



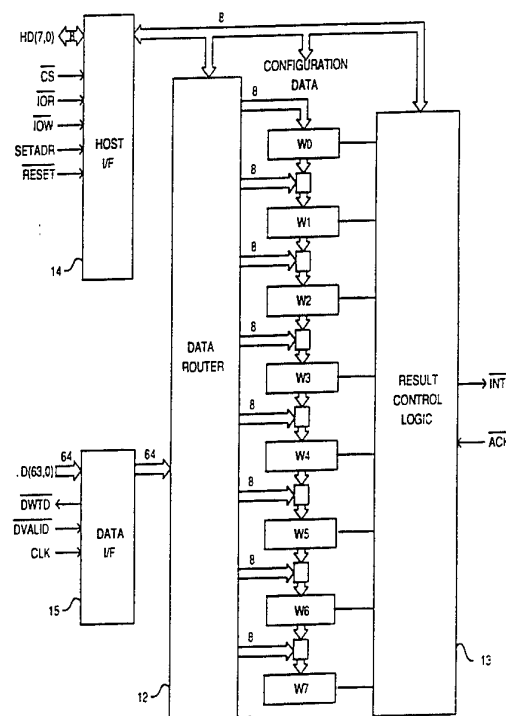
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁵ : G06F 15/40, 15/80	A1	(11) International Publication Number: WO 94/09443 (43) International Publication Date: 28 April 1994 (28.04.94)
(21) International Application Number: PCT/NO92/00173 (22) International Filing Date: 16 October 1992 (16.10.92) (71)(72) Applicant and Inventor: HALAAS, Arne [NO/NO]; Overlege Bratts vei 60, N-7026 Trondheim (NO). (74) Agent: J.K. THORSENS PATENTBUREAU A/S; P.O. Box 9276 Grønland, N-0134 Oslo (NO). (81) Designated States: CA, JP, KR, NO, US, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, SE).		Published <i>With international search report.</i>

(54) Title: NON-NUMERIC COPROCESSOR

(57) Abstract

A non-numeric coprocessor for fuzzy information retrieval and pattern recognition has means for information processing and is connectable to a host computer and a data source. A plurality of internal processing elements are organized in a number of simultaneously operable window modules (W0, W1, ...) arranged for inspecting data streams from said source. The processing elements compare data stream bytes with predetermined upper and lower bounds, to decide whether a byte is within said bounds, and, if so, to produce a hit signal. Each window module has a window match logic for correlating hit signals from its different processing elements, and to produce a window match signal by the occurrence of a predefined match. By structuring the coprocessor in this manner, a parallel processing potential is attained, which can be utilized by data routing means (12) to allow for separate data streams to be routed to individual or chained window modules selectably configured as groups of windows or super-windows according to application needs.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NE	Niger
BE	Belgium	GN	Guinea	NL	Netherlands
BF	Burkina Faso	GR	Greece	NO	Norway
BG	Bulgaria	HU	Hungary	NZ	New Zealand
BJ	Benin	IE	Ireland	PL	Poland
BR	Brazil	IT	Italy	PT	Portugal
BY	Belarus	JP	Japan	RO	Romania
CA	Canada	KP	Democratic People's Republic of Korea	RU	Russian Federation
CF	Central African Republic	KR	Republic of Korea	SD	Sudan
CG	Congo	KZ	Kazakhstan	SE	Sweden
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovak Republic
CM	Cameroon	LU	Luxembourg	SN	Senegal
CN	China	LV	Latvia	TD	Chad
CS	Czechoslovakia	MC	Monaco	TG	Togo
CZ	Czech Republic	MG	Madagascar	UA	Ukraine
DE	Germany	ML	Mali	US	United States of America
DK	Denmark	MN	Mongolia	UZ	Uzbekistan
ES	Spain			VN	Viet Nam
FI	Finland				

Non-numeric coprocessor

5

Technical Field

This invention relates to a non-numeric coprocessor for fuzzy
information retrieval and pattern recognition by means of
10 electronic computing devices.

Background Art

15 By means of complex programming and data structuring
techniques, traditional computer systems can be utilized for
describing, storing, recognizing and retrieving information.
However, by making use of such known methods, system perform-
ance often becomes very poor, especially for such tasks as
20 retrieval of complex items of information and recognition of
complex patterns. During information retrieval, text and
database searching, and pattern matching, most computer users
have faced well-known problems caused by the inefficiency of
the traditional approach to non-numeric computation.

25

For example, reading across huge sets of measurements to
answer queries of the type: "Find all dates and localities
where Cadmium is measured in the period 1980-85 at sites
where the concentration of Phosphorus is measured above 15
30 micrograms per liter", and even simpler queries than this,
may cause traditional systems to get into severe performance
problems.

Inspired by human behavior while searching through and
35 recognizing complex items in large sets of information, the
present invention seeks to remove the mismatches between
systems based on traditional methods and the performance
requirements of today. Thus, the object of the present

invention is to make provisions for simple non-numeric computation programming and at the same time provide for a capability of browsing huge data volumes at a rate which is substantially higher than that for traditional non-numeric computation systems, while looking for a specific full sentence, or, alternatively, a complex combination of many word fragments at a lower speed. Also, the present invention seeks to avoid the need for extensive segmentation, vectorization and duplicate storing of data for the purpose of searching for complex items of information, and which is often required when employing known techniques.

In the disclosure of the present invention, a 'byte' should be interpreted as a group of adjacent bits which is processed as a unit, the number of bits not necessarily being eight.

Disclosure of Invention

According to the invention the problems faced with prior art techniques are overcome by means of a non-numeric coprocessor for fuzzy information retrieval and pattern recognition, the coprocessor having means for information processing and being connectable to a host computer and a data source, and characterized in that said information processing means comprises a plurality of internal processing elements organized in a given number of simultaneously operable window modules arranged for inspecting streams of data from said data source, each processing element being designed for comparing one byte, e.g. one 8-bit byte, in a stream of data, with predetermined, individually programmable upper and lower boundary values assigned to said processing element, to decide if the value of the byte present in that processing element is within said boundary values, and, if so, produce a hit signal which is communicated to a window match logic provided in each window module for correlating hit signals received from different processing elements in that window

module, and to produce a window match signal by the occurrence of a predefined match in said window module.

By structuring the coprocessor in this manner, a powerful

5 parallel processing potential is attained, which can be utilized to give the coprocessor system a performance exceeding by far that of traditional systems for information retrieval and pattern recognition.

10 Preferably the coprocessor according to the invention further comprises data routing means allowing separate data streams from the source to be routed to said simultaneously operable window modules on an individual bases or in a manner whereby said window modules are chained into selectable ones of
15 different window configurations, such as individual super-windows, groups of super-windows, or a single super-window including all window modules, according to configuration data corresponding to application needs.

20 Chaining the windows into longer windows, gives the possibility of more complex retrieval conditions, and routing data input to the different windows may be done in several ways depending on the needs of the currently running application for window length and data streams. In fact, within the
25 limitation of the number of windows available, any number of streams each consisting of one byte, can be processed. For example, a number of individual data inputs, each preferably 8-bit wide, can be routed to different windows for parallel processing, avoiding the need for duplicate storing of data
30 when the same data stream is to be processed by different windows. Thus, the the coprocessor according to the invention can give a flexible, configurable data routing capability, which also supports using the processor in less than 64-bit wide applications.

35

Preferably the data routing means comprises a network of multiplexers organized in different levels, each multiplexer element being capable of selecting one out of two data inputs

to be routed to its output. In particular, in a preferred embodiment of the coprocessor the levels of multiplexers comprise a folding, a parallel and a serial multiplexer level, respectively.

5

The coprocessor may further comprise a static random access memory (RAM) for internal storage of said window configuration loadable into the coprocessor. Then there is no need for supplying configuration data to the coprocessor for each

10 search operation (only possible alterations in the configuration have to be communicated), and development of software including customized sets of downloadable configuration data becomes advantageous.

15 In an embodiment of the coprocessor, each processing element comprises a latch cell for temporary storage of the byte to be inspected, and two comparator cells loaded with said upper and lower boundary values for that processing element, the comparator cells being arranged to generate said hit signal.

20

It is advantageous to include in the coprocessor a result control logic arranged for receiving and comparing said window match signals with a programmable central hit mask allowing any logical combination of window hits to be defined
25 and addresses of all occurrences found (hit address mode), or alternatively the total number of matches within the data volume inspected (hit count mode), to be reported.

To those skilled in the art, the simplicity with which the
30 present coprocessor can be programmed and controlled, and further advantages and features of the coprocessor, will appear from the description given below.

35 Brief Description of Drawings

With reference to the accompanying drawings, the invention will now be described in detail by means of an example of a

preferred embodiment of the coprocessor according to the present invention, on which drawings:

Figure 1 shows a typical application of the coprocessor according to the invention,

5 Figure 2 is a block diagram of the coprocessor according to the invention, connected with a host computer and a data source,

Figure 3 is a diagram illustrating a single window in the coprocessor according to the invention,

10 Figure 4 is a diagram illustrating a single processing element of a window in the coprocessor according to the invention,

Figure 5 shows the data routing network in the coprocessor according to the invention,

15 Figures 6 - 15 serves to illustrate various preferred configurations of the data routing network in the coprocessor according to the invention,

Figures 16 and 17 serves to illustrate in detail the data flow through the routing network in the coprocessor, according to two different configurations shown in respective ones of Figures 7 - 16,

20 Figure 18 serves to illustrate a typical host/coprocessor configuration,

Figure 19 serves to illustrate the address map organization in the coprocessor, and

25 Figure 20 serves to illustrate a part of the window for a given application example.

30 Description of Preferred Embodiment

As shown in Figure 1, the coprocessor chip 1 according to the present invention typically is connected with a host computer 2 by means of a two-way data transfer link, and with a data source 3 by means of a one-way data transfer link.

The preferred embodiment of the coprocessor chip 1 includes, as can be seen in Figure 2, a series of eight window modules

W0 - W8 logically located between a data router module 12 and a result control logic 13 interconnected by an 8-bit data bus which is also connected with a host interface module 14. A data source interface module 15 provides for one-way 64-bit data transfer from the source 3 to the data router module 12.

Referring now to Figure 3, the eight data window modules W0 - W8 each contain a window match logic 16 and a 32 byte shift register for a corresponding number of processing elements PE0 - PE31. As shown in Figure 4, each processing element PE is divided into a latch cell 17 and two associated comparator cells 18, 19 for match checking with individually programmable upper and lower bounds.

