PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 3 :		(11) International Publication Number:	WO 83/ 02016
G06F 1/00	A1	(43) International Publication Date:	9 June 1983 (09.06.83)

(21) International Application Number: PCT/US82/01694

(22) International Filing Date: 1 December 1982 (01.12.82)

(31) Priority Application Number: 327,626

(32) Priority Date: 4 December 1981 (04.12.81)

(33) Priority Country: US

(71) Applicant: BURROUGHS CORPORATION [US/US]; Burroughs Place, Detroit, MI 48232 (US).

(72) Inventor: FLORA, Laurence, Paul ; 1381 North Sunflower Street, Covina, CA 91724 (US).

(74) Agents: CHUNG, Edmund, M.; Burroughs Corporation, Burroughs Place, 4B22, Detroit, MI 48232 (US) et al.

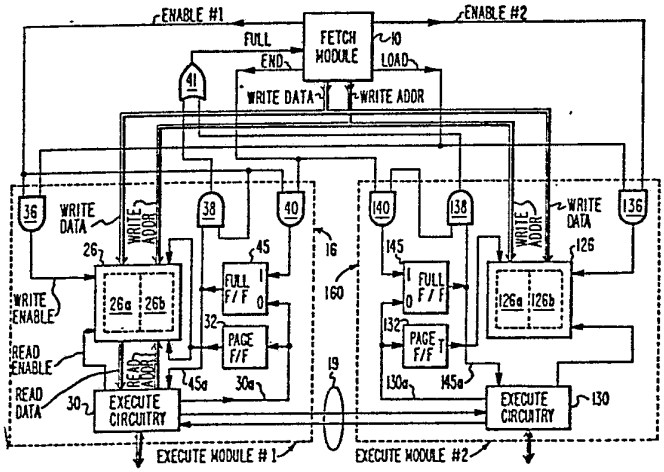
(81) Designated State: JP.

Published
With international search report.

(54) Title: DATA PROCESSING SYSTEM PROVIDING IMPROVED DATA TRANSFER BETWEEN MODULES

(57) Abstract

In a modular type of data processor, it is advantageous that being processed be asynchronously transferred between modules with minimum required complexity and without detracting from overall processor performance. Instead of using the conventional approach of employing input and output queues or direct wiring between modules which result in the above disadvantages, the disclosed invention provides one or more modules, such as an execute module (16), with a randomly accessible scratchpad memory (26) which is logically divided into two switchable pages (26a) and (26b). During operation, one page can be written with new instruction data from a different module, such as a fetch module (10), while a previously written page is concurrently being read by the execute module for execution of a designated data processing operation. When the execute module completes execution and is ready for a new block of data, the two pages are rapidly switched by toggling a bit of the page address using a page flip-flop (32).



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	**LI**	Liechtenstein
AU	Australia	**LK**	Sri Lanka
BE	Belgium	**LU**	Luxembourg
BR	Brazil	**MC**	Monaco
CF	Central African Republic	**MG**	Madagascar
CG	Congo	**MR**	Mauritania
CH	Switzerland	**MW**	Malawi
CM	Cameroon	**NL**	Netherlands
DE	Germany, Federal Republic of	**NO**	Norway
DK	Denmark	**RO**	Romania
FI	Finland	**SE**	Sweden
FR	France	**SN**	Senegal
GA	Gabon	**SU**	Soviet Union
GB	United Kingdom	**TD**	Chad
HU	Hungary	**TG**	Togo
JP	Japan	**US**	United States of America
KP	Democratic People's Republic of Korea		

DATA PROCESSING SYSTEM PROVIDING IMPROVED DATA TRANSFER BETWEEN MODULES

BACKGROUND OF THE INVENTION

This invention relates generally to data processing apparatus and, more particularly, to improved apparatus and methods for enhancing the performance of data processing operations.

In many types of data processing systems, each task is divided into several smaller pieces to permit concurrent performance of the task by several smaller data processors or modules. Although this approach permits each of these smaller data processing modules to be optimized for performance of its particular portion of the task, the transferring of the partially completed task between modules can involve considerable overhead.

