamientos en procesadores de datos.		
71 SOLICITANTE (ES) GENERAL INSTRUMENTS MICROELECTRONICS LIMITED, entidad británica.		
DOMICILIO DEL SOLICITANTE residente en Rolls House, 7 Rolls Buildings, Fetter Lane, Londres EC 4A INL, Formerly of Moor House, Londres Wall, Londres ECL., Inglate rra.		
72 INVENTOR (ES) BRIAN ROBINSON KIRK, y BRIAN COLIN HARDEN.		
73 TITULAR (ES)		
74 REPRESENTANTE D. Jose Miguel Gomez-Acebo y Pombo.		

La presente invención se refiere a un procesador de datos para procesar un conjunto de señales de entrada de acuerdo con un conjunto de ecuaciones Boolean para proporcionar un conjunto de señales de salida.

5. Dicho procesador comprende circuitería lógica para realizar una pluralidad de funciones lógicas y se dispone de modo que pueda ser controlado por una memoria de programa en la cual se almacena dicho conjunto de ecuaciones en forma de una pluralidad de etapas de instrucciones que se disponen para localizadas en secuencia.

10. Una ventaja de dicho procesador es que puede ser programao fácilmente por un ingeniero que esté familiarizado con el álgebra de Boolean pero que no esté familiarizado con los lenguajes de programación como tales. No obstante, cuando se tienen que valorar particularmente ecuaciones Boolean agrupadas, los procesadores propuestos anteriormente, que nosotros conocemos, emplean necesariamente un número indebidamente elevado de etapas de instrucciones. Por lo tanto, un objeto del presente invento es proporcionar un procesador de datos de la clase descrita, pero en el cual se pueden reducir al mínimo las etapas de instrucciones, particularmente cuando se tienen que valorar ecuaciones agrupadas.

15. Por lo tanto, según el invento, un procesador de datos de la clase mencionada comprende además una pila para el almacenamiento temporal de datos que, cuando se usa el procesador, la lectura de entrada o de salida se toma de la parte superior de la pila.

20. Por el término "pila" se entiende un registrador de almacenamiento generalmente en forma de registrador de corrimiento pero en el cual la lectura de datos de entrada y de salida

30.

se toma de un extremo común del registrador, cuyo extremo se denominará en la presente memoria como "parte superior" de la pila. De éste modo se obtiene de hecho, una "pila" de valores de datos, que aumenta o disminuye añadiendo o extrayendo datos de la parte superior de la pila.

5.

La pila se puede emplear durante un ciclo de proceso para almacenar temporalmente términos acumulados durante la evaluación de una ecuación, con el fin de facilitar la reducción al mínimo del número de etapas de instrucciones necesarias, particularmente para una ecuación agrupada.

10.

Para poder comprender mejor el invento y la forma que se puede poner en práctica, hacemos referencia, a título de ejemplo, al dibujo adjunto, cuya única figura es un diagrama de conjuntos de una modalidad del invento, que podemos llamar analizador Boolean por secuencias (SBA).

15.

El analizador Boolean por secuencia (SBA) es un procesador de bitios simples muy sencillo que puede evaluar directamente un conjunto de ecuaciones Boolean.

20.

El empleo de ecuaciones Boolean como "lenguaje de programación" tiene un cierto número de ventajas únicas en su género:

(i) es concisos

(ii) está normalizado a escala mundial

(iii) los ingenieros ya lo utilizan y lo entienden

25.

(iv) las universidades lo enseñan y lo han enseñado durante muchos años.

(v) sirve para la doble finalidad de programa y documentación

(vi) ha pasado la prueba del tiempo

30.

Las ecuaciones definen la lógica que controla el siste

ma al que se conectan las entradas y salidas del SBA, según se ilustra más adelante.

5. Además de la lógica Boolean, la mayoría de los sistemas exigen que algunos acontecimientos se tengan que "recordar", siendo esta la razón para el empleo de basculadores en las ejecuciones lógicas de tipo TTL. En el BSA se habilitan para dichas finalidades una pluralidad de elementos internos de almacenamiento.

10. Se utiliza una memoria para contener una versión codificada de las ecuaciones Boolean que definen la función deseada del SBA y existe una correspondencia de uno a uno entre los datos de esta memoria y las ecuaciones Boolean planteadas por un ingeniero.

Principios de Operación

15. En la única figura de los dibujos se ilustra un diagrama de conjuntos del SBA que representa una memoria de programa 3, entradas 1, unidad lógica 4, pila 5, estados almacenados 6 y salidas 8.

El SBA funciona como sigue:

20. 1. Las nuevas entradas se introducen desde el sistema controlado y quedan retenidas en retenes de entrada 2.

25. 2. El SBA extrae ahora las ecuaciones Boolean de la memoria 3 y, empleando la unidad lógica 4 y la pila 5, procesa los datos de entrada y estados almacenados a razón de un término Boolean cada vez para producir resultados que no recordados, como estados almacenados o colocados en una memoria tampón de salida 7.

30. 3. Las ecuaciones Boolean se toman de la memoria 3 término por término, y cuando todas las ecuaciones se han evaluado, los resultados se transfieren desde las memoria tampones de sa-

lida 7 hasta las salidas 8 y, por lo tanto, al sistema externo 9 controlado. El contador de localizaciones del programa se re- pone a cero y comienza el ciclo de nuevo.