By means of a data routing network, shown in detail in Figure 5, in the data router module 12, the windows can be chained into longer windows, giving the possibility of more complex data retrieval conditions. The data router comprises three levels of multiplexers, each selecting one out of two 8-bit wide inputs to be routed to its output.

The first multiplexer level, to which eight 8-bit wide source data streams are fed, is made up of a folding multiplexer (the upper mux row in Figure 5), i.e. a multiplexer that may fold data streams in a circular manner, thus making the windows independent of where the actual data stream is input to the coprocessor. The folding multiplexer is connected to a parallel multiplexer (the middle mux row in Figure 5) which may duplicate data streams to let different windows read the same data stream simultaneously. Finally a serial multiplexer (the bottom mux row in Figure 5) selects whether the windows should be chained into super-windows or not.

These three multiplexer levels make several combinations of inputs and chaining of windows possible. With reference to Figures 6 through 17, some of the possible configurations are described below, the input data streams being numbered 0

through 7, so that stream 0 corresponds to $D(7,0)$, stream 1 corresponds to $D(15,8)$, etc.:

- 5 a) Eight data streams, each supplied to one window in parallel. This is the simplest routing strategy. Each stream is routed to its corresponding window. This configuration is shown in Figure 6, and in detail in Figure 16 with the actual configuration in boldface.
- 10 b) Four data streams, each supplied to two windows in parallel. The data streams are routed in pairs to two windows in parallel. The streams can alternatively be data set 0,2,4,6 or data set 1,3,5,7; making stream 1 an extension of stream 0, and so on. This configuration is
15 shown in Figure 7.
- c) Four data streams, each supplied to two windows in series. The configuration is similar to that in b), except for each pair of windows being chained to form one
20 super-window of double length. This configuration is shown in Figure 8.
- d) Two data streams, each supplied to four windows in parallel. The streams are coupled together four-by-four
25 and routed to respective ones of two sets of four windows in parallel. The streams can alternatively be data set 0,4; 1,5; 2,6; or 3,7; making streams 1, 2 and 3 extensions of stream 0, and the streams 5, 6 and 7 extensions of stream 4. This configuration is shown in
30 Figure 9.
- e) Two data streams, each supplied to two parallel groups of two windows in series. As in d), but with chained windows and less parallelism. This configuration is shown in
35 Figure 10, and detail in Figure 17 with input streams 2 and 6 and the actual configuration in boldface.

- 5 f) Two data streams, each supplied to four windows in series. The configuration is similar to that in d), except for each group of windows being chained to form one super-window of four times the length of a separate window. This configuration is shown in Figure 11.
- 10 g) One data stream supplied to eight windows in parallel. The data streams are coupled together to form one stream routed to eight windows in parallel. Any of the eight streams can be used as input, treating the whole data storage as one long 8-bit data file. This configuration is shown in Figure 12.
- 15 h) One data stream supplied to four parallel groups of two windows in series. As in g), but with chained windows for more complex search. This configuration is shown in Figure 13.
- 20 i) One data stream supplied to two parallel groups of four windows in series. As in g) and h), but with even more windows chained for increased complexity. This configuration is shown in Figure 14.
- 25 j) One data stream supplied to eight windows in series. As in g), but with all windows connected together to form one single super-window for maximum search complexity. This configuration is shown in Figure 15.

30 The different filter and data path configurations are determined by the configuration data loaded into the chip. All configurations will be able to give 10 giga single byte comparisons per second due to the routing of data. However, complex queries are done by trade off with the number of simultaneous data paths. This makes it easy to map a large
35 number of applications onto the chip.

By means of a host interface 14, the coprocessor 11 is typically linked to the a computer in a manner as illustrated

in Figure 18. The host computer shown includes a disk unit 21 and an associated disk controller 22; a central processing unit, CPU, 23; a system memory 24, and a system bus 25. The host interface 14 is based on an 8-bit bidirectional port (HD-bus), and read and write cycles can be carried out asynchronously as well as synchronously. The host interface itself is controlled by the assertion of IOR or IOW in combination with CS and the appropriate polarity on the SETADR line (see Figure 2).

The configuration data is stored in a configuration RAM, and consists of a total of 828 bytes, making a complete reconfiguration possible in about 100 microseconds. In most systems the configuration time will be determined by the transfer rate from the host. Using the I/O channel of a personal computer will typically take 1000 μ s, assuming a transfer rate in the order of 1 MB/s.

Minor changes in configuration data can be done by addressing through the internal address register of the coprocessor, making configuration time even less. This gives the ability to search through the same amount of data with different criteria, since both reconfiguration and searching is very fast.

In count mode, the coprocessor will accumulate on chip the number of matching data items found. In report mode, the coprocessor will assert an interrupt signal upon detecting a hit. This signal will be kept asserted until ACK or IOW is asserted by the host. The internal result position counter will be stored in a shadow register (not shown). The configuration data determines if the chip should stop accepting data or not, whenever a hit occurs.

If the chip is configured to stop the data stream, DWTD (see Figure 2) will go inactive until ACK is asserted. When the data stream is programmed to flow non-stop despite a match, the counter stored in the shadow register will not be

overwritten until ACK has been asserted. This could be advantageous with text searches, where the hits most often cluster together in the portions of the text containing the desired data. When the problem is to find only the right
5 part (chapter, article) of the text, it may be irrelevant to find several accurate occurrences within that part of the text.

The 64-bit synchronous data interface 15 (see Figure 2) is controlled by a simple handshaking procedure. Whenever the
10 coprocessor is ready to accept data, DWTD is asserted one clock cycle ahead of the actual data read. This allows more comfortable timing when designing the interface.

15 If the data source has data ready, it asserts DVALID. Asserting DVALID when DWTD was inactive at the last rising clock, will not cause any data to be read by the coprocessor. Therefore the data source should not regard a data transfer as completed until the data and DVALID has been present
20 during the first rising clock after a rising clock edge with DWTD asserted. Corresponding timing schemes can be implemented for such synchronous or asynchronous read and write functions.

25 If the input stream contains data that is to be interpreted as numerical values spanning more than one byte, the most significant byte must arrive first into the window.

Programming of the chip is done by writing configuration data
30 to the different addresses in the host interface. The different parts of the configuration are indirectly addressable. This gives the ability to change only parts of the configuration whenever wanted, within a small amount of time. The data stream does not need to be stopped during a
35 reconfiguration, but false matches may occur due to the transient conditions present when the configuration is only partly written. There may also be a problem with misaligned records, since the record counters are reset by a write to

the corresponding window. It is therefore recommended not to alter the configuration without stopping the data stream.

The internal configuration address in the coprocessor consists of 11 bits, ADR(10,0), which is generated in an internal address register. The eight most significant bits of this register is loadable through the host interface. The three least significant bits will be cleared when the most significant bits are loaded. A load is done by setting up the address bits on the HD-bus, and asserting CS, IOW and SETADR simultaneously. The organization of the 11-bit address is shown in Figure 19.

The coprocessor has twelve internal modules, each with its own address. A module address comprises the four most significant bits in the address, which only changes by writing a new value from the HD-bus. The module base addresses are shown in Table 1. The seven least significant bits are held in a counter, incrementing after each access from the host interface.

Table 1 - Coprocessor Module Addresses

	Module Name	Module Address
25	Window 0	0000 (2)
	Window 1	0001 (2)
	Window 2	0010 (2)
	Window 3	0011 (2)
	Window 4	0100 (2)
30	Window 5	0101 (2)
	Window 6	0110 (2)
	Window 7	0111 (2)
	Hit Mask RAM	1000 (2)
	Mode Register	1001 (2)
35	Result Counter	"
	Hit Pattern	"
	Version Register	"
	Data Router Setup	1010 (2)
	Data Path	1011 (2)

40

Thus, consecutive bytes within one module may easily be accessed. Since there are holes in the address map between

modules, auto-incrementing of module addresses is not supported.

Within most modules offset addresses may be used for more detailed addressing. To access a single byte, the absolute address is calculated as:

$$\text{Address} = (\text{Module address} * 128_{(10)}) + \text{Offset address}$$

The value is then loaded into the address register through the HD interface, followed by accesses to auto-increment the address register. Read accesses are easiest since they do not require any knowledge of the previous configuration.

Each window module consists of:

- 32 lower limit bytes
- 32 upper limit bytes
- 32 bytes for field separator mask (In all bytes, only the least significant bit is significant.)
- 2 bytes for match latency value
- 1 byte for record length value

The offset addresses for these registers are shown in Table 2 below.

Table 2 - Window Address Offsets

Offset	PE	Register
0 (16)	0	Lower limit
1 (16)	0	Upper limit
2 (16)	0	Field separator (bit 0)
3 (16)	1	Lower limit
4 (16)	1	Upper limit
5 (16)	1	Field separator (bit 0)
:	:	
2A (16)	14	Lower limit
2B (16)	14	Upper limit
2C (16)	14	Field separator (bit 0)
2D (16)	15	Lower limit
2E (16)	15	Upper limit
2F (16)	15	Field separator (bit 0)
30 (16)	-	Match latency (LSB)
31 (16)	-	Match latency (MSB)
32 (16)	-	Record length

(cont.)

	Offset	PE	Register
	40 (16)	31	Lower limit
	41 (16)	31	Upper limit
5	42 (16)	31	Field separator (bit 0)
	43 (16)	30	Lower limit
	44 (16)	30	Upper limit
	45 (16)	30	Field separator (bit 0)
	:	:	
10	6A (16)	17	Lower limit
	6B (16)	17	Upper limit
	6C (16)	17	Field separator (bit 0)
	6D (16)	16	Lower limit
	6E (16)	16	Upper limit
15	6F (16)	16	Field separator (bit 0)

The limit registers are loaded with the appropriate values for the actual search. The field separator mask has a 1 in the position which is the least significant byte in each field. It should be noticed that in the field separator mask only the least significant bit in each byte is used, i.e. 1 should be written to the separator mask byte matching the processing elements, PE's, holding the last byte in a field, 0 elsewhere.

The match latency is the number of clock cycles during which a window should remember a hit. A match latency of zero implies that a hit is only reported from the window to the central match logic in the clock cycle it occurs. A match latency of e.g. 4 would imply that the match is reported for four additional cycles, i.e. five cycles in total. The value written to the match latency register should be $65535_{(10)}$ minus said latency, i.e. a match latency of e.g. 4 is specified by writing the value $65531_{(10)}$ to the register.

All routing of data streams takes the same amount of time, making the data streams appear at the input of the windows (or the first window of a chain) simultaneously, if they have been input to the chip simultaneously. When a stream is passed to a chained window, an accumulative delay of 32 cycles is introduced. This should be taken under consideration when calculating match latencies.

