



(11) (21) (C) **2,081,913**
(86) 1991/06/05
(87) 1991/12/08
(45) 2000/05/30

(72) Ryan, Robert P., US

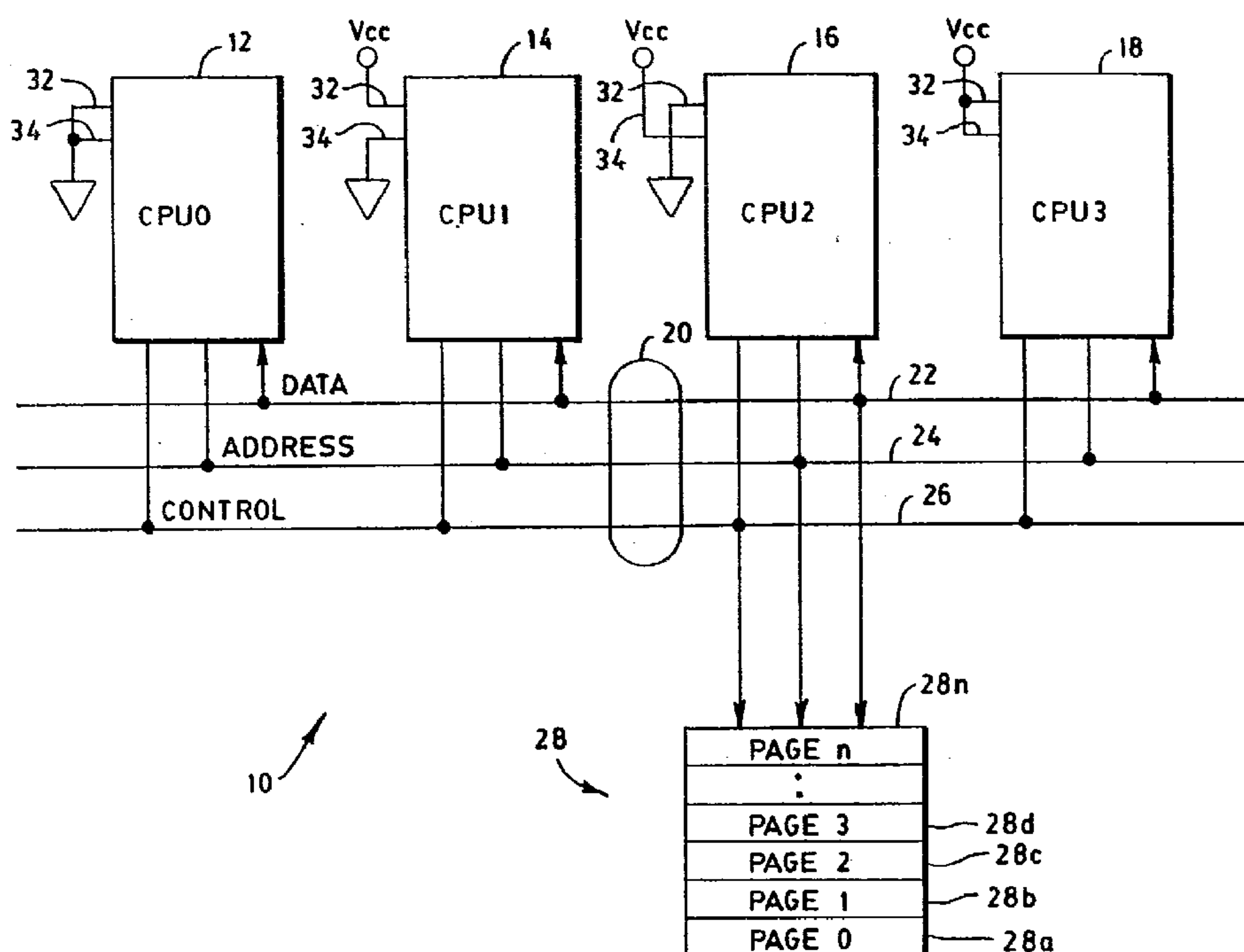
(73) LG SEMICON CO., LTD., KR

(51) Int.Cl.⁵ G06F 12/02, G06F 15/16

(30) 1990/06/07 (534,392) US

(54) **METHODE ET DISPOSITIF D'ACCES A LA PAGE ZERO DE LA MEMOIRE D'UN SYSTEME MULTIPROCESSEUR**

(54) **METHOD AND APPARATUS FOR MANAGING PAGE ZERO MEMORY ACCESSSES IN A MULTI-PROCESSOR SYSTEM**



(57) Appareil et procédé destinés à être utilisés dans un système de traitement de données multi-UC (10) dans lequel chaque UC (12 à 18) est couplée à un bus commun (20) et par l'intermédiaire de ce dernier à une mémoire centrale (28). Le procédé et l'appareil comportent un programme, tel qu'un système d'exploitation, fonctionnant sur chacun des accès des UC à une page de données se trouvant dans la mémoire centrale. Chacune des UC produit une première adresse destinée à identifier un emplacement ou des emplacements en mémoire à l'intérieur d'une page de la mémoire centrale. La première adresse est modifiée en fonction de la première

(57) Apparatus and method for use in a multi-CPU data processing system (10) wherein each CPU (12-18) is coupled to a common bus (20) and through the common bus to a main memory (28). The method and apparatus provides a program, such as an operating system, that is operating upon each of the CPUs access to a page of data within the main memory. Each of the CPUs generates a first address for identifying a memory location or locations within a page of the main memory. The first address is modified as a function of the first address and as a function of an identification of the CPU to generate a second address for identifying a memory location or



(11) (21) (C) **2,081,913**
(86) 1991/06/05
(87) 1991/12/08
(45) 2000/05/30

adresse et en fonction d'une identification de l'UC afin de produire une seconde adresse destinée à identifier un emplacement ou des emplacements en mémoire se trouvant soit dans la même page soit dans une autre page de la mémoire centrale. L'adresse modifiée est appliquée à la mémoire afin d'accéder à un ou à plusieurs emplacements en mémoire.

locations that are either within the same page or within another page of the main memory. The modified address is applied to the memory for accessing the memory location or locations.



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :

G06F 12/02, 15/16

2081913

A1

(11) International Publication Number:

WO 91/19254

(43) International Publication Date:

12 December 1991 (12.12.91)

(21) International Application Number: PCT/US91/03479

(22) International Filing Date: 5 June 1991 (05.06.91)

(30) Priority data:

534,392

7 June 1990 (07.06.90)

US

(71) Applicant: WANG LABORATORIES, INC. [US/US];
One Industrial Avenue, Lowell, MA 01851 (US).