Lógica de Combinaciones

5. Si el SBA se utiliza para emular la lógica de combina- ciones la ecuación Boolean que define la función lógica conten- drá solamente términos de entrada y términos de salida. Por ejemplo o:

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

10. $L = T + E.F. (X + -X.G)$

donde . significa conjunción lógica.

+ significa disyunción lógica

- significa negación (-D de lectura, no D)

A y L son las salidas.

15. todas las demás letras son entradas.

El número de etapas del programa necesarias para eva- luar las ecuaciones Boolean mencionadas, se puede reducir emplean- do un estado almacenado para ahorrar el valor de E.F. (X + -X.G) después que se ha evaluado la primera vez; este resultado par- cial se puede emplear en las segundas ecuaciones. Ahora tendre- mos:

a. estado almacenado = E.F.(X + -X.G)

b. $A = B.C.-D + \text{estado almacenado}$

c. $L = T + \text{estados almacenado.}$

25. Este ejemplo sirve para ilustrar la forma en que las ecuaciones se pueden optimizar a veces comprometiendo la memo- ria de estado almacenado contra la memoria del programa.

Lógica de Secuencias

30. Aunque algunos de los problemas son de combinación en naturaleza la basta mayoría de los problemas prácticos son de

secuencia.

5. Un sistema de secuencia es aquel en el cual la respuesta a un conjunto dado de condiciones de entrada depende de la historia anterior del sistema. Un ejemplo podría ser un reloj digital donde, normalmente, los dígitos de los segundos se incrementan excepto cuando llegan a 59, en cuyo punto se reponen a cero cero. En otras palabras la respuesta siguiente del contador depende de su valor actual y puede haber diferentes respuestas dependiendo de diferentes valores de corriente.

10. Todos los sistemas de secuencias se pueden describir por una red de combinaciones en la cual se almacenan en una memoria los resultados de las ecuaciones Boolean. Es precisamente esta memoria la que recuerda la historia o "estado" del sistema de secuencia.

15. Por lo tanto, la memoria de estado almacenado del SBA tiene uso principal para recordar el "estado" del sistema que se ejecuta. Cada vez que el SBA valúa el conjunto completo de ecuaciones Boolean que describe un sistema de secuencia, utiliza los estados almacenados como parte de las ecuaciones. Según prosigue la evaluación, los estados almacenados pueden cambiar si las ecuaciones Boolean lo exigieran.

20. Un ejemplo sencillo se tiene en una máquina expendedora automática donde existen dos estados principales:

- 25. 1. Cantidad insuficiente de dinero para comprar cualquier cosa.
- 2. Cantidad suficiente de dinero para expender los artículos.

30. En este ejemplo, se utilizaría algunos de los estados almacenados para contar el dinero que se ha introducido en la máquina. Las ecuaciones Boolean que controlan la distribución de los

5. artículos contendrían todas un término que comprendería el estado almacenado que nunca podría ser lógicamente verdadero si no existiera suficiente dinero para comprar el artículo. Tan pronto como se hubiera adquirido el artículo y este se hubiera suministrado, los estados almacenados que mantienen la "cantidad" de dinero, se alterarían para reflejar que el artículo se había suministrado conmutado, de éste modo, el sistema de nuevo al estado uno.

Descripción Detallada

10. El dibujo ilustra un diagrama de conjuntos del analizador Boolean de secuencia. Consiste en los componentes principales siguientes: 1. Una memoria de programa \mathcal{J} que mantiene el conjunto de ecuaciones Boolean que definen las operaciones del sistema. En una versión de circuito integrado de un solo bloquecito del SBA, este se encuentra contenido en un ROM programable por máscara en el bloquecito. Puede existir otra versión en la cual la memoria del programa esté fuera del bloquecito, y puede ser entonces del tipo ROM, RAM o PROM, según sea necesario. Las ecuaciones Boolean que definen las relaciones lógicas entre las entradas del SBA, estados almacenados y salidas, se almacenan en la memoria como palabra de 8 bits en una forma codificada. Más adelante se dará una definición exacta del código.

15.

20.

25. 2.- Un conjunto que alcanza hasta 30 memorias tampones 2 de entrada que quedan retenidas al comienzo de la evaluación de las ecuaciones Boolean. Se realiza de modo que los valores de entrada sean consistente durante todo el periodo de tiempo que lleva el evaluar todas las ecuaciones Boolean una vez.

30. 3.- Un número de páginas 61 a 64 de 30 basculadores de estado almacenado que se pueden agrupar para emular contadores y registradores de corrimiento, o utilizarse simplemente como "ban

0	1	1	0	1	0
1	0	1	1	0	0
1	1	0	1	1	1

Salida cierta cuando

Salida cierta cuando

5. las entradas difieren

las entradas se comparan

Tabla A

La unidad lógica tiene siempre dos entradas y produce una salida y existen dos tipos de acción.

10. (a) una entrada procede de un retén de entrada o estado almacenado; la segunda entrada procede de la parte superior 5l de la pila 5 (veáse el punto 5, más adelante) y el resultado se coloca en la parte superior de la pila.

15. (b) una entrada procede de la parte superior 5l de la pila 5; la segunda entrada procede del lugar siguiente de la pila, y el resultado se deja sobre la parte superior de la pila.