For example, a typical way of implementing data transfers between modules in a system such as described above is to provide input and output scratchpad queues for each module so as to permit data to be asynchronously transferred between modules. In such an implementation, a sending module places data to be transferred in its output queue and then

forgets about it. The output queue asynchronously transfers the data to the receiving module's input queue when it is available to receive the data. Then, when the receiving module requires the transferred data, it simply reads the data from its input queue. Although the use of input and output queues in this manner solves the problem of having to synchronize module operations, it has the disadvantage that the queue structure is expensive and also involves considerable overhead.

Of course, a more direct way of implementing data transfers between modules is to simply provide sufficient wiring between modules so that data can be directly transferred therebetween with little or no overhead penalty. However, such an approach not only requires a large and expensive interconnection structure, but also requires that the data be in registers rather than in more cost-effective scratchpads.

SUMMARY OF THE INVENTION

Accordingly, it is a broad object of the present invention to provide improved apparatus and methods for transferring data between units in a data processing system.

A more specific object of the invention is to provide economical apparatus and methods for transferring data between data processing modules in a data processing system in a manner which significantly reduces the transfer overhead.

Another object of the invention is to provide for transferring data between units in a data processing system using a relatively simple and economical interconnection arrangement.

An additional object of the invention is to provide for transferring data between units in a data processing

system wherein the data to be transferred resides entirely in scratchpad memories.

The above objects are accomplished in a preferred embodiment of a data processing system in accordance with the invention by providing a receiving module with a scratchpad memory which is logically divided into two switchable pages. During operation, one page can be written by a sending module while the other page is being read by the receiving module. Each page is randomly addressable by the corresponding module so that the sending module need not send the data in any particular order, and the receiving module need not spend any time in transferring the data from one place to another. When the receiving module is finished with a page it is reading and requires a new block of data available in the other page (as a result of being written by a sending module), the two pages are logically switched. This is accomplished without any transfer of data. Instead, all that is required is that an address bit be toggled so that the pages addressed by the sending and receiving modules are effectively switched. Thus, in essentially zero time (the time required to toggle the address bit which is extremely fast), an entirely new block of data is caused to appear in the receiving module's addressable page. The transfer overhead is thereby substantially eliminated without the need of any expensive queue structure, since the data may reside entirely in economical scratchpad memory chips.

The specific nature of the invention as well as other objects, uses, features and advantages thereof will become evident from the following description of a preferred embodiment taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is an overall block diagram illustrating a plurality of cooperating data processing modules which may advantageously incorporate the present invention.

FIG. 2 is a block diagram of a preferred implementation illustrating the cooperation provided between the fetch and execute modules of FIG. 1 in accordance with the invention.

FIG. 3 is a series of timing diagrams illustrating the timing relationships between the various signals transferred between the fetch and execute modules in FIG. 2.

DETAILED DESCRIPTION OF THE INVENTION

Like numerals and characters designate like elements throughout the figures of the drawings.

Referring initially to FIG. 1, illustrated therein is a block diagram of a plurality of cooperating data processing modules which may advantageously incorporate the present invention. As indicated in FIG. 1, a fetch module 10 cooperates with a read/write memory controller module 12 to access instructions from a memory module 14 via bus 13 for application to one or more execute modules (such as modules 16 and 116) via bus 15. Signals required between the execute modules 16 and 116 are applied via lines 19. Data involved in the data processing operations to be performed is transferred between the memory controller 12 and the respective fetch and execute modules 10, 16 and 116 also via bus 13.

More specifically, the fetch module 10 is specially designed to resolve the field lengths and addresses of the retrieved instructions prior to sending to the execute modules 16 and 116. Each of the execute modules 16 and 116 is specially designed to perform the data processing called for by the applied instructions, and may be configured in either a single or dual configuration. The memory controller module 12 interfaces the fetch module 10 and the execute modules 16 and 116 to the memory module 14.