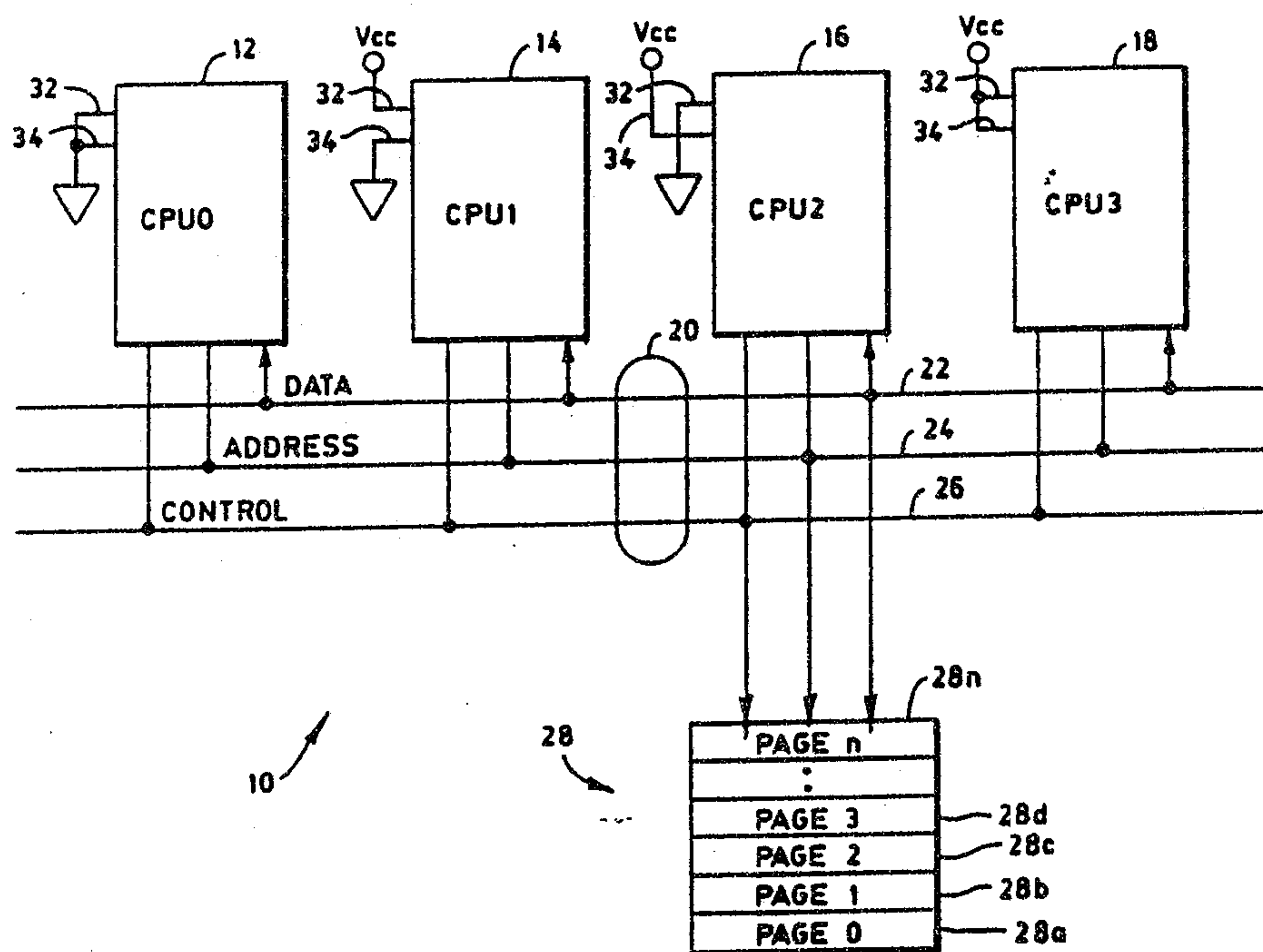
(72) Inventor: ROBERT, P., Ryan ; 111 Houde Street, Marlboro, MA 01752 (US).

(74) Agents: SHANAHAN, Michael, H. et al.; One Industrial Avenue, Lowell, MA 01851 (US).

(81) Designated States: AT (European patent), AU, BE (European patent), CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent).

Published*With international search report.**Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.*

(54) Title: METHOD AND APPARATUS FOR MANAGING PAGE ZERO MEMORY ACCESSES IN A MULTI-PROCESSOR SYSTEM



(57) Abstract

Apparatus and method for use in a multi-CPU data processing system (10) wherein each CPU (12-18) is coupled to a common bus (20) and through the common bus to a main memory (28). The method and apparatus provides a program, such as an operating system, that is operating upon each of the CPUs access to a page of data within the main memory. Each of the CPUs generates a first address for identifying a memory location or locations within a page of the main memory. The first address is modified as a function of the first address and as a function of an identification of the CPU to generate a second address for identifying a memory location or locations that are either within the same page or within another page of the main memory. The modified address is applied to the memory for accessing the memory location or locations.