La operación exacta de los diversos códigos de evaluación de ecuaciones Boolean se definirá más adelante.

20. 5.- Una pila 5, que interviene siempre en ecuaciones lógicas como la parte superior 5l de la pila, es siempre uno de los operados para la unidad lógica. La pila es simplemente una pila de valores Boolean y se pueden imaginar como un registrador de corrimiento vertical en el cual se coloca siempre los datos en la parte superior o se extraen de la parte superior. Cuando se añaden datos a la pila, se dice que se "empujan" sobre la pila. El nuevo dato pasa a ser la parte superior de la pila y todos los datos anteriores son empujados hacia abajo en la pila un lugar. Si se realiza la operación inversa, el dato de la pila asciende un lugar y se dice que la pila "salta hacia arriba".
25. La parte superior de la pila se perderá y el dato que se encontraban anteriormente debajo del superior pasará a ser el nuevo
30.

dato superior de la pila.

La pila de datos en el SBA se utiliza normalmente como espacio de trabajo o acumulador, y la parte superior de la pila se utiliza en la mayoría de las instrucciones, algunas veces junto con el lugar siguiente, no obstante, la pila se puede utilizar también para almacenar variables Boolean temporales y ayuda enormemente a evaluar ecuaciones Boolean que contienen (ecuaciones agrupadas). Por ejemplo, una función como

$$A.(B.C. + (D.E. + F.G.) . (H.I + J.K))$$

se evaluaría del siguiente modo:

	<u>Operación</u>	<u>Resultado</u>
	(a) evaluación D.E.	D.E.
	(b) introducción en la pila	
	(c) valoración de F.G	
15.	(d) Disyunción	D.E. + F.G
	(e) introducción en la pila	
	(f) valoración de H.I + J.K. como en	(a)-(d)
	(g) Conjunción	(D.E. +F.G).(H.I.+J.K)
	(h) introducción en la pila	
20.	(i) valoración de B.C	
	(j) Disyunción	(B.C +(D.E +F.G).(H.I+J.K)
	(k) Conjunción y la pila con A	Resultado final

Naturalmente no este el único medio en que se puede utilizar la pila estando limitadas las variaciones solamente por la imaginación del usuario.

6. Se valora un conjunto que puede alcanzar hasta 30 memorias tampones de salida que tienen sus nuevos valores almacenados en las mismas ecuaciones Boolean. Cuando se completan todas las ecuaciones, al final del ciclo de un programa, los datos en las memorias tampones de salida se vuelven a retener en los exci

tadores de salida conectados al sistema controlado. De éste modo, las salidas externas del SBA se actualizan una vez por ciclo y permanecen sin cambiar hasta el final de la evaluación completa siguiente de las ecuaciones Boolean.

5. Entrada/Salida

Pueden existir hasta 30 entradas y hasta 30 salidas disponibles en el SBA. No obstante, debido a la limitación física de un dispositivo real, si existen más de un total de 30 entradas y salidas, las entradas y las salidas se multiplexan en las mismas patillas. Por lo tanto, el dispositivo pueden encontrarse disponible con 30 patillas programables en máscara a las entradas, salidas o entradas / salidas multiplexadas.

Circuito Lógico de Control

15. Asi como los bloques funcionales principales descritos anteriormente, el SBA contiene un circuito lógico de control simple que funciona de una forma evidente para el usuario.

Al final de un ciclo de programa completo, v.g., en una instrucción de "nueva iniciación", se realizan las acciones siguientes: (a) el contenido de las memorias tampones de salida se vuelve a retener en las salidas

(b) la parte superior de la pila se coloca en un "1" lógico

(c) el contador de páginas se pone en la página inicial

(d) se retienen nuevas entradas

(e) el contador de localizaciones del programa se pone en el

25. primer término de la primera ecuación Boolean.

Una vez que el SBA ha iniciado un ciclo de programa, el contador de localizaciones del programa simplemente aumenta cada ciclo de cronometración del SBA, tomándose lectura de la instrucción y actuándose sobre la misma.

30. Tiempo de Respuesta del SBA

5. Va implícito en la descripción del funcionamiento que la velocidad de respuesta del SBA a un sistema externo controlado está determinada por el tiempo que necesita para evaluar el conjunto completo de ecuaciones Boolean una vez. Esto se debe a que las entradas y las salidas se retienen solamente una vez por ciclo del programa del funcionamiento del SBA.

10. El SBA se diseña de tal modo que todas sus operaciones lógicas y transferencias de datos lleven el mismo tiempo de ejecución (de hecho uno de los ciclos de cronometración interna del SBA), y por lo tanto, el tiempo de respuesta de la salida del SBA a las nuevas entradas del sistema controlado por el SBA es directamente proporcional al número de operaciones Boolean necesarias para definir la función de control. Este tiempo de respuesta será normalmente del orden de unos cuantos milisegundos.

15. Trayectos de los Datos

20. Los trayectos de los datos disponibles en el SBA se ilustran en el dibujo. El punto focal en lo que se refiere a los datos es la parte superior de la pila, puesto que todos los datos se trasladan a la parte superior de la pila o proceden de la misma.

La pila se carga desde una entrada a estado almacenado.