Clock cycles with no data transfers, i.e. DVALID inactive, will not contribute to the number of latency cycles. Thus, the match latency is measured in data transfers, not in physical clock cycles.

5

The record length is used to suppress hits due to occasionally matching patterns which are not aligned with record boundaries. By setting the record length to one, all hits will be reported to the central match logic. A record length of e.g. 6 will cause only matches occurring on every sixth data transfer to be reported from the window. The byte counting for a window is reset by all write operations with module address corresponding to that window; i.e. the first possible match would be six data transfers after writing the window configuration. The record length value is programmed into the corresponding register in the coprocessor as $256_{(10)}$ minus record length. Thus, a record length of $10_{(10)}$ is written as $246_{(10)}$.

20 The 256-bit hit **mask RAM** is organized as 32×8 bits as seen from the host interface. The 32 bytes are selected by the five least significant bits in the internal address register, leaving bit 5 and 6 as "don't care", and bit 7 through 10 as module address (1000_2). A writing to byte address 0, influences bit 0 through 7 in the 256-bit addressing scheme, with the least significant bit in the byte corresponding to bit position 0. Similarly, a write to byte address 4 affects bit 32 through 39 in the 256-bit addressing scheme.

30 The windows are addressing the RAM as 256×1 bit. The match signals from the eight windows are used as an address with the match from Window 0 being the least significant address bit. If a 1 is stored in the actual location, a hit is detected.

35

The module comprising the mode register, result counter, hit pattern and version register has no internal offset addresses, but is organized as a shift register chain with serial

read and write. All writes to this module will occur in the mode register, which is the only writeable in the chain. It affects how the coprocessor acts upon a hit. Only the three least significant bits have a function, the others should always be written as 0. The mode bits are explained in Table 3. These bits give 8 combinations of operation, all explained in Table 4.

Table 3 - Mode Register Bits

10	Bit	Symbol	Value	Operation
	0	Report	1	Report mode: Give interrupt on hit. Count data transfers.
15			0	Count mode: No interrupts. Count number of hits.
	1	Stop	1	Set DWTD inactive upon a hit, active after receiving ACK.
20			0	DWTD always asserted.
	2	Flank	1	Account only for first successive hits.
			0	Account for all hits.

25 Table 4 - Mode Bits Combinations

	Report	Stop	Flank	Operation
30	1	1	0	Generate an interrupt upon a hit. Count number of data transfers. Set DWTD inactive upon a hit until ACK is asserted.
35	1	0	0	Generate an interrupt upon a hit. Count number of data transfers. Keep DWTD active even upon a hit. Additional hits occurring before ACK is asserted will be lost.
40	1	1	1	Generate an interrupt only for the first hit in a continuous series. Count number of data transfers. Set DWTD inactive upon a hit until ACK is asserted.

45

(cont.)

	Report	Stop	Flank	Operation
5	1	0	1	Generate an interrupt only for the first hit in a continuous series. Count number of data transfers. Keep DWTD active even upon a hit. Additional hits occurring before ACK is asserted will be lost.
10	0	0	0	Generate no interrupts. Count number of hits occurring. Keep DWTD always asserted.
15	0	0	1	Generate no interrupts. Count hits, but only the the first of successive hits is accounted for. Keep DWTD always asserted.
20	0	1	0	Not recommended.
	0	1	1	Not recommended.

When reading from this module, the values are presented in the following order:

- 25 1. Result counter, byte 3 (most significant byte)
2. Result counter, byte 2
3. Result counter, byte 1
4. Result counter, byte 0 (least significant byte)
5. Hit pattern
- 30 6. Mode
7. Version number

All registers, except the mode register, are of the read-only type. Not all values have to be read after first
 35 initiating a read. Sending an ACK or IOW to the coprocessor will allow the values in the result counter and the hit
pattern to be overwritten by a new hit, as well as making new reads start with accessing the most significant byte of the result counter. (IOW must be qualified with CS.) This is also
 40 valid when INT have not been asserted.

The **result counter** is counting the number of hits or the number of data transfers that have occurred since the last write to the **mode register**. Thus, writing any value to this

module will clear the result counter. The counter is 32 bit wide. No counter overflow indication is given.

The hit pattern is the reported matches from each of the eight windows in the last hit. This can be used when the programming of the hit mask RAM allows several match patterns to generate a hit. The pattern that actually triggered the hit can be read to give more information to make post-processing easier.

10

The version register contains a number indicating which version of coprocessor that is present. This is for use with later revisions, giving the software a possibility to adjust to the hardware actually present.

15

The data router setup module is organized as a 3 level deep, one byte wide shift register. These are the bytes controlling the multiplexers described above with reference to Figures 5 through 17. The bytes are read and written in the following

20 order:

- 1) Serial multiplexer
- 2) Parallel multiplexer
- 3) Folding multiplexer

Read operations are destructive, i.e. all multiplexer configuration data must be rewritten after any readout. This is of little significance, since multiplexer configuration is only read during system testing.

The most significant bit in each byte corresponds to the leftmost multiplexer in Figure 5. Any combination of values is legal in the serial multiplexer. In the two others, configurations with more than four consecutive bits in one of the multiplexer control bytes, are illegal (due to propagation delays). This applies to the circular feedback, too, i.e. a multiplexer configuration of C3₍₁₆₎ is also illegal. The configuration bytes for each of the configurations a) through j) described above with reference to Figures

35

6 through 17, and which are guaranteed to work, are given in Table 5 below.

Table 5 - Multiplexer Setups

5	Config.	Input	Serial	Parallel	Folding
	a)	0,1,2,3,4 5,6,7	00 (16)	00 (16)	00 (16)
10	b)	0,2,4,6 1,3,5,7	00 (16) 00 (16)	AA (16) 00 (16)	00 (16) 55 (16)
	c)	0,2,4,6 1,3,5,7	AA (16) AA (16)	00 (16) 00 (16)	00 (16) 55 (16)
15	d)	0,4 1,5 2,6 3,7	00 (16) 00 (16) 00 (16) 00 (16)	EE (16) CC (16) 88 (16) 00 (16)	00 (16) 11 (16) 33 (16) 77 (16)
20	e)	0,4 1,5 2,6 3,7	AA (16) AA (16) AA (16) AA (16)	66 (16) 44 (16) 00 (16) 00 (16)	00 (16) 11 (16) 33 (16) 77 (16)
25	f)	0,4 1,5 2,6 3,7	EE (16) EE (16) EE (16) EE (16)	00 (16) 00 (16) 00 (16) 00 (16)	00 (16) 11 (16) 33 (16) 77 (16)
30	g)	0 1 2 3 4 5 6 7	00 (16) 00 (16) 00 (16) 00 (16) 00 (16) 00 (16) 00 (16) 00 (16)	0E (16) 1C (16) 38 (16) 70 (16) E0 (16) C1 (16) 83 (16) 07 (16)	F0 (16) E1 (16) C3 (16) 87 (16) 0F (16) 1E (16) 3C (16) 78 (16)
35					
40	h)	0 1 2 3 4 5 6 7	AA (16) AA (16) AA (16) AA (16) AA (16) AA (16) AA (16) AA (16)	06 (16) 1C (16) 18 (16) 70 (16) 60 (16) C1 (16) 81 (16) 07 (16)	F0 (16) C1 (16) C3 (16) 07 (16) 0F (16) 1C (16) 3C (16) 70 (16)
45					

(cont.)

	Config.	Input	Serial	Parallel	Folding
5	i)	0	EE (16)	00 (16)	F0 (16)
		1	EE (16)	1C (16)	01 (16)
		2	EE (16)	18 (16)	03 (16)
		3	EE (16)	10 (16)	07 (16)
		4	EE (16)	00 (16)	0F (16)
10		5	EE (16)	C1 (16)	10 (16)
		6	EE (16)	81 (16)	30 (16)
		7	EE (16)	01 (16)	70 (16)
15	j)	0	FE (16)	00 (16)	00 (16)
		1	FE (16)	00 (16)	01 (16)
		2	FE (16)	00 (16)	03 (16)
		3	FE (16)	00 (16)	07 (16)
		4	FE (16)	00 (16)	0F (16)
		5	FE (16)	C1 (16)	00 (16)
		6	FE (16)	81 (16)	00 (16)
		7	FE (16)	01 (16)	00 (16)
20					

The coprocessor chip according to the invention may also contain a data path module intended for fabrication tests only, and allowing read operations giving data output from window 7.

The preferred embodiment of a chip containing the non-numeric coprocessor according to the present invention, is a massive parallel VLSI-chip for installation on an expansion card for a personal computer or workstation. The chip is preferably a 100 pin PQFP package operating on +5V power supply, manufactured by a CMOS process and provided with TTL and CMOS compatible inputs. Having an operating frequency of 20 MHz such a coprocessor chip is capable of 160 MB/s sustainable data throughput, doing 10 giga single byte comparisons per second.

Programming Example

40

The following example illustrates the setup for searching for persons in a telephone directory. Figure 20 shows the configuration of a part of a window for the purpose of this example. This window will report a match for persons having a

- family name beginning with the character 'A' through 'G', and telephone number in the range of 142000 to 160000. By setting the appropriate bits in the field separation mask, the telephone number will be treated as a numerical field spanning six bytes. In the window shown, the other two fields are also indicated, but it would make no difference if all bits were 1 for fields #1 and #2 since they are not involved in any comparisons including more than one byte.
- 10 The bytes which are not contained in the search criteria, are set to a "don't care" condition matching all data patterns within the maximum range of $FF_{(16)}$ to $00_{(16)}$. A programmed record length of 16 bytes ensures that no matches appear for data not being aligned with record bounds in the configuration. As only one window is used, no match latency is needed.

If the above is the only search criteria, it may be copied to all windows, setting the hit mask RAM to have a 1 in all positions except position 0. This will cause a match signal to be generated whenever at least one of the windows has a hit. An appropriate mode value must also be set, e.g. Report=1, Stop=1, Flank=0. The coprocessor would then generate an interrupt for each match, and for each of these, the match position relative to the start of the search may be read.

Functional Description

- 30 As shown in Figure 3, the preferred embodiment of the coprocessor consists of 8 data windows, each containing a 32 byte shift register. As shown in Figure 4, each register element is associated with two comparators, checking for match with an upper and a lower bound. Each bound is individually programmable, and gives the opportunity of matching the data within any continuous interval in the byte range. The two comparators report a match to the match logic

connected to each window. For items larger than one byte, combinations of matches in different bytes can be combined. This gives the ability to handle data records of up to 256 bytes. Data fields may consist of up to eight bytes for any
5 interval test, and up to 256 bytes when testing for equality.