2081913

1

METHOD AND APPARATUS FOR MANAGING PAGE ZERO MEMORY
ACCESSES IN A MULTI-PROCESSOR SYSTEM

FIELD OF THE INVENTION:

5

This invention relates generally to digital data processing systems and, in particular, to method and apparatus for managing accesses to a system memory by a plurality of data processors concurrently executing programs each requiring a predefined, reserved memory area.

10

BACKGROUND OF THE INVENTION:

15

20

25

Most, if not all, operating system programs for use with digital data processors require a reserved area of system read/write memory for the storage of operating system-related data such as device tables, interrupt vectors and so on. In many systems this reserved area of memory is predefined to be a "lowest" page of physical memory, often referred to as "page zero". Page zero will extend across a physical address range of zero to some larger address such as 1023 or 2047. The use of page zero for such operating system functions is advantageous if for no other reason than many data processors have a special form of addressing for page zero that includes a shorter instruction length resulting in a faster instruction fetch and execution.

30

A problem is created for those types of systems characterized as a multi-processor system wherein a



2681013

plurality a data processors, each of which executes an associated controlling program such as an operating system, are coupled to a common system memory. Each of the data processors requires access to an associated page zero within the common memory. However, in that each of the data processors may be operating independently of the other data processors the processors cannot be allowed to "share" the page zero memory area. For example, if one of the data processors were to modify the page zero data the modified data may be inappropriate for use by the other data processors.

It is therefore an object of the invention to provide apparatus and method for use in a multi-processor data processing system for enabling each of the processors to access an associated predefined region of system memory.

It is another object of the invention to provide apparatus and method for use in a multi-processor data processing system for providing each of the processors an associated "page zero" within system read/write memory to enable concurrent execution of programs without conflict by the data processors.

It is a further object of the invention to provide a data processing system having n data processors each running a program requiring a predetermined region of system read/write memory and to also provide n regions of system memory each of which is mapped to appear, to an associated processor, as the predetermined region of system memory.

2081913

3

It is a still further object of the invention to provide a data processing system having n data processors each running a program requiring a page zero region of system read/write memory and to also provide n effective page zero regions of memory to enable concurrent program execution by the data processors.

SUMMARY OF THE INVENTION

10 The foregoing problems are overcome and the objects of the invention are realized by apparatus and method for use in a data processing system of the type that includes n data processors where n is greater than one. Each of the data processors executes instructions including
15 instructions that generate a first address for accessing a predetermined region of system memory. The apparatus includes, for each of the data processors, circuitry for modifying the first address to a second address. The circuitry includes a first input for receiving the first
20 address, a second input for receiving an identification of the data processor, and an output for providing the second address therefrom. The second address is a function of the first and the second inputs.

25 In a presently preferred embodiment of the invention n equals from one to four. The identification (ID) of each of the data processors is expressed in binary as 00, 01, 10 or 11 and the predetermined region of system memory is a memory PAGE0. The circuitry modifies the first address
30 for accessing one of n memory pages (PAGE0, PAGE1, PAGE2 or PAGE3) in accordance with the following table:

2681013

FIRST				
ADDRESS	ID=00	ID=01	ID=10	ID=11
PAGE0	PAGE 0	PAGE 1	PAGE 2	PAGE 3
PAGE1	PAGE 1	PAGE 0	PAGE 1	PAGE 1
5 PAGE2	PAGE 2	PAGE 2	PAGE 0	PAGE 2
PAGE3	PAGE 3	PAGE 3	PAGE 3	PAGE 0.

10 A data processor other than the data processor having the ID=00, when generating a first address that targets physical PAGE0, has the address modified such that another page of memory is accessed instead. As a result, each of the system data processors has an associated PAGE0 region of system memory. Thus each of the data processors concurrently executes an operating system that

15 requires exclusive use of PAGE0 without requiring modification of the operating system. Furthermore a data processor is enabled to read, if desired, PAGE0 information relating to another of the operating systems executing on another of the data processors.