La unión lógica que pueda realizar cualquier función lógica y el resultado se puede almacenar en cualquier memoria tampón de salida o estado almacenado.

25. Observese que no es posible que las ecuaciones Boolean emplean términos que comprenden datos en las memoria tampón de salida. Si esto fuera absolutamente necesario, entonces se debe una copia del estado de memoria tampón de salida en un estado almacenado de modo que se pueda realizar el trayecto del dato a la parte superior de la pila.

30.

La salida del SBA se pueden conectar de nuevo a las entradas bien directamente o por una circuiteria. Una conexión directa forma un estado almacenado al que se puede tener acceso desde el exterior del SBA; esto puede ser también una fuente limitada de estados almacenados extra. El circuito lógico externo se puede conectar entre salidas y entradas, o aun entre teclas y conmutadores. Esta última posibilidad es útil, por ejemplo, en la exploración de un conmutador de matriz o en la elección de un conmutador de códigos, y reduce la circuiteria externa.

5.

10.

No obstante, en general, la salida del SBA se conectan a las partes del sistema externo que controla sus acciones y las salidas del SBA se conectan a partes del sistema que verifican su estado actual.

Ecuaciones Boolean en la Memoria del Programa

15.

Para utilizar al máximo el espacio disponible para memoria, los códigos que representan las operaciones Boolean deberán ser lo más eficaces posibles. Se ha determinado que 20 instrucciones aproximadamente proporcionaría un buen compromiso entre el número y eficaces de las instrucciones. Suponiendo también que aproximadamente 30 entradas y salidas pueden ser apropiadas para las exigencias del tipo de sistema, que probablemente se controle por el SBA, se emplea el esquema siguiente.

20.

25.

Se utiliza una palabra de 8 bitios por cada código de instrucción. Cinco bitios proporcionan una localización para las entradas, salidas o estados almacenados, y si se reservan dos de las 32 localizaciones disponibles para instrucciones sin localización, los tres bitios restantes del código permiten disponer un total de 24 instrucciones.

30.

La palabra binaria de 8 bitios está representada convenientemente por tres digitos octales (radiz 8) del orden de 0 a

5. 3 377(00 000 000 a 11 111 111 binario). Los tres dígitos binarios menos expresivos se tratan como la instrucción y los 5 bits más expresivos como la localización. Las localizaciones 1 a 36 (octales) representan las 30 localizaciones necesarias en todo el SBA (1-30 en decimal, y las localizaciones 0 y 37 (todos 0 y todos 1 en binario) se reservan para instrucciones y según se ilustran en las tablas B y C a continuación.

CODIGOS DE ECUACIONES BOOLEAN

	<u>CÓDIGO DE MEMORIA EN OCTAL</u>	<u>MNEMONICO</u>	<u>DESCRIPCION FUNCIONAL DEL CODIGO</u>
10.		REINICIAR	Reiniciar la evaluación de ecuaciones
		INVERTIR	Invertir la parte superior de la pila
		PAGINA	Cambiar la página
15.		INICIAL	Vuelta a la página inicial
		EMPUJE 0	Empujar la lógica 0 sobre la pila
		EMPUJE 1	Empujar la lógica 1 sobre la pila
		EMPUJE C	Empujar y copiar la parte superior de la pila
		SOLTAR	Soltar la pila
20.		Y	Realizar la función en los dos lugares superiores de la pila. Almacenar resultado en la parte superior de la pila
		0	
		EXOR	
		COMP	
		PAND	Realizar la función
25.		POR	Realizar la función en los dos lugares superiores de la pila, Almacenar resultado en la parte superior de la pila
		PEXOR	
		PCOMP	
			Empujar el resultado al interior de la pila dejando la lógica 1 en la parte superior

30.

TABLA B

CODIGOS DE ECUACIONES BOOLEAN

<u>CODIGO DE MEMORIA EN OCTAL</u>	<u>MNEMONICO</u>	<u>DESCRIPCION FUNCIONAL DEL CODIGO</u>
5.	ANDIN	La entrada especificada por la localizacion se pone en conjunción con la parte superior de la pila.
	NANDIN	La inversa lógica de la entrada especificada por la localización se pone en conjunción con la parte superior de la pila
10.	ANDSS	El estado almacenado especificado por la localización* se pone en conjunción con la parte superior de la pila.
	NANDSS	La inversa lógica del estado almacenado especificado por la localización* se pone en conjunción con la parte superior de la pila.
15.	ASPI	El estado almacenado especificado por la localización* se pone en conjunción con la parte superior de la pila, entonces se empuja un 1 sobre la pila.
	NASPI	La inversa del estado almacenado especificado por la localización* se pone en conjunción con la parte superior de la pila y después se empuja un 1 sobre la pila.
20.	ALMACENAMIENTO	El valor de la parte superior de la pila se almacena en el estado almacenado especificado por la localización* y queda un 1 lógico en la pila.
25.	SALIDA	El valor de la parte superior de la pila se almacena en la memoria tampón de salida especificado por la localización y queda un 1 lógico sobre la pila.

*Nota: La localización de un estado almacenado define 1 de 30 en la página activada en ese momento.

TABLA C

30. Descripción de Códigos

Los códigos empleados para definir las ecuaciones Booleanas quedan comprendidos en 4 categorías:

(a) instrucciones de operandos simples que afectan a la pila.