Each window reports its individual hit to the on-chip central hit mask. The eight individual window matches are used as an address in a 256 bit user programmable RAM. This RAM has a 1
10 stored for any combination of window hits that should be detected as a hit. By being programmable, the user can select to be informed for instance on a hit in only one window, in four out of eight windows, or in all windows. Generally, any logic combination of the 8 window hits can be a user defined
15 hit. The chip can report addresses of all occurrences found, or alternatively the total number of matches within the data volume.

When set in report mode, upon a hit, the coprocessor stores
20 in a shadow register the internal counter containing the position of the data found. This stored indication may later be read by the host. The shadow register will not be overwritten until the host has acknowledged the match by asserting ACK.

25

Each window can be set to remember a hit for a programmable time. This gives the ability of context sensitive searching in cases where exact matches cannot be found. This conforms with the principle of the coprocessor: Many weak conditions
30 upon the data wanted are used in combination instead of one (strict) search key. This feature is particularly important for searches in complex and/or huge amounts of data.

Examples of Applications

Searching In Unstructured Text

With the speed offered by the coprocessor, building and
5 maintaining traditional indices will become obsolete. The
combination of text fragments with wildcards, at controlled
relative distance, may uniquely identify important informa-
tion. Synonymous words may be concurrently used in a search.
Below follows two different types of simple queries:

10

Q1: "How many times and where does the text 'take it in what
sense thou wilt' occur in Shakespeare's works?"

Q2: "Find newspaper articles where at least 3 of the five
15 following (partial) names 'Jelts', 'Mitterr', 'Kohl',
'Major' and 'Bush' occur, but not 'Gorbat'."

Text style studies of type Q1 cannot efficiently be executed
by traditional text searching systems. It is also well known
20 that indices may require more space than the texts alone.

The coprocessor according to the given example will accept a
sustained data rate of 160 MB/s for queries of type Q1.
Complex queries of type Q2 are processed at a sustained data
25 rate of 20 MB/s.

Pattern Matching

From images, e.g. fingerprints, different types of features
are extracted. The combination of many such features
30 identifies an object. A search among a large number of
candidates may greatly benefit from the robust, fuzzy
mechanism of the present coprocessor. In general, the
abilities of the coprocessor are well suited to problems
involving partial matching, e.g. as done in DNA research.

35

Image Archives

The need for storing and handling images of various types is
pushing technological developments on many fronts. Efficient

image retrieval systems rapidly becomes crucial, e.g. in hospitals, for newspapers and for real estate agents. Images may represent multidimensional objects with considerable amounts of derived and added properties attached to them.

5

Data Base Searching

Most database systems rely on hierarchical structures and strictly defined identifiers. Fuzzy queries, where each attribute has low selectivity, pose severe performance problems in existing systems. Such queries are ideal for the present coprocessor, combining all weak constraints on the fly.

Concrete studies, e.g. on databases for environmental measurements and for chemicals, have shown that the potential for simplification and for increased performance is radical.

Signal Processing

Potential applications include non-linear filtering, radar target correlations and detection of abnormal signals.

Data Networks

Potential applications are parasitic watch-dog functions, e.g. to report illegal address ranges or to snap information. Several personal computers and workstations equipped with the present coprocessor can also use a network as a data source emerging from a central data pump which periodically broadcasts the whole amount of data, taking away all problems of coherence in distributed systems.

30

Disk Controllers

The coprocessor according to the invention is an ideal disc controller component. It may drastically reduce the need for transferring data via buses to the host computer, simply by restricting this data to positively requested items only. The coprocessor functions go beyond classical content addressing to a more advanced "data property" addressing.

Claims

1. A non-numeric coprocessor (1) for fuzzy information retrieval and pattern recognition, the coprocessor having
5 means for information processing and being connectable to a host computer (2) and a data source (3),
c h a r a c t e r i z e d i n that said information processing means comprises a plurality of internal processing elements (PE0, PE1, ...) organized in a given number of
10 simultaneously operable window modules (W0, W1, ...) arranged for inspecting streams of data from said data source (3), each processing element being designed for comparing one byte, e.g. one 8-bit byte, in a stream of data, with predetermined, individually programmable upper and lower
15 boundary values assigned to said processing element, to decide whether the value of the byte present in that processing element is within said boundary values, and, if so, to produce a hit signal which is communicated to a window match logic (16) provided in each window module (W0, W1, ...)
20 for correlating hit signals received from different processing elements (PE0, PE1, ...) in that window module, and to produce a window match signal by the occurrence of a predefined match in said window module.
- 25 2. A coprocessor according to claim 1,
c h a r a c t e r i z e d i n that said coprocessor further comprises data routing means (12) allowing separate data streams from the source (3) to be routed to said
30 simultaneously operable window modules (W0, W1, ...) on an individual basis or in a manner whereby said window modules are chained into different, selectable window configurations, such as individual super-windows, groups of super-windows, or a single super-window including all window modules, according to configuration data corresponding to application needs.
- 35 3. A coprocessor according to claim 2,
c h a r a c t e r i z e d i n that said data routing means (12) comprises a network of multiplexers organized in

different levels, each multiplexer being capable of selecting one out of two data inputs, each preferably 8-bit wide, to be routed to its output.

5 4. A coprocessor according to claim 3,
c h a r a c t e r i z e d i n that said levels of multi-
plexers comprises a folding, a parallel and a serial multi-
plexer level, respectively.

10 5. A coprocessor according to claim 2,
c h a r a c t e r i z e d i n that said coprocessor
further comprises a static random access memory (RAM) for
internal storage of said window configuration loadable into
the coprocessor.

15 6. A coprocessor according to claim 1,
c h a r a c t e r i z e d i n that each processing
element (PE0, PE1, ...) comprises a latch cell (17) for
temporary storage of the byte to be inspected, and two
20 comparator cells (18, 19) loaded with said upper and lower
boundary values for that processing element, said comparator
cells being arranged to generate said hit signal.

7. A coprocessor according to claim 1,
25 c h a r a c t e r i z e d i n that said coprocessor
further comprises a result control logic (13) arranged for
receiving and comparing said window match signals with a
programmable central hit mask supporting the definition of
any logical combination of window matches and allowing the
30 report of addresses of all occurrences found (hit address
mode), or alternatively the report of the total number of
matches within the data volume inspected (hit count mode).

8. A coprocessor according to claim 1,
35 c h a r a c t e r i z e d i n that each window module
(W0, W1, ...) is designed to contain a record length value
for a data record present in said window module, a field
separator mask to separate the fields of said data record,

and a match latency value enabling each window to be set to remember a hit for a programmable length of time.

9. A coprocessor according to claim 1,

5 c h a r a c t e r i z e d i n that said number of window modules (W0, W1, ...) is eight, each of which being designed to handle 8-bit wide byte inputs and comprising 32 processing elements (PE0, PE1, ...) with a shift register of corresponding length for the data stream supplied to each window
10 module.

10. A coprocessor according to any preceeding claim, c h a r a c t e r i z e d i n that said coprocessor further comprises:

- 15 - host interface means (14), preferably an 8 bit interface with interrupt capabilities, designed for allowing use of the coprocessor with any microprocessor; and
- data source interface means (15), preferably an 64 bit interface, allowing the connection of the coprocessor to
20 any high-speed data source, i.e. RAM-banks, disc arrays, or a network, the coprocessor preferably being programmable for 64-, 56-, 48-, 40-, 32-, 24-, 16- or 8-bit data transfers.