20

BRIEF DESCRIPTION OF THE DRAWING

25 The above set forth and other features of the invention are made more apparent in the ensuing Detailed Description of the Invention when read in conjunction with the attached Drawing, wherein:

30 Fig. 1 is a block diagram illustrating a multi-processor data processing system constructed and operated in accordance with the invention;

2081913

5

Fig. 2 is block diagram illustrating a page zero access decoder having an address output for coupling to a system memory; and

5 Fig. 3 is a flow chart illustrating a method of the invention.

DETAILED DESCRIPTION OF THE INVENTION

10 The following description is made in the context of a multi-processor system wherein each of the processors executes an operating system known as the VS operating system provided by Wang Laboratories, Incorporated of Lowell, Massachusetts. However, it should be realized
15 that the use of the invention is applicable to a wide variety of data processing system employing other operating systems. For example, the use of the invention is also applicable to a multi-processor system employing a plurality of embedded data processors having
20 controlling programs executed from ROM, the data processors being coupled to a common memory. As such, the practice of the invention is not intended to be limited solely to the presently preferred embodiment disclosed below.

25

Referring now to Fig. 1 there is shown a portion of a multi-processor data processing system 10. System 10 includes a plurality of bus agents such as data processors CPU0 12, CPU1 14, CPU2 16 and CPU3 18. Each
30 of the CPUs 12-18 is of substantially identical construction and each executes a substantially identical version of the VS operating system. Each of the CPUs includes instruction fetch and decode circuitry, address

2021013

generation circuitry, arithmetic/logic circuitry and other circuits for performing operations as directed by stored instructions. Each of the CPUs 12-18 is coupled to a system bus 20. The system bus 20 includes a plurality of data signal lines 22, address signal lines 24 and control lines 26. By example, the data signal lines 22 include 64 discrete signal lines and associated parity signal lines. Address signal lines 24 include 28 address lines. The control signal lines 26 include memory read and write strobes, system clocks, system bus arbitration signal lines and other known types of signals. During operation each of the CPUs operates to generate physical addresses for application to the system bus 20, the physical addresses being generated by a virtual address to physical address translation unit (not shown) that forms a portion of each of the CPUs.

Each of the CPUs 12-18 includes a mechanism for defining a CPU identification (ID) value, the ID value being a function of where the CPU is physically installed upon a common system backplane or motherboard. For the four CPU system shown two input signal lines are employed for this identification function and are defined as CPUID0 32 and CPUID1 34. These two lines are electrically coupled to logic signals hardwired on the backplane such that when a CPU is physically and electrically installed within a mating connector or connectors provided upon the backplane the ID value is provided to the CPU. These logic signals are shown schematically in Fig. 1 as being circuit ground and circuit power (Vcc) for expressing a logic zero and a logic one, respectively. CPUID0 32 and CPUID1 34 thus together express a two digit binary value.

2081913

7

Depending upon the implementation this value could be expressed in two's complement or in any suitable format.

5 Also coupled to system bus 20 is a system read/write memory 28 that is comprised of a plurality of pages (PAGE0 28a-PAGEn 28n) of memory storage locations. In the VS operating system each page is defined to be 2048 bytes (2Kbytes) in length. Thus, PAGE0 28a occupies an addressable region of memory from address 0-2047, PAGE1 10 28b occupies a region from address 2048-4095, etc.

The VS operating system requires physical PAGE0 28a (addresses 0-2047) as storage for operating system data structures and other operating system-related data. 15 However, it can readily be seen that if each of the CPUs 12-18 is executing a separate VS operating system only one of the CPUs 12-18 is enabled to actually be coupled to the physical PAGE0 of the system memory 28. Without the benefit conferred by the use of the invention it 20 would be required to modify the operating systems of the other CPUs to use another region of memory instead of PAGE0 28a or to make some other type of modification. However, it is usually desirable that any such modifications to standard software systems be avoided. 25 Furthermore, other problems may result if additional CPUs were subsequently added to the system 10 to increase the capability of the system.