5. (b) operaciones lógicas que toman entrada de la pila y almacenan el resultado en la pila.

(c) instrucciones de transferencias de datos para entradas, estados almacenados y salidas

(d) instrucciones de control de programa.

10. Categoría (a) - Manipulación de la pila. Vease la tabla D a continuación.

MANIPULACION DE LA PILA

	<u>INVERSION</u>									
15.		<table border="1"> <tr><td>A</td></tr> <tr><td>B</td></tr> <tr><td>Antes</td></tr> </table>	A	B	Antes	<table border="1"> <tr><td>\bar{A}</td></tr> <tr><td>B</td></tr> <tr><td>Después</td></tr> </table>	\bar{A}	B	Después	
A										
B										
Antes										
\bar{A}										
B										
Después										
20.	<u>EMPUJE \emptyset</u>	<table border="1"> <tr><td>A</td></tr> <tr><td>B</td></tr> <tr><td>Antes</td></tr> </table>	A	B	Antes	<table border="1"> <tr><td>\emptyset</td></tr> <tr><td>A</td></tr> <tr><td>B</td></tr> <tr><td>Después</td></tr> </table>	\emptyset	A	B	Después
A										
B										
Antes										
\emptyset										
A										
B										
Después										
25.	<u>EMPUJE 1</u>	<table border="1"> <tr><td>A</td></tr> <tr><td>B</td></tr> <tr><td>Antes</td></tr> </table>	A	B	Antes	<table border="1"> <tr><td>1</td></tr> <tr><td>A</td></tr> <tr><td>B</td></tr> <tr><td>Después</td></tr> </table>	1	A	B	Después
A										
B										
Antes										
1										
A										
B										
Después										
	<u>EMPUJE + COPIA</u>	<table border="1"> <tr><td>A</td></tr> <tr><td>B</td></tr> <tr><td>Antes</td></tr> </table>	A	B	Antes	<table border="1"> <tr><td>A</td></tr> <tr><td>A</td></tr> <tr><td>B</td></tr> <tr><td>Después</td></tr> </table>	A	A	B	Después
A										
B										
Antes										
A										
A										
B										
Después										
30.	<u>SOLTAR</u>	<table border="1"> <tr><td>A</td></tr> <tr><td>B</td></tr> <tr><td>Antes</td></tr> </table>	A	B	Antes	<table border="1"> <tr><td>B</td></tr> <tr><td>Después</td></tr> </table>	B	Después		
A										
B										
Antes										
B										
Después										

TABLA D

1. Invertir la parte superior de la pila. El valor Boolean se toma de la parte superior de la pila y se reemplaza por su inversa lógica, un 1 pasa ha ser un 0 y un 0 pasa ha ser un 1. La pila si aumenta ni disminuye.
5. 2. Introducción de 0 en la pila. Todos los valores de los datos en la pila son empujados hacia abajo un lugar y se introduce un 0 lógico en la parte superior de la pila.
3. Introducción de 1 en la pila. Todos los datos en la pila empujan hacia abajo un lugar y se introduce un 1 lógico en la parte superior.
10. 4. Introducción y copia de la parte superior de la pila. Los datos descienden en la pila y la parte superior de la pila anterior (que es ahora la posición siguiente en sentido descendente) se copia en la parte superior.
15. 5. Extracción de la pila. Todos los valores de los datos ascienden un lugar en la pila, y se pierde la parte superior anterior de la pila.

Categoría (b) - operaciones lógicas. Vease la tabla E a continuación.

20. OPERACIONES LOGICAS

PARTE SUPERIOR DE LA PILA = A función B

donde la función= Y, O, O EXCLUSIVO, COMPARACION

25.

A
B
C
Antes

A f B
C
Después

En el caso de empuje Y, empuje O, empuje O exclusivo, empuje de comparación

30.

A
B
C
Antes

1
A f B
C
Después

TABLA E

Existen dos tipos en éste caso: una función lógica (conjunción, disyunción, (Conjunción, Disyunción, Exor, Comp) y la función seguida por PUSH (Introducción).

5. CONJUNCION, DISYUNCION, EXOR, COMP. Los dos elementos superiores de la pila saltan a la unidad lógica; la función lógica se realiza con los mismos y el resultado se vuelve a introducir en la pila.

10. CONJUNCION-INTRODUCCION, etc. Los dos valores superiores de la pila saltan a la unidad lógica; se realiza la función apropiada con los mismos y el resultado se vuelve a introducir en la pila seguido de un 1 lógico. Esta forma de operación lógica se utiliza cuando se ha de ahorrar el resultado en la pila para proceso ulterior y el 1 se pone en la pila para dejarlo en disposición de evaluaciones adicionales.

15. Categoría (c) -operaciones de entrada/salida, Vease las tablas F y G a continuación.

OPERACIONES DE ENTRADA

20.	ANDIN A;	B	A . B
	ANDSS A;	C	C
		Antes	Después
	NANDIN A;	B	$\bar{A} . B$
	NANDSS A;	C	C
		Antes	Después
25.	ANDSS - A;	B	1
	PUSH 1	C	A . B
		Antes	C
30.			Después
	NANDSS - A;	B	1
	PUSH 1	C	$\bar{A} . B$
		C	
		Antes	Después

TABLA F

OPERACIONES DE SALIDA

ALMACEN A;

B
C
Antes

1
C
Después

5. El valor de B en la parte superior de la pila se almacena en estado almacenado de localización A y reemplazado por un 1 lógico.