1/9

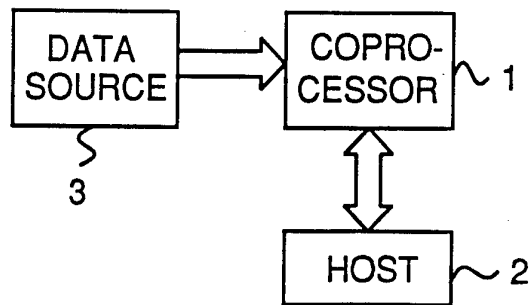


Fig. 1

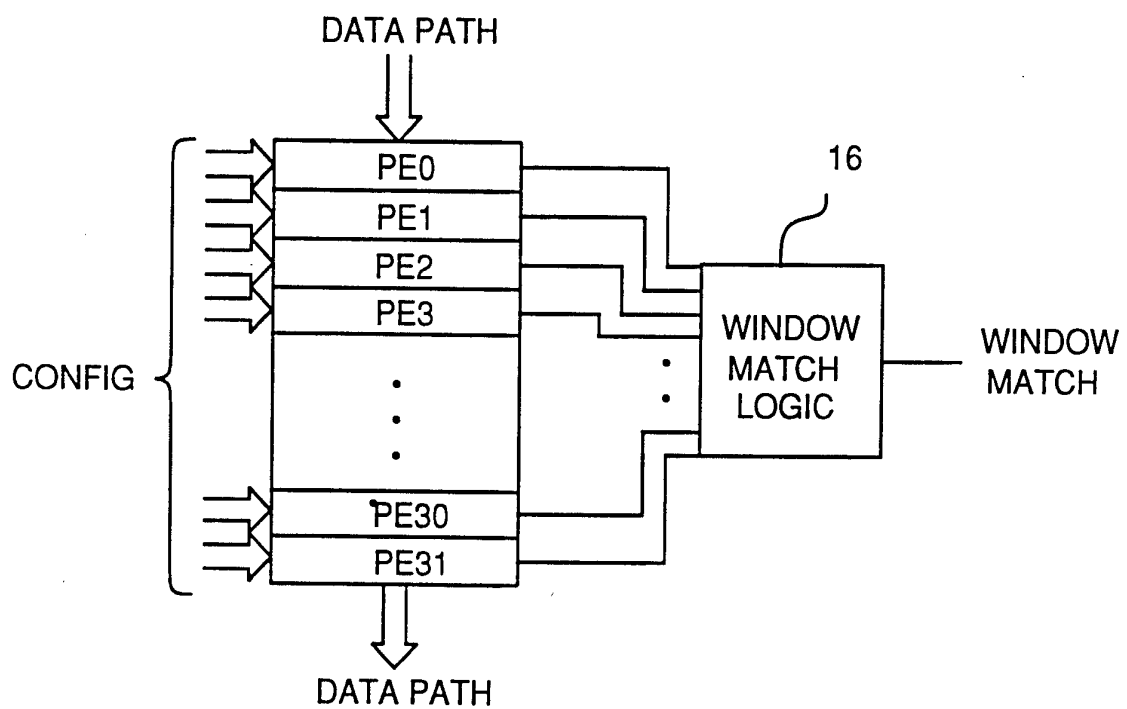


Fig. 3

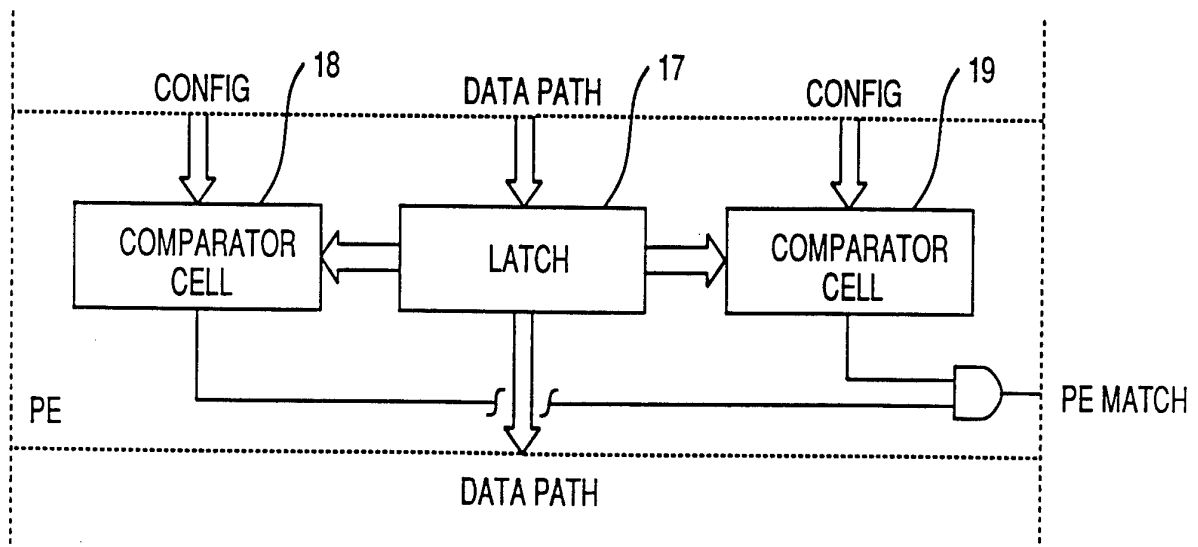


Fig. 4

SUBSTITUTE SHEET

2/9

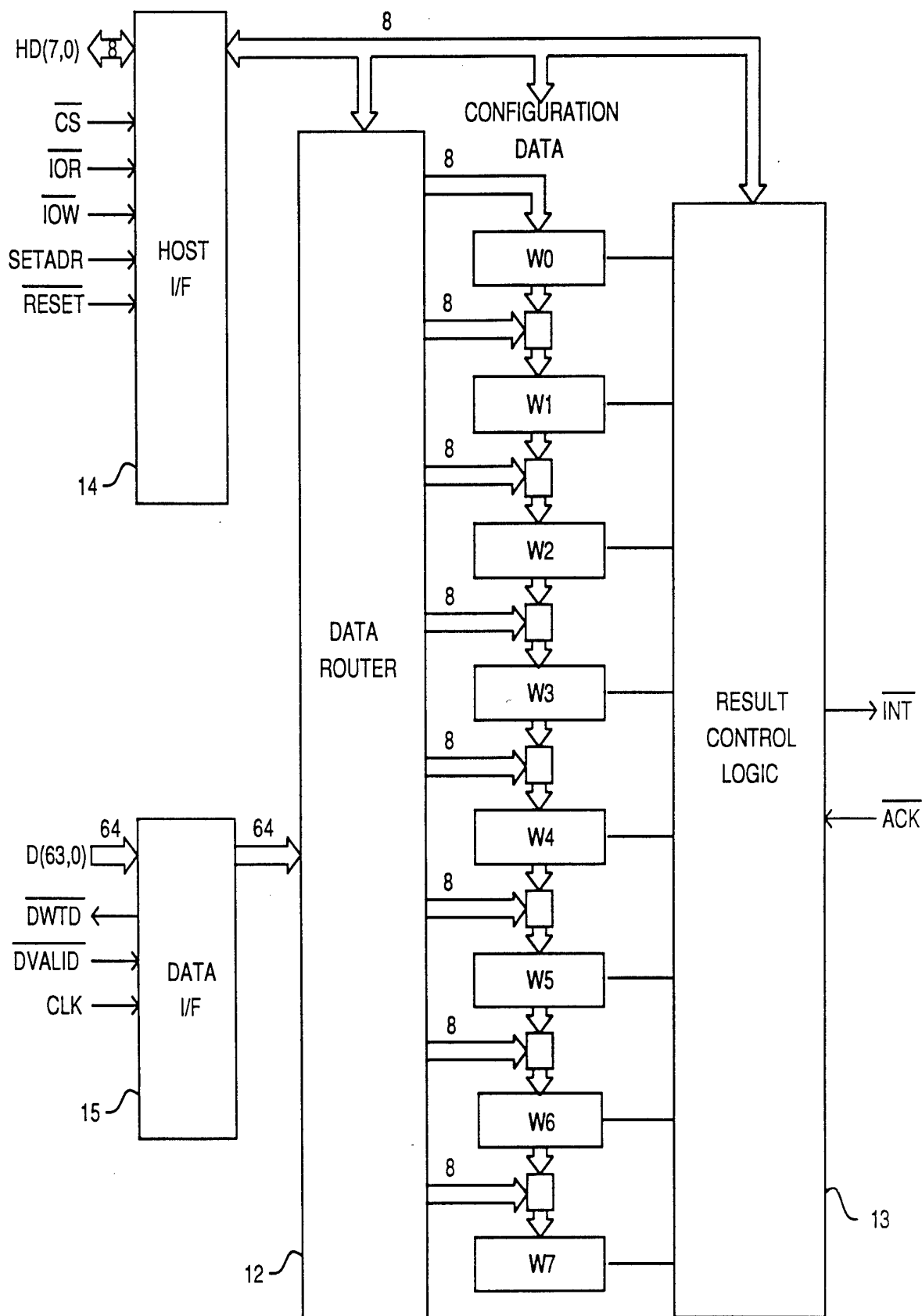


Fig. 2

SUBSTITUTE SHEET

3/9

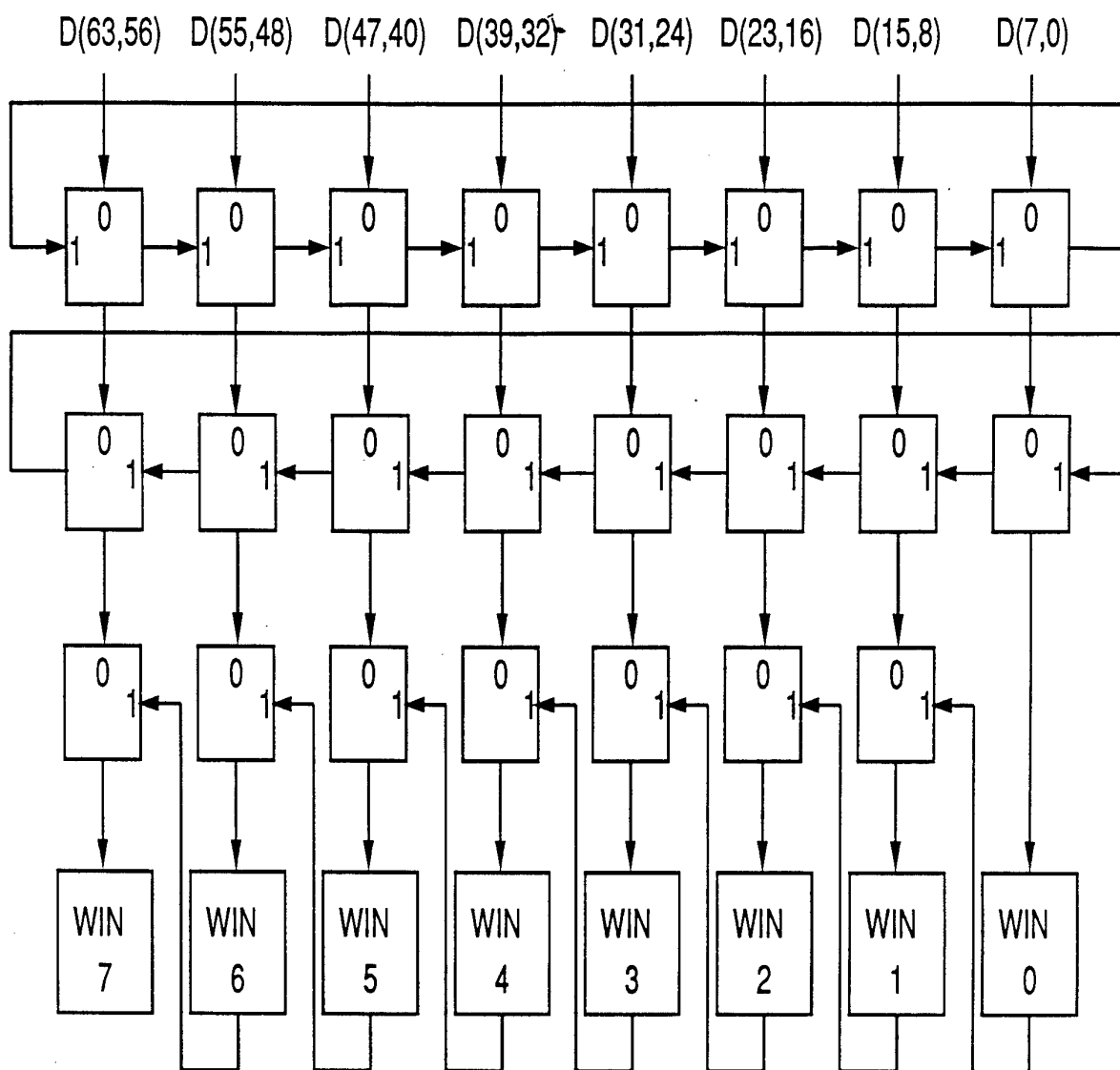


Fig. 5

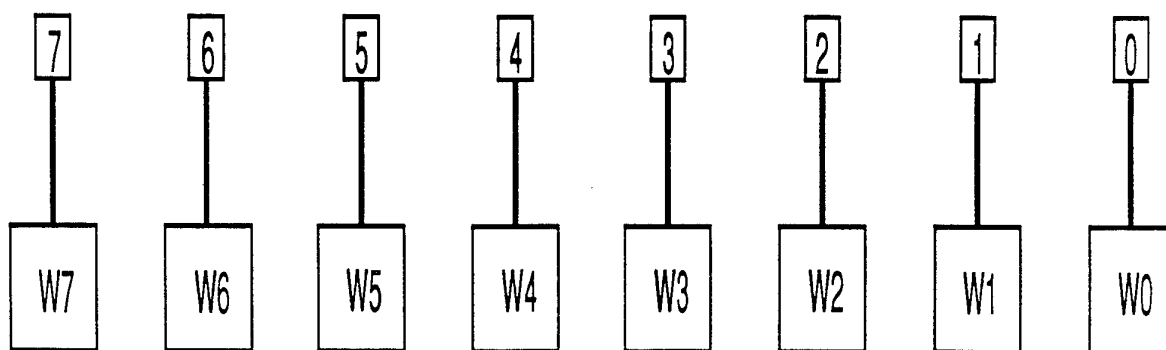


Fig. 6

SUBSTITUTE SHEET

4/9

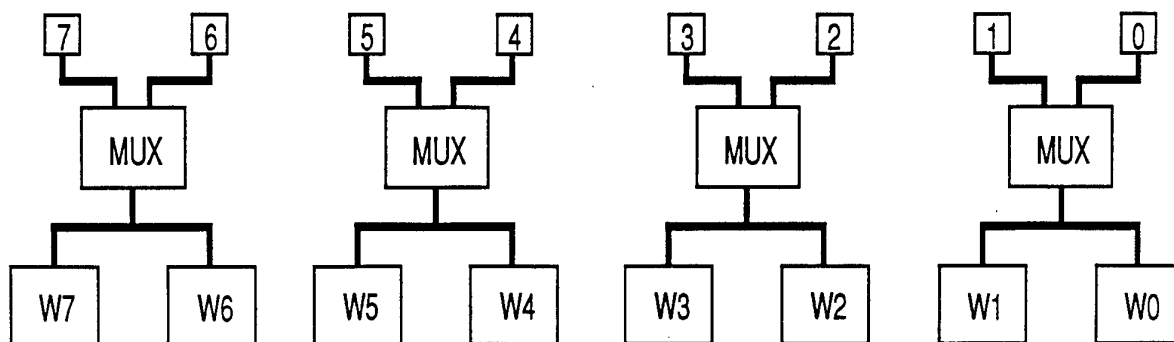


Fig. 7

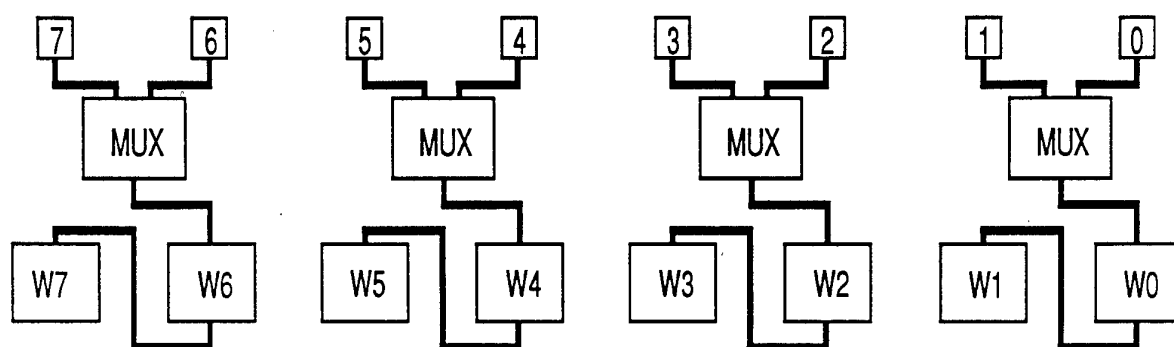


Fig. 8

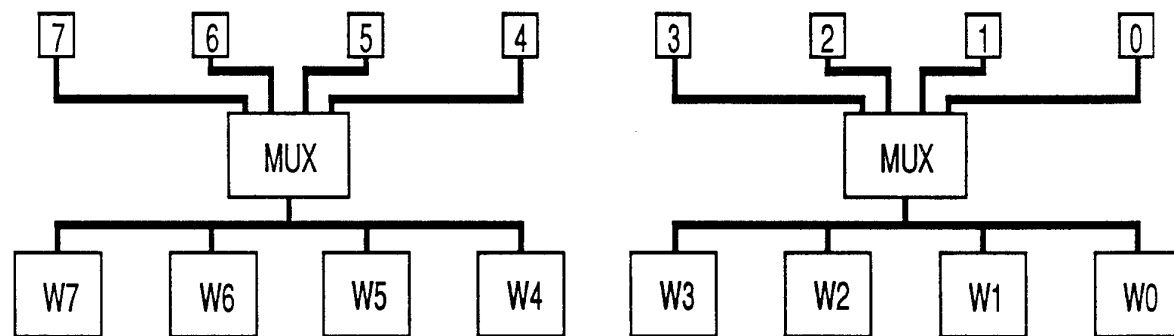


Fig. 9

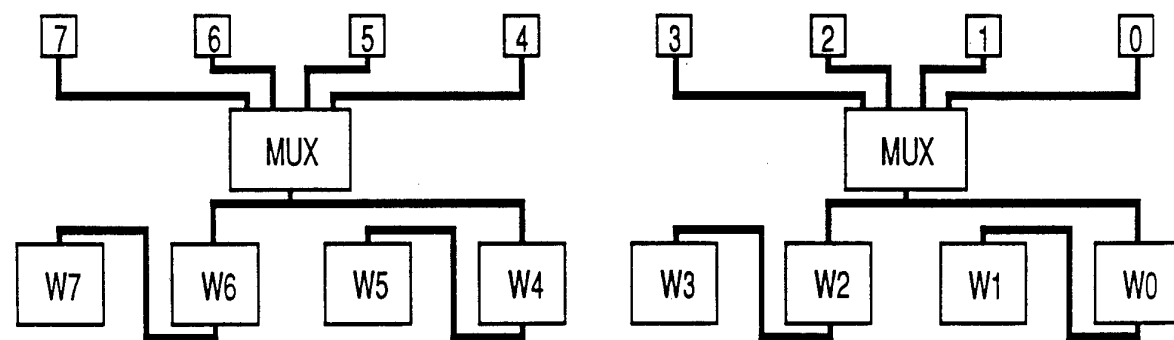


Fig. 10

5/9

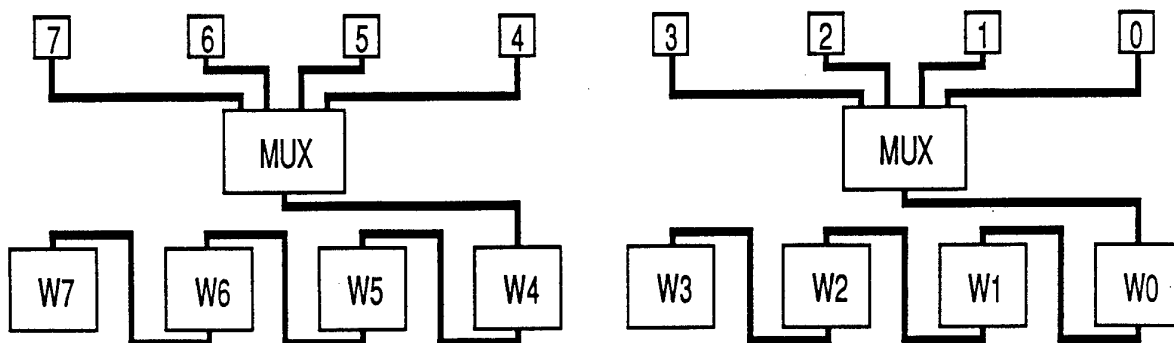