The present invention will be illustrated as applied to the fetch module 10 and the execute modules 16 and 116 in FIG. 1. For this purpose, attention is now directed to FIG. 2 which is a preferred implementation illustrating the cooperation provided between the fetch module 10 and the execute modules 16 and 116 in accordance with the invention.

As illustrated in FIG. 2, each of the execute modules 16 and 116 includes a scratchpad random access

memory 26 or 126, respectively, each memory 26 and 126 being functionally divided into two pages 26a and 26b and 126a and 126b respectively. As will hereinafter be described in more detail, these two pages in each memory are switchable by their respective execute circuitry 30 or 130 after every instruction so that each page may alternately function as a fetch page to receive instruction data from the fetch module 10 and as an execute page for use by its respective execute circuitry in performing an instruction. Thus, while the execute circuitry 30 or 130 in an execute module is executing an instruction from one page, the fetch module 10 can fill up the other page with the data required for a succeeding instruction.

Furthermore, in the preferred embodiment, the memories 26 and 126 are chosen to be of the random access type and each location in the memory is designated to store a particular portion of the applied instruction data. Thus, the fetch module 10 can send its data to an execute module in any order for writing at appropriate designated addresses in the current fetch page; also, the execute circuitry after switching a fetch page for use as an execute page, can selectively read out the data from the appropriate designated addresses as required for executing the indicated instruction.

Before continuing with the description of FIG. 2, it should be noted that, in order to simplify the description of FIG. 2, references will hereinafter be made primarily to the construction and operation of execute module #1 identified by the numerical 16, which, for the purposes of this description, may be assumed to be identical to execute module #2 identified by the 100-greater numerical designation 116. For ease of comparison, components of execute module #2 are given 100-greater numerical designations than like components in execute module #1. Also, in order to simplify

FIG. 2, clock inputs to the various components (which are conventional) are not shown, and should be assumed as provided where required.

Returning to the description of FIG. 2, a page flip-flop 32 is provided whose state is controlled by the execute circuitry 30, and whose output provides one bit of the write address or read address applied to the random access memory 26. The organization of the memory 26 is such that, when the page flip-flop 32 is in one state, an applied write address from the fetch module 10 causes accompanying write data to be written at the designated address in a first page of the memory 26, while concurrently permitting an applied read address from the execute circuitry 30 to read out data from the designated address in the other page of the memory. The opposite occurs when the page flip-flop 32 is in its other state -- that is, when the page flip-flop is in its second state, an applied write address causes accompanying write data to be written at the designated address in the second page of the memory 26, while an applied read address causes data to be read out from the designated address in the first page of the memory 26. It will thus be evident that the two pages provided in each execute module memory 26 or 126 can be switched merely by switching the respective page flip-flop 32 or 132, which can be done very rapidly (for example, in a few nanoseconds).

The construction and operation of the remaining portions of the execute module #1 will become evident from the following consideration of the signals applied between the fetch module 10 and execute module #1 during typical operations. For this purpose, attention is additionally directed to the timing diagram of FIG. 3 which illustrates the timing relationships between the various signals

transferred between the fetch and execute modules during typical operation.

Typically, the fetch module 10 sends instructions alternately between execute modules #1 and #2 in order to speed up overall processing time. The required sequencing of instructions is enforced by signals sent via lines 19 between the execute circuitry 30 and 130 of the execute #1 and #2 modules. These signals act to prevent an execute module from changing machine state until the other module has completed execution of a previous instruction. The specific manner in which this is accomplished is not pertinent to the present invention and will thus not be further described; however, it is to be noted that such instruction sequencing control between modules can readily be provided by those skilled in the art.