SALIDA A;

B
C
Antes

1
C
Después

10. El valor de B en la parte superior de la pila se almacena en la memoria tampón de salida de la localización A y se reemplaza por un 1 lógico.

TABLA G

15. Estas instrucciones tiene dos partes, la orden y la localización. La orden define el trayecto de datos (entrada-pila, estado almacenado-pila, pila-estado almacenado o pila-salida) y se realiza a operación lógica, si la hubiera, La localización define cual los 30 elementos de datos se ha de manipular.
20. EL ALMACEN y SALIDA toman el valor lógico de la parte superior de la pila y los transfieren al estado almacenado apropiado o memoria tampón de salida reemplazándose la parte superior de la pila por un 1 lógico dispuesto para la evaluación siguiente. ANDIN toma el valor de la entrada localizada y lo pone en conjunción con la parte superior de la pila.
25. NANDIN toma la inversa lógica de la entrada localizada y la pone en conjunción con la parte superior de la pila. ANDSS toma el valor del estado almacenado localizado y lo pone en conjunción con la parte superior de la pila.
30. NANDSS toma la inversa lógica del estado almacenado localizado y

lo pone en conjunción con la parte superior de la pila. En las cuatro operaciones anteriores la pila ni aumenta ni se reduce. ASPI toma el valor del estado almacenado localizado, lo pone en conjunción, con la parte superior de la pila e introduce un 1.

5.

en la pila.
NASPI toma la inversa lógica del estado almacenado localizado, lo pone en conjunción con la parte superior de la pila y introduce un 1 en la pila.

Categoría (d) -control

10.

PAGINA. Existen normalmente 4 páginas de estados almacenados en el SBA y las instrucciones de escritura y lectura en los estados almacenados solamente proporcionan una localización dentro de la página actualmente activada. La instrucción de página hace funcionar el contador de páginas y activa la página siguiente.

15.

HOME. (Estado Inicial). Si fuera necesario activar una página de estado almacenados que ha pasado, la instrucción HOME hace que el contador de páginas vuelva a activar la página inicial. Por ejemplo, si el contador de páginas está activando actualmen

20.

te la página 3 y fuera necesario actualizar un estado almacenado 4 se utilizaría la instrucción de página. Si fuera ahora necesario un estado almacenado en la página 2 una instrucción HOME volvería a la página 1 y una instrucción de página saltaría a la página 2.

25.

REINIZACION. Este código es siempre el último código en la memoria del programa (observese que su valor es convenientemente todos 0) y cuando es observado por la única lógica de control, se realiza lo siguiente:

30.

(i) el contenido de las memorias tampón de salida se vuelve a retener en las salidas.

- (ii) se retiene un nuevo conjunto de entradas.
- (iii) la parte superior de la pila se pone en un l
lógico.
- 5. (iv) el contador de páginas se repone a la página ini
cial
- (v) el contador de localizaciones del programa se re-
pone para volver a iniciar la evaluación de ecuaciones Boolean.

ENTRADAS/SALIDAS

- 10. La capacidad de localización del SBA permite 30 entra-
das y 30 salidas. Existen disponibles 30 patillas como entra-
das/salidas en un paquete de 40 patillas y, por lo tanto, si
se han de utilizar más de un total de 30 entradas y salidas,
se deben multiplexar.
- 15. Existe una opción programable por máscara en cada una
de las 30 patillas para que puedan ser entradas, salidas o
entradas/salidas multiplexadas. Cualquiera que sea la opción,
la localización interna de las patillas permanece igual y así,
por ejemplo si una aplicación exige solamente una entrada y
20. la disposición del cuadro exige que fuera la última patilla,
el programa debe emplear la localización 30.
Entrada. Cuando se programa como entrada nada podrá sacarse
de la patilla aún cuando el programa cargue algo en la memo-
ria tampón de salida respectiva.
- 25. Salida. Cuando se programa como salida, el trayecto de dato
de entrada está conectado todavía y el valor en la patilla se
retendrá con todas las demás entradas en el instante apropia-
do. Este hecho se puede utilizar como sigue:
 - a) a veces es necesario que el valor de una salida se
utilice en el proceso durante el ciclo siguiente y normalmen-
30. te se tendría que copiar en un estado almacenado. Alimentando

una salida de nuevo como una entrada se evita de éste modo es te problema y se puede utilizar también como suministro limitado de estados almacenados extra, si se pudiera disponer de patillas de más, de lo que se puede también tomar lectura fuera del bloquecito.

(b) además de leer el valor de la salida, directamente la salida se puede modificar utilizando una construcción de drenaje abierto del excitador de salida. El valor de la salida se puede modificar por un dispositivo descendente activo conectado exteriormente e introducirse el resultado en la entrada. Algo así como un dispositivo manual para sobrepasar sería un modo simple utilizar esta característica, además de que se pueden realizar fácilmente funciones lógicas más complejas.

ENTRADA/SALIDA MULTIFLEXADA

Para una separación completa de entradas y salidas, la característica multiplexada se debe utilizar también aún cuando se debe emplear un circuito lógico externo para desmultiplexar.