Fig. 11

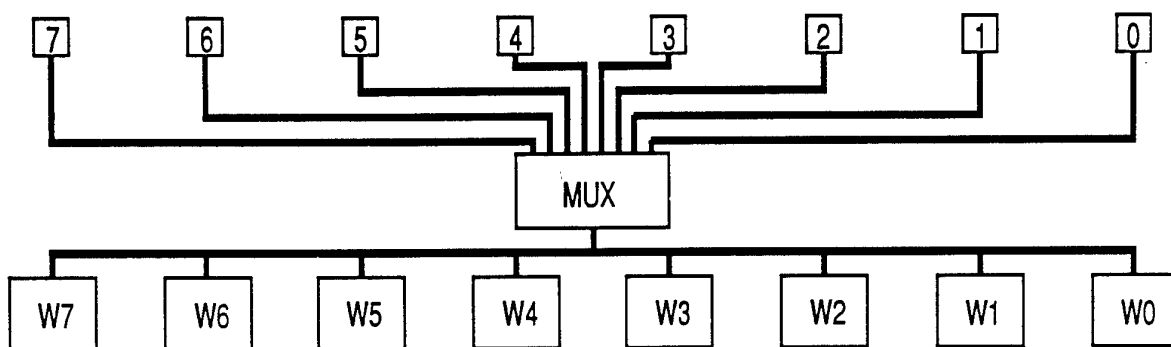


Fig. 12

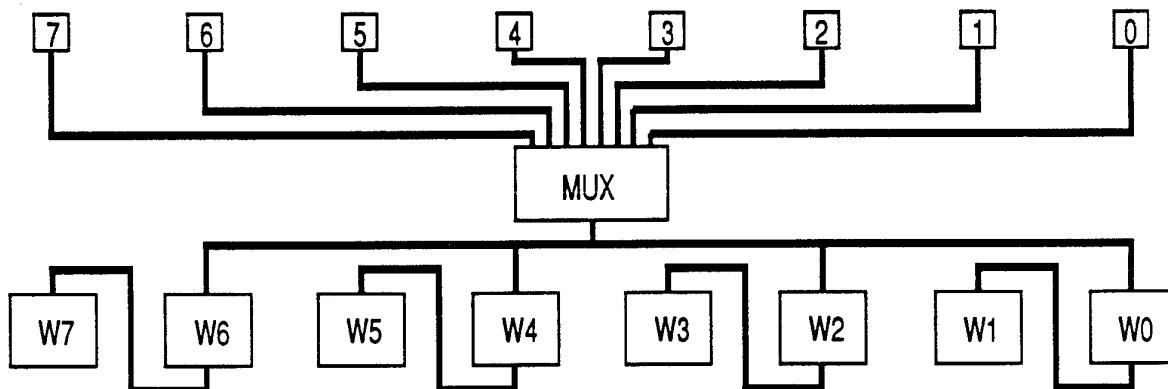


Fig. 13

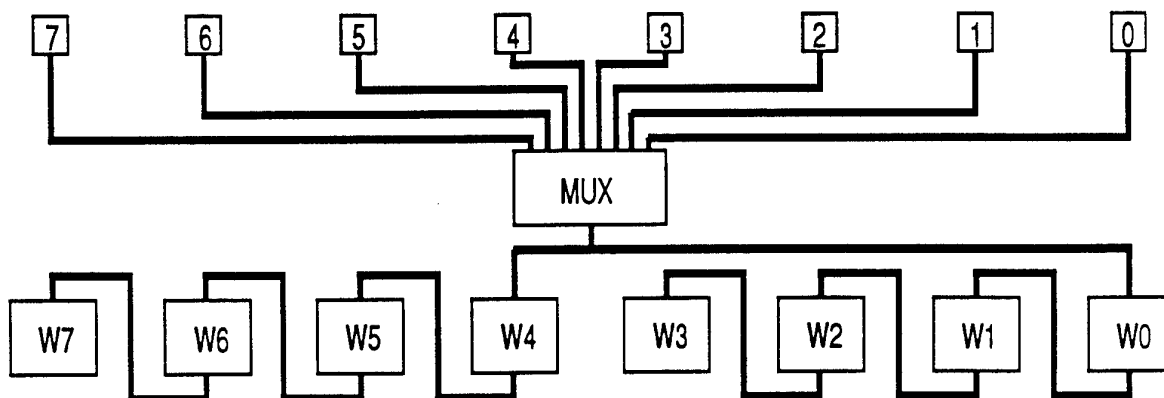


Fig. 14

6/9

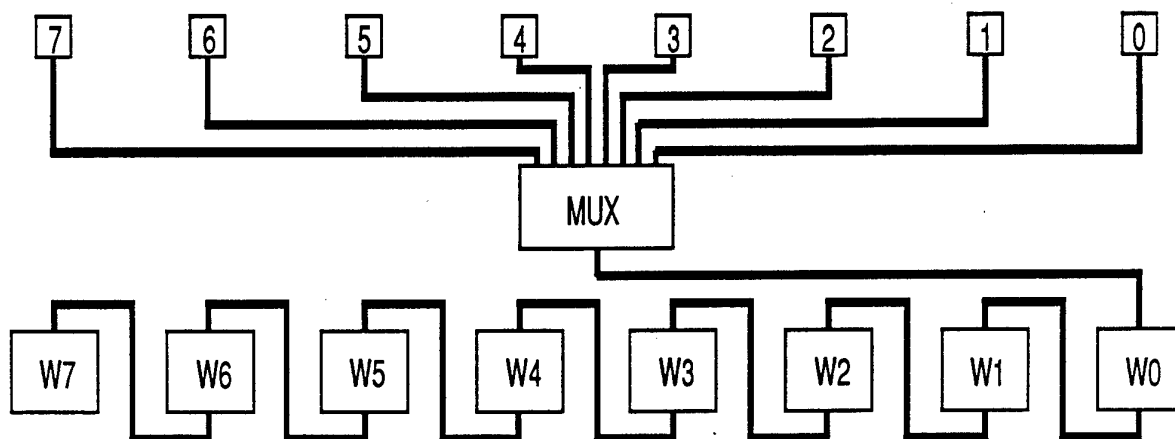


Fig. 15

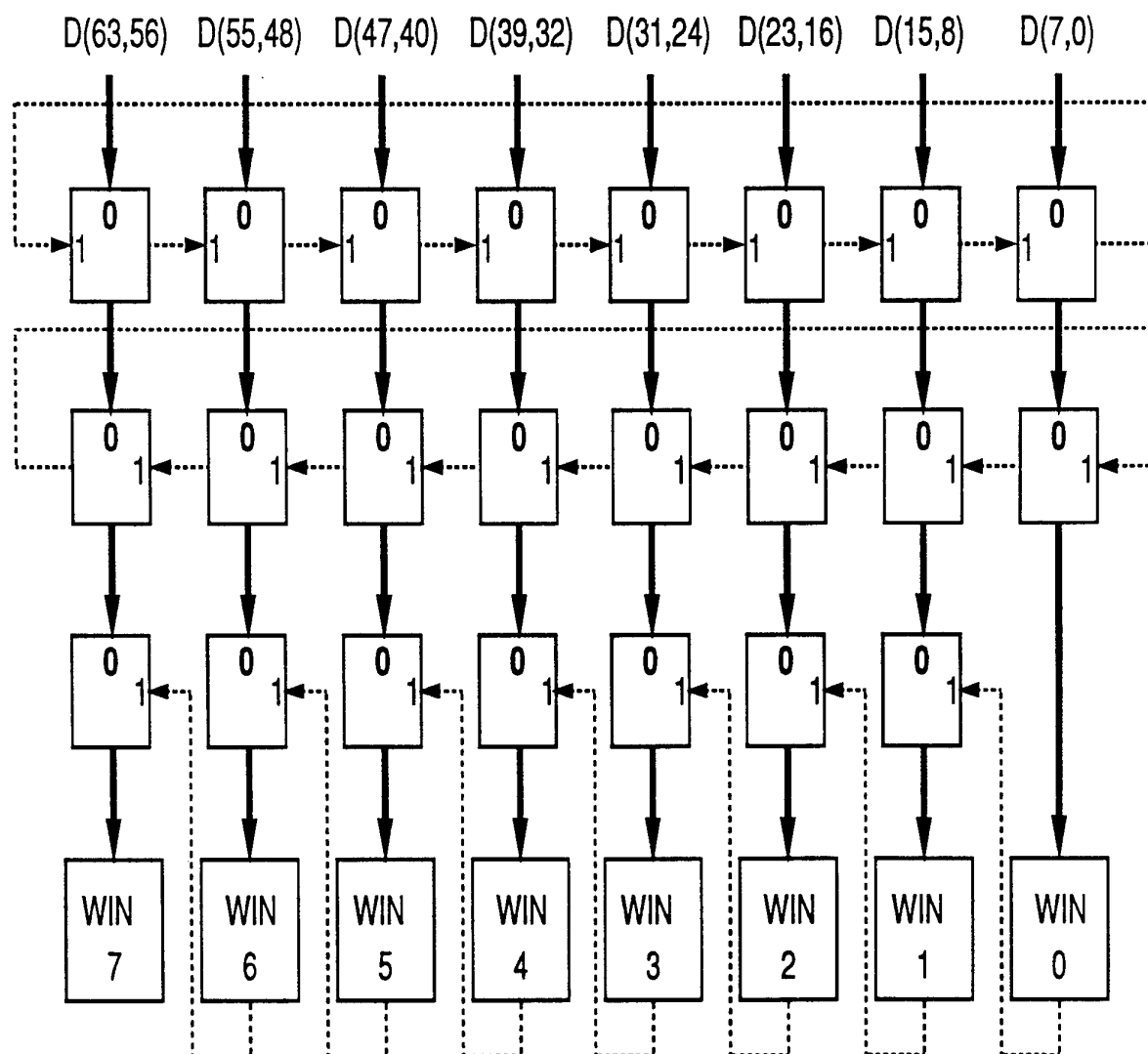


Fig. 16

SUBSTITUTE SHEET

7/9

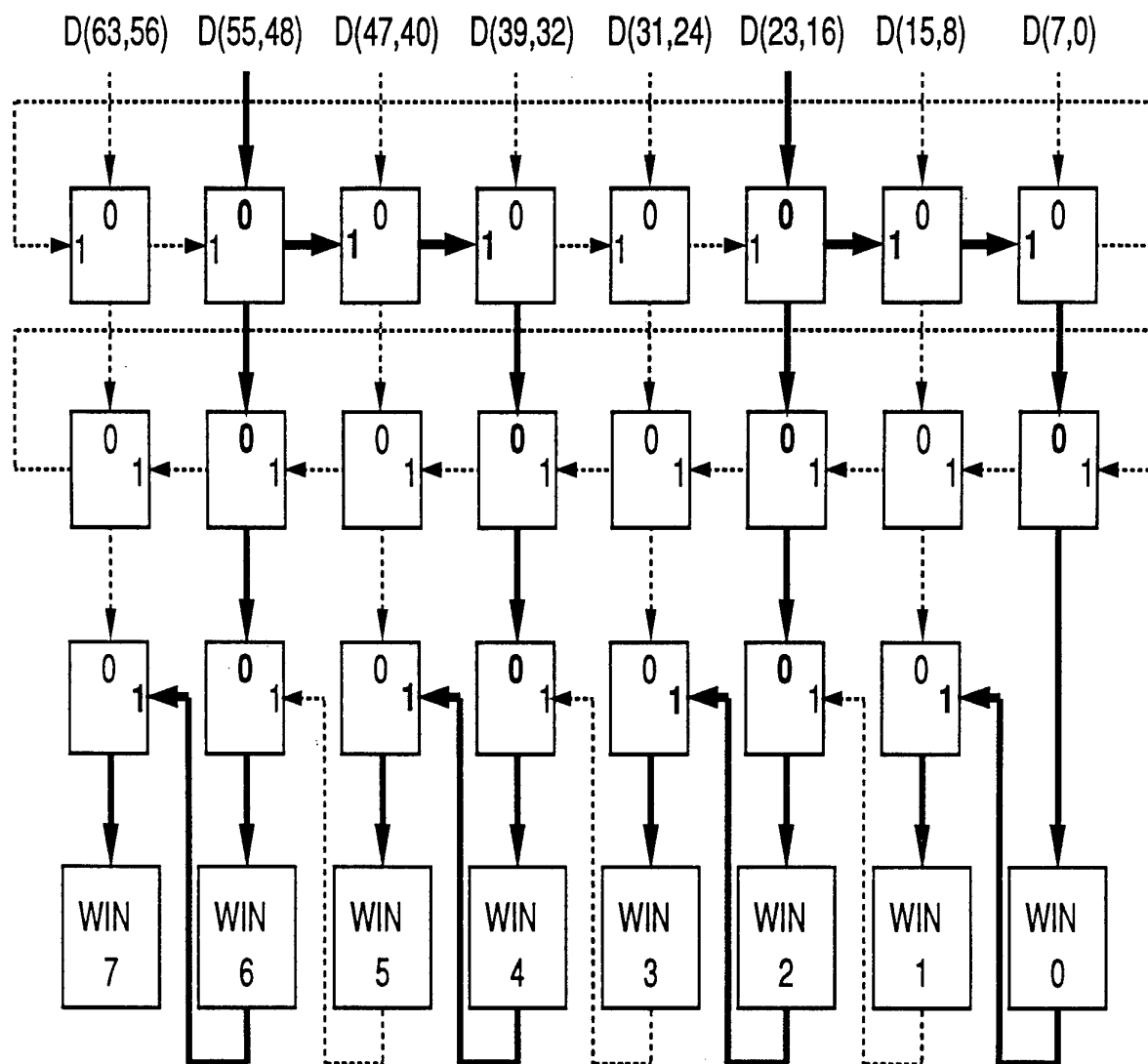
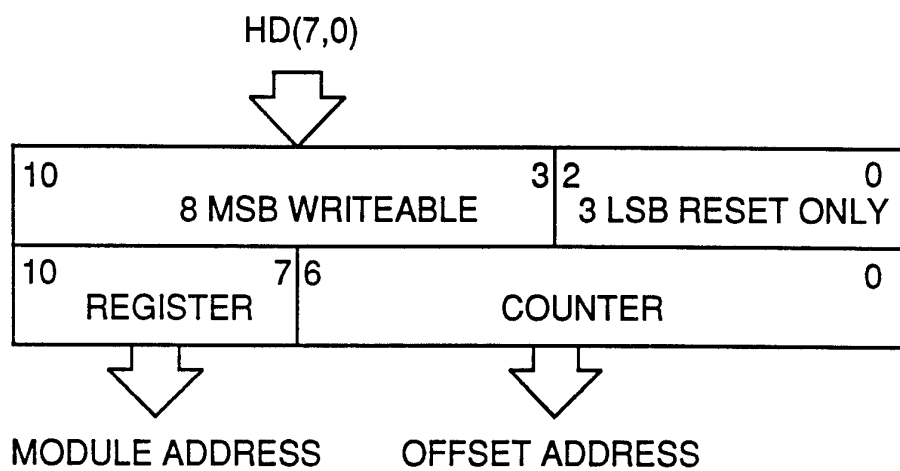
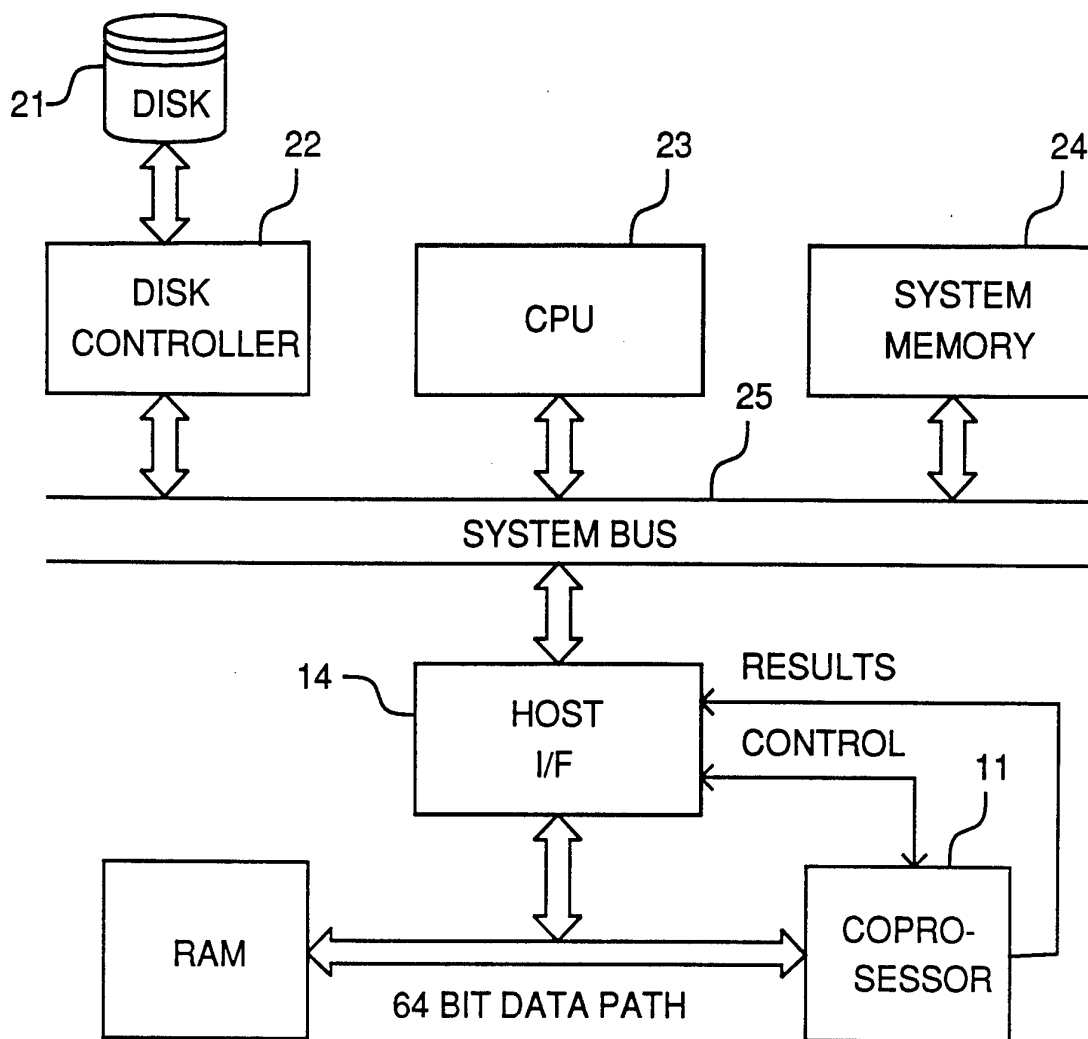


Fig. 17

8/9



SUBSTITUTE SHEET

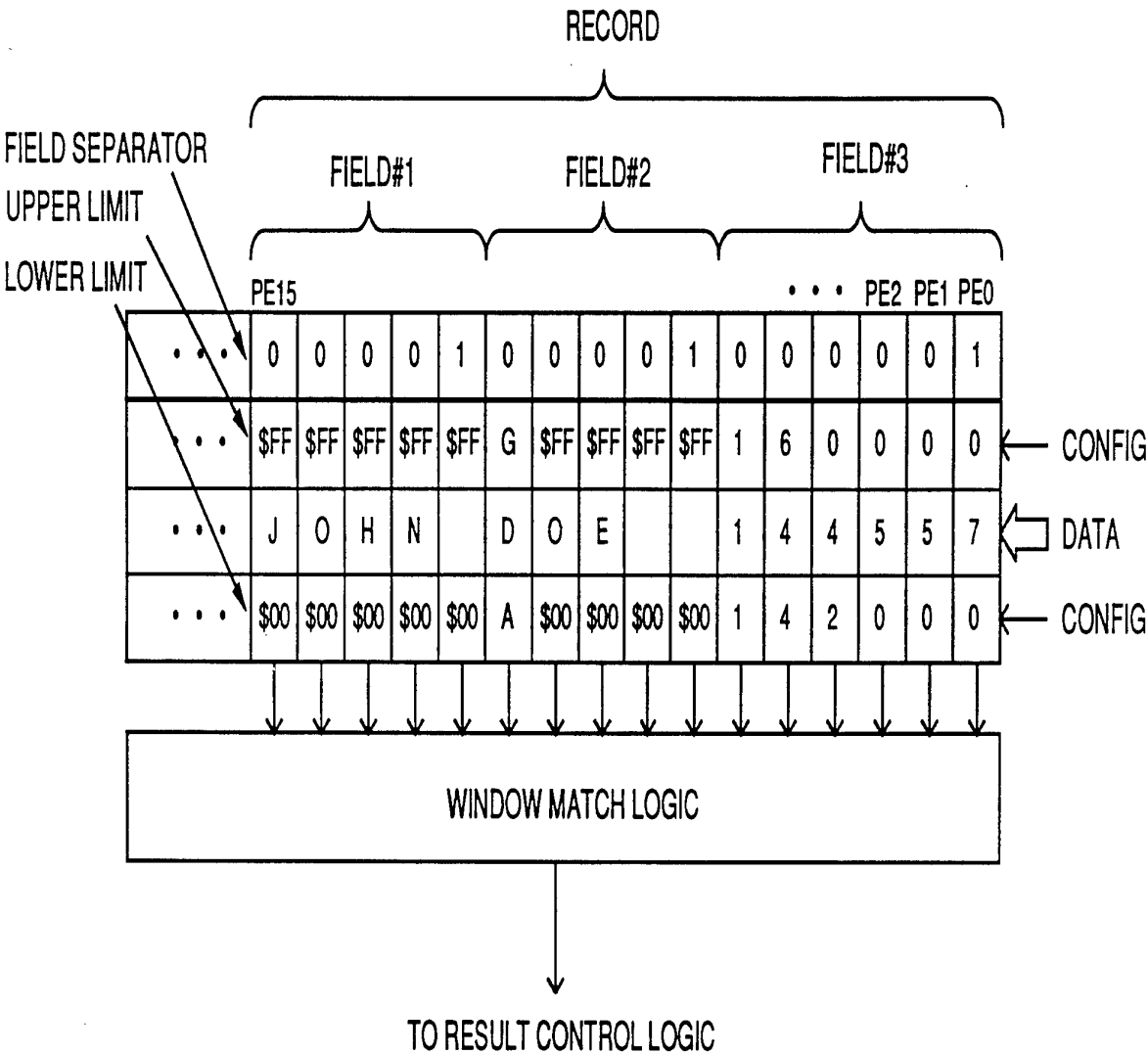


Fig. 20

INTERNATIONAL SEARCH REPORT

International application No.

PCT/NO 92/00173