For the purposes of the present description of FIG. 2 using the timing diagrams of FIG. 3, it will be assumed that, during the first clock indicated in the top graph in FIG. 3, the ENABLE #1 signal illustrated in the second graph in FIG. 3 becomes "1" (positive) which enables execute module #1 by enabling AND gates 36, 38 and 40. The enabling of AND gate 38 causes the current state of a full flip-flop 45 to be applied via an OR gate 41 to the fetch module 10 as a FULL signal, as illustrated in the third graph in FIG. 3. The state of the full flip-flop 45 and thus the enabled FULL signal indicates whether the enabled execute module #1 is ready to receive new instruction data -- that is, whether the execute circuitry 30 has completed execution of its previous instruction and switched to the other page. The execute circuitry 30 manifests this instruction completion by applying, via output line 30a, a "1" signal to the page flip-flop 32 to switch the page in memory 26, while also applying this "1" signal to the full flip-flop 45 to switch

its output to a "0" to indicate that the current fetch page is now available to receive new instruction data.

It will also be assumed that, when the fetch module 10 sends the ENABLE #1 signal to the execute module #1, the previous instruction has been completed so that the execute circuitry 30 will have switched to the next page and will have reset the full flip-flop 45 to the "0" state to indicate that the execute module #1 is ready to receive new instruction data. Thus, when the applied ENABLE #1 signal from the fetch module 10 enables AND gate 38 (along with AND gates 36 and 40), a "0" FULL signal is received by the fetch module 10, which is indicated in FIG. 3 by the FULL signal going negative. This "0" FULL signal permits the fetch module 10 to load a data word into execute module #1. This is accomplished by the fetch module 10 sending out to execute module #1 a LOAD signal along with a DATA word and an accompanying WRITE ADDRESS. The LOAD signal is applied, via AND gate 36 (which remains enabled by the ENABLE #1 signal) for use as a write enable signal for writing the transmitted DATA word into the memory 26 at the accompanying WRITE ADDRESS in the page designated by the output of the page flip-flop 32. As indicated in FIG. 3, the fetch module 10 can send a new DATA word and a different accompanying WRITE ADDRESS at each succeeding clock which will be written in memory 26 in a like manner.

When the fetch module 10 sends its last DATA word and accompanying WRITE ADDRESS, it also sends a "1" END signal to the execute #1 module. This "1" END signal passes, via AND gate 40 (which remains enabled by the ENABLE #1 signal) to the "1" input of the full flip-flop 45 to switch it to a "1" state to thereby indicate that the current page is full. At the next clock following the END signal (which is the last clock shown in FIG. 3), the ENABLE #1 and LOAD signals

return to "0", while the ENABLE #2 signal becomes "1"; the FULL signal applied to the fetch module 10 thus now represents the status of the FULL flip-flop 145 on execute module #2. As pointed out previously, the full flip-flop 45 or 145 remains in a "1" state until its respective execute module #1 or #2 completes its previous instruction and switches the pages.

So far, it has been described how the fetch module 10 sends data words to specified addresses of the current fetch page, that is, the page designated by the page flip-flop 32 for storing data sent by the fetch module. While the fetch module 10 is writing into the current fetch page, the execute circuitry 30 is able to concurrently and selectively read out instruction data words stored in the other (or execute) page designated by the page flip-flop 32, as required in order to execute the indicated instruction. This is readily accomplished since the execute circuitry 30 need merely provide a read enable signal and READ ADDRESS to the memory 26 for each data word to be read out. Each applied READ ADDRESS in conjunction with the output from the page flip-flop 32 will then read out the selected READ DATA word from the designated address in the execute page. When the instruction is completed and the full flip-flop is "1", the execute circuitry 30 switches the state of the page flip-flop 32 to switch the pages and also resets the full flip-flop 45 to "0" to permit the next instruction data to be sent from the fetch module 10. Also, when the execute circuitry 30 completes an instruction, it sends a signal via one of lines 19 to the execute circuitry 130 of the execute module #2 for use in enforcing the proper sequencing of instructions.

Although this description has been directed to a particular illustrative embodiment of the invention, it is to

be understood that the invention is subject to a wide variety of possible modifications and variations in construction, arrangement and use without departing from the true scope and spirit of the invention. For example, it is to be understood that the present invention is also applicable for use in a system which employs multiple execute modules, wherein each module performs only a part of an overall job and then passes the job on to the next module. Accordingly, the present invention is to be considered as encompassing all possible apparatus and methods coming within the scope of the invention defined by the appended claims.