TEMPORIZACION DEL DISPOSITIVO

La cronometración se puede generar interiormente, por ejemplo, con ayuda de una red de R/C externa conectada a patillas (reloj) o se puede utilizar un reloj externo conectado a una entrada de reloj.

El SBA realiza las instrucciones en el ROM, una por ciclo del reloj hasta que se alcanza la instrucción de reinización. El procesador tiene convenientemente una entrada de marcha para recibir una instrucción de marcha con el fin de iniciar un ciclo de proceso, y una patilla de detención para proporcionar una señal de detención cuando ha acabado un ciclo

de proceso.

Temporización de Entrada

Las entradas se modulan en el dispositivo una vez por ciclo de programa, si son entradas exclusivas o se multiplexan.

5.

Temporización de Salida

Si una patilla se dedica a función de salida, el nivel lógico permanece constante en todo el programa y se actualiza al final de cada ciclo.

10.

En un I/O, multiplexada, el dato de salida está presente en dos ciclos de reloj al final de cada ciclo de programa, proporcionando un control de I/O una modulación para que el dato se pueda almacenar en un dispositivo externo.

Modos de Operación

15.

(a) Si el SBA tiene que ciclar continuamente, la entrada de marcha será la correcta. En éste caso, existe un ciclo de reloj entre control de I/O que desaparece y el SBA modula las entradas. Este es el modo más común de operación.

20.

(b) Si no hay presente instrucción de marcha después que el control I/O es alto, aparecerá una instrucción de salida de detención y se detendrá el SBA. Cuando aparece la instrucción de marcha, desaparecerá la de detención y el SBA continuará enganchando a un nuevo conjunto de entradas. Aún cuando este modo puede ser útil, suele ser mejor poner un

25.

control de reinización en el propio programa, y funcionar continuamente.

30.

El SBA se puede fabricar como un circuito integrado por un proceso de N canales de implantación iónica de bajo voltaje, proporcionando una operación de gran velocidad y baja disipación de energía.

El SBA es microprogramable y puede formar el elemento básico de control para muchos sistemas que fijan temporización y funciones de control. Las entradas y salidas del SBA son preferiblemente compatibles con TTL. Normalmente el SBA puede tener 1023 palabras de programa, una pila de 16 elementos y una memoria de estados almacenados de 120 elementos. El proceso serial de entradas de información almacenada proporciona una programación fácil en el sistema lógico Boolean.

- 5.
10. Describa 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.

REIVINDICACIONES

5. 1.- Perfeccionamientos en procesadores de datos, para procesar un conjunto de señales de entrada de acuerdo con un conjunto de ecuaciones Booleanas, para proporcionar un conjunto de señales de salida, del tipo que comprende una circuiteria lógica para realizar una pluralidad de funciones lógicas y con disposición para ser controlado por una memoria de programa en la cual se almacena el conjunto de ecuaciones en forma de una pluralidad de etapas de instrucciones que se organizan para ser localizadas en secuencia, caracterizados porque cada procesador comprende, una pila para el almacenamiento temporal de datos que, en el uso del procesador, la lectura de entrada y salida se realiza en la parte superior de la pila.
10. 2.- Perfeccionamientos según la reivindicación 1, caracterizados porque la circuiteria lógica se organiza para recibir solamente dos entradas de bits simples y proporcionan tan solo una salida de un bit simple.
15. 3.- Perfeccionamientos según las reivindicaciones 1 ó 2 caracterizados porque la circuiteria lógica puede realizar las funciones lógicas de conjunción, disyunción, disyunción exclusiva y comparación.
20. 4.- Perfeccionamientos según las reivindicaciones 1, 2 ó 3, caracterizados porque se puede realizar la función lógica de negación para obtener la inversa de por lo menos uno de los valores de datos en el procesador en cualquier instante dado.
25. 5.- Perfeccionamientos según cualquiera de las reivindicaciones anteriores, caracterizados porque el dato en la parte superior de la pila es siempre un operando para la circuiteria
- 30.

6.- Perfeccionamientos según las reivindicaciones anteriores, caracterizados porque la salida de la circuitería lógica se almacena siempre en la parte superior de la pila.

5. 7.- Perfeccionamientos según la reivindicación 6, caracterizados porque se organiza para realizar una operación de introducción en la pila por la cual todos los valores en la pila son empujados en sentido descendente en la pila un lugar para dejar un valor deseado en la parte superior de la pila.

10. 8.- Perfeccionamientos según la reivindicación 7, caracterizados porque el valor deseado puede ser, discrecionalmente un "1" lógico, un "0" lógico, o el valor anterior en la parte superior de la pila.

15. 9.- Perfeccionamientos según las reivindicaciones 7 u 8, caracterizados porque la operación de introducción se puede combinar con una función lógica de la circuitería lógica.

20. 10.- Perfeccionamientos según cualquiera de las reivindicaciones anteriores, caracterizados porque se organiza para recibir una instrucción de reinización procedente de la memoria del programa al final del ciclo de proceso, por lo que el procesador se repone quedando dispuesto para comenzar otro ciclo.

25. 11.- Perfeccionamientos según cualquiera de las reivindicaciones anteriores, caracterizados porque comprende una pluralidad de retenes de entrada para dejar estáticos valores de entrada durante un ciclo del proceso.