A. CLASSIFICATION OF SUBJECT MATTER

IPC5: G06F 15/40, G06F 15/80

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC5: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

DIALOG: 340,347,350,351

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP, A2, 0233401 (TRW INC.), 26 August 1987 (26.08.87), page 6, line 24 - page 7, line 30, figures 1,2 --	1-10
A	US, A, 5060143 (KUO-CHU LEE), 22 October 1991 (22.10.91), column 7, line 27 - column 8, line 54, figures 2,3 --	1-10
A	EP, A1, 0428328 (AMT (HOLDINGS) LIMITED), 22 May 1991 (22.05.91), abstract -- -----	1-10

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

17 May 1993

Date of mailing of the international search report

19 -05- 1993

Name and mailing address of the ISA/

Swedish Patent Office

Box 5055, S-102 42 STOCKHOLM

Facsimile No. +46 8 666 02 86

Authorized officer

Katarina Fredriksson

Telephone No. +46 8 782 25 00

INTERNATIONAL SEARCH REPORT
Information on patent family members

31/03/93

International application No.
PCT/NO 92/00173

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
EP-A2-	0233401	26/08/87	JP-A- US-A-	62217321 5051947	24/09/87 24/09/91

US-A-	5060143	22/10/91	NONE		

EP-A1-	0428328	22/05/91	JP-A-	3211688	17/09/91