What is claimed is:

1. In a data processing system, the combination comprising:

first and second portions;

said second portion including a memory having a plurality of switchable pages;

said second portion also including means for reading data from one of said pages for use in the performance of data processing operations while concurrently permitting data required for use in subsequent data processing operations to be written into a different one of said pages;

said second portion further including means for switching said pages in a manner such that, after said second portion has completed data processing with respect to data in one page, the pages are switched so that another page can be read by said second portion while said one page becomes available to have new data written therein; and

transferring means for writing data provided by said first portion into a page of said second portion which is available to have data written therein.

2. The invention in accordance with claim 1, wherein said memory is selectively addressable, wherein the composite address applied to said memory is comprised of either a write address provided by said first portion or a read address provided by said second portion along with a predetermined number of bits provided by said second portion which selects the particular page to which an applied write address or read address is to be applied.

3. The invention in accordance with claim 2, wherein said second portion provides for switching said pages by changing the value of said predetermined number of bits.

4. The invention in accordance with claim 3, wherein said memory is randomly accessible so as to permit each of said first and second portions to randomly address a selected location in the page selected by said predetermined number of bits.

5. The invention in accordance with claim 3, wherein said second portion includes means for indicating which page is available to have new data written therein.

6. The invention in accordance with claim 5, wherein said second portion includes means for indicating when the writing of new data in a page has been completed.

7. The invention in accordance with claim 2, 3, 4, 5 or 6, wherein said memory comprises two pages, wherein said predetermined number of bits is a single bit, and wherein said second portion provides for switching said pages by changing the value of said single bit.

8. The invention in accordance with claim 2, 3, 4, 5 or 6, wherein said first and second portions respectively comprise first and second modules each performing a predetermined function.

9. The invention in accordance with claim 2, 3, 4, 5 or 6, wherein said memory is a scratchpad memory.

10. The invention in accordance with claim 8, wherein said second module is an execute module for performing data processing operations in response to instruction data read from a page of said memory.

11. The invention in accordance with claim 10, wherein said first module is a fetch module which provides instruction data to an available page in said memory via said transferring means.

12. The invention in accordance with claim 11, including a second execute module of like construction as said first execute module for alternately receiving instruction data from said fetch module via said transferring means.

13. An improved method for operating a data processing system, said method comprising:

providing a memory having a plurality of switchable pages arranged so that one page is available for use in the performance of a data processing operation while another page is concurrently available to receive data for a subsequent data processing operation;

performing a data processing operation by reading data from a page which is available for such use while concurrently writing new data in a page which is available for receiving data; and

switching said pages after each completion of data processing with respect to a page available for such use so that another page which has received data is now available for use in a data processing operation while the page with respect to which data processing has been completed becomes available to receive new data.

14. The invention in accordance with claim 13, wherein data is written in and read from said memory using a composite address comprised of a read address or a write address along with a predetermined number of bits which determines the particular page to which an applied write address or read address is to be applied, and wherein the step of switching is performed by changing the value of said predetermined number of bits.

15. The invention in accordance with claim 14, wherein said reading from and writing in said memory is performed in a random manner.

16. The invention in accordance with claim 13, 14 or 15, wherein said memory comprises two pages, wherein said predetermined number of bits is a single bit, and wherein the step of switching is performed by changing the value of said single bit.