30. 12.- Perfeccionamientos según cualquiera de las reivindicaciones anteriores, caracterizados porque comprende una pluralidad de memorias tampones de salida para retener valores de salida según se calculan durante un ciclo de proceso.

13.- Procedimiento según cualquiera de las reivindicaciones anteriores, caracterizados porque comprende una memoria de acceso aleatorio (RAM) para almacenar datos presentes en el procesador en cualquier instante dado.

5. 14.- Perfeccionamientos según la reivindicación 13, caracterizados porque la memoria de acceso aleatorio se organiza para ser localizada como una pluralidad de páginas cada una de las cuales contiene una pluralidad de localizaciones.

10. 15.- Perfeccionamientos según cualquiera de las reivindicaciones anteriores, caracterizados porque el procesador comprende la memoria del programa.

16.- Perfeccionamientos según la reivindicación 15, caracterizados porque la memoria del programa es una memoria de lectura solamente (ROM).

15. 17.- Perfeccionamientos según la reivindicación 15, caracterizados porque la memoria del programa es una memoria de lectura solamente programable (ROM).

20. 18.- Perfeccionamientos según la reivindicación 15, caracterizados porque la memoria del programa es una memoria de lectura solamente alterable eléctricamente (EAROM).

19.- Perfeccionamientos según cualquiera de las reivindicaciones anteriores caracterizados porque se fabrica como un circuito integrado de un solo bloquecito.

25. 20.- Perfeccionamientos según la reivindicación 19, caracterizados porque el circuito integrado tiene la forma de un paquete de cuarenta patillas.

30. 21.- Perfeccionamientos, según las reivindicaciones 19 ó 20, caracterizados porque el paquete de circuito integrado tiene por lo menos una patilla combinada de entrada/salida dispuesta para ser localizada por una técnica de multiplexa-

ción.

22.- Perfeccionamientos según cualquiera de las reivindicaciones anteriores, caracterizados porque se organiza para recibir una instrucción de marcha para iniciar un ciclo del proceso.

5.

23.- Perfeccionamientos, según cualquiera de las reivindicaciones anteriores, caracterizados porque se organiza para que proporcione una señal de detención cuando se ha acabado el ciclo del proceso.

10.

24.- Perfeccionamientos en procesadores de datos, tal y como queda sustancialmente descrito en la presente Memoria y en los dibujos adjuntos.

Esta Memoria consta de veintisiete hojas escritas a máquina por una sola cara.

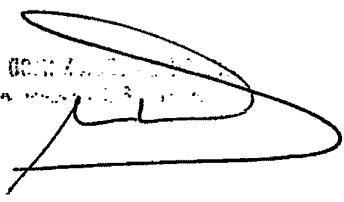
15.

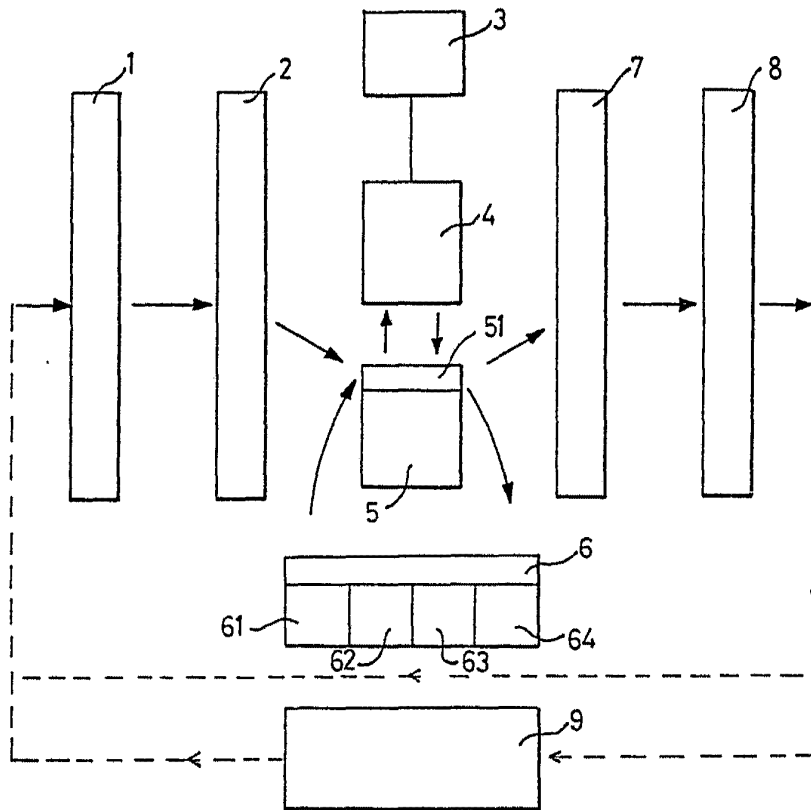
Madrid,

NOV. 1977

GENERAL INSTRUMENTS MICROELECTRONICS
LIMITED.

J. M. GONZÁLEZ
D. N.º 1.111.111





GENERAL INSTRUMENTS
VARIEDA

NOV 1977
L. M. GOMEZ AGUIRRE Y PARRA
Ingeniero en Electrónica

ESCALA VARIABLE.