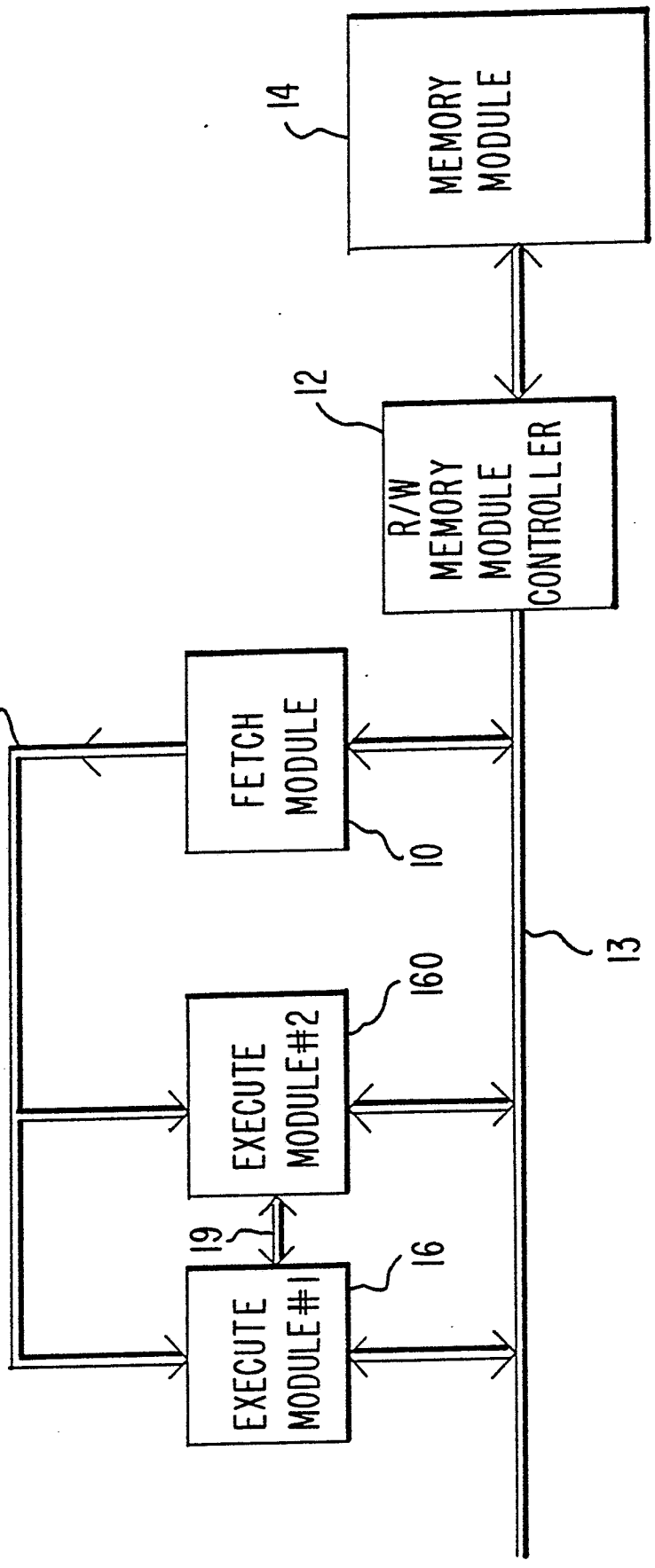


FIG.1.

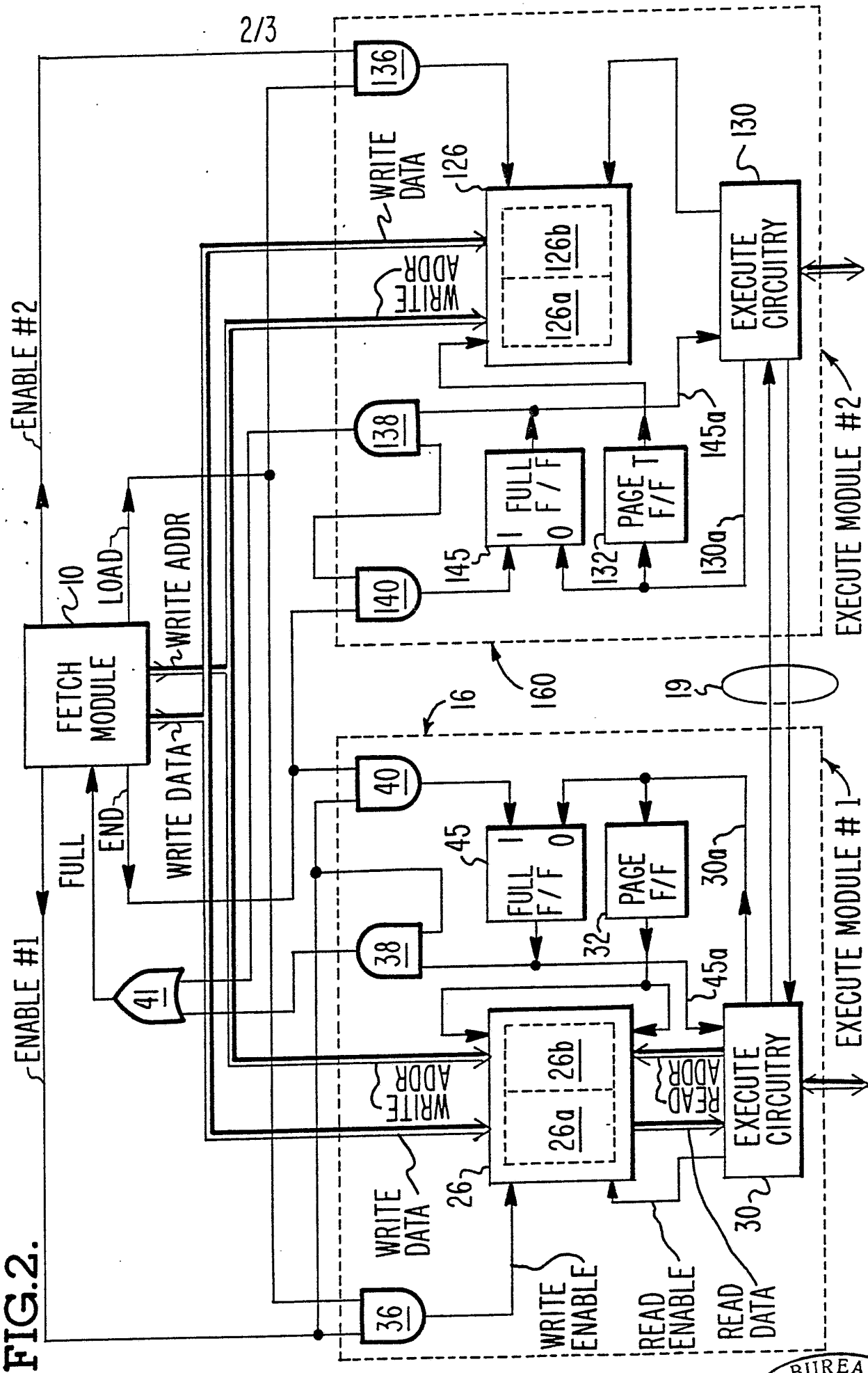


FIG.2.

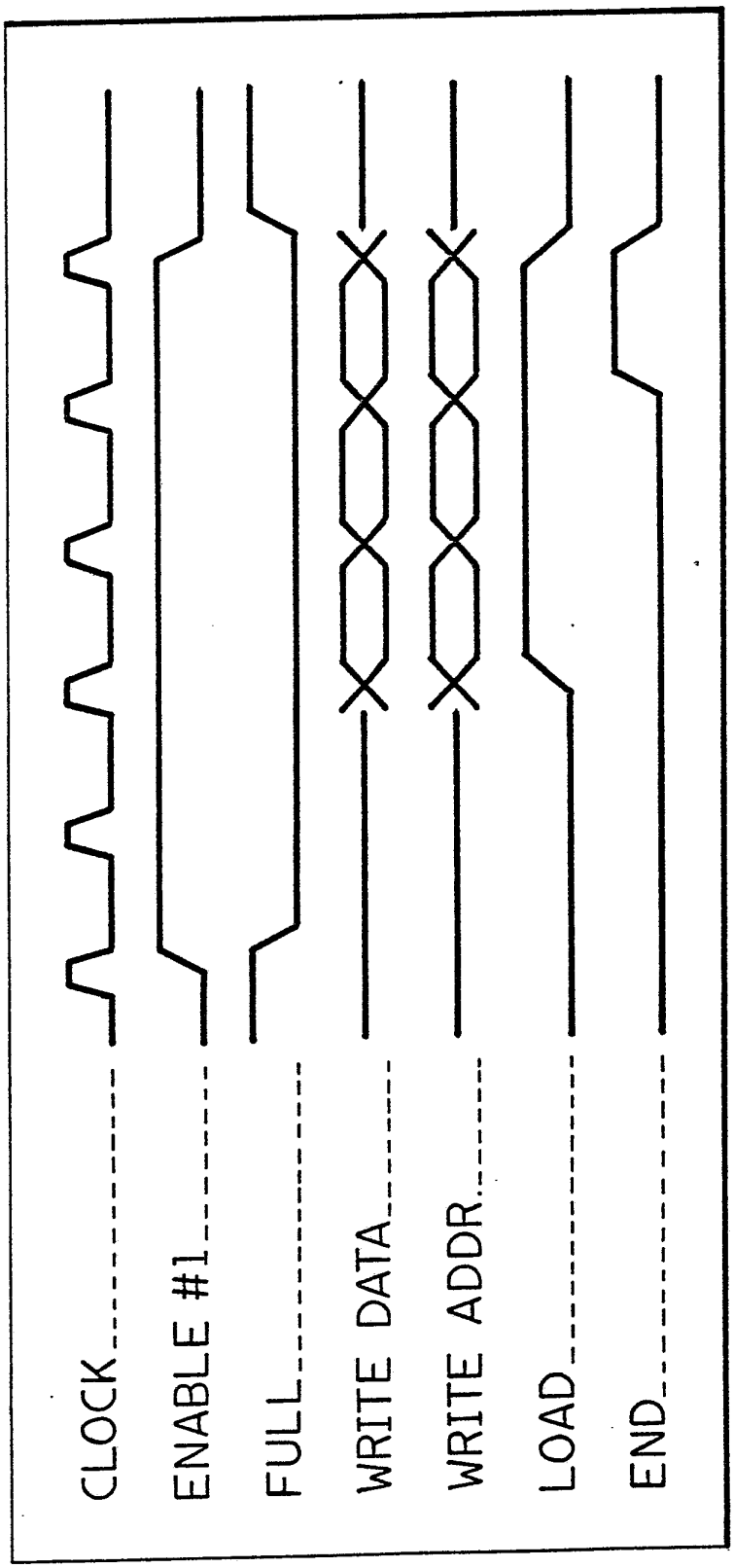


FIG. 3.

INTERNATIONAL SEARCH REPORT

International Application No PCT/US82/01694

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [3]

According to International Patent Classification (IPC) or to both National Classification and IPC

INT. CL.[3] G06F 1/00 U.S. CL. 364/200

II. FIELDS SEARCHED

Minimum Documentation Searched [4]

Classification System	Classification Symbols
U.S.	364/200

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [5]

III. DOCUMENTS CONSIDERED TO BE RELEVANT [14]

Category *	Citation of Document, [16] with indication, where appropriate, of the relevant passages [17]	Relevant to Claim No. [18]
X	US,A 4,173,783 (Couleur et al) 06 November 1979	1-16
X	US,A 4,124,891 (Weller, et al) 07 November 1978	1-16
X	US,A 4,177,510 (Appell, et al) 04 December 1979	1-16
X	US,A 4,136,386 (Annuniziata et al) 23 January 1979	1-16

* Special categories of cited documents: [15]

"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

"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

IV. CERTIFICATION

Date of the Actual Completion of the International Search [2]	Date of Mailing of this International Search Report [2]
17 FEBURARY 1983	03 MAR 1983
International Searching Authority [1]	Signature of Authorized Officer [20]
ISA/US	

Form PCT/ISA/210 (second sheet) (October 1981)