Referring now to Figs. 2 and 3 there is shown, in 30 accordance with the invention, a PAGE0 decoder 30 that forms a portion of each of the CPUs 12-18 and a method of employing same, respectively. The particular decoder 30 shown in Fig. 2 forms a part of the CPU0 12. The decoder

2081913

8

30 may be implemented as a group of discrete logic devices, as a pre-programmed read only memory, as a programmable logic array (PLA) or in a number of other suitable hardware embodiments. In the presently preferred embodiment of the invention the decoder 30 is implemented as a PLA. Decoder 30 includes a plurality of CPU local address (LADDR) 24a signal line inputs and also the aforementioned CPUID signal lines, specifically CPUID0 32 and CPUID1 34. Outputs of decoder 30 include the plurality of system bus address lines 24. The LADDR 24a signal lines are those provided as outputs of the virtual address to physical address translation unit (not shown) that translates virtual memory addresses generated by the CPU into the physical ADDRESS 24 signal lines.

In the presently preferred embodiment of the invention the first four pages (8Kbytes) of physical memory 28 are permanently allocated memory locations. The decoder 30 employs the CPUID bits, in conjunction with the LADDR 24a inputs, to map these first four pages of system memory 28. The following TABLE 1 shows a presently preferred mapping scheme for the first four pages of system memory 28 as viewed by each of the CPUs 12-18.

TABLE 1

LADDR	CPUID=00	CPUID=01	CPUID=10	CPUID=11
PAGE0 28a	PAGE 0	PAGE 1	PAGE 2	PAGE 3
PAGE1 28b	PAGE 1	PAGE 0	PAGE 1	PAGE 1
PAGE2 28c	PAGE 2	PAGE 2	PAGE 0	PAGE 2
PAGE3 28d	PAGE 3	PAGE 3	PAGE 3	PAGE 0

2081913

9

As can be seen, the CPU identified with CPUID=00 does not remap the first four pages of system memory 28. That is, the PAGE0 for CPU0 12 is the actual physical PAGE0 28A of memory 28, PAGE1 is physical PAGE1 28b, etc. However, CPU1 14 maps its associated PAGE0 to PAGE1 28b in memory 28 and maps its associated PAGE1 to PAGE0 28A. CPU2 16 maps its PAGE0 to PAGE2 28c in memory 28 and maps its PAGE2 to memory PAGE0 28A. Similarly, CPU3 18 maps its PAGE0 to PAGE3 28D and maps PAGE3 to PAGE0 28a. In addition, each of the CPUs 12-18 is preferably constrained from performing a WRITE access to its associated PAGE1-PAGE3 of system memory 28 in order to avoid inadvertently modifying another CPUs' page zero data. Addresses not associated with the lowermost four pages of system memory 28 pass through the decoder 30 and are not modified.

From the above it is apparent that a CPU other than CPU0, when generating an address that targets physical PAGE0 28a, modifies the address such that another page of memory is instead accessed. As a result, each of the system CPUs 12-18 has an associated PAGE0 region of system memory 28 and each of the VS operating systems can be executed, without modification, independently by the four CPUs. Furthermore a CPU is enabled to read, if desired, PAGE0 information relating to another of the operating systems executing on another of the CPUs.

It should be realized that the mapping scheme shown in TABLE 1 is but one suitable mapping arrangement. For example, TABLE 2 illustrates another suitable mapping arrangement.

2081013

TABLE 2

LADDR	CPUID=00	CPUID=01	CPUID=10	CPUID=11
PAGE0 28a	PAGE 0	PAGE 3	PAGE 2	PAGE 1
PAGE1 28b	PAGE 1	PAGE 0	PAGE 3	PAGE 2
PAGE2 28c	PAGE 2	PAGE 1	PAGE 0	PAGE 3
PAGE3 28d	PAGE 3	PAGE 2	PAGE 1	PAGE 0

Also, it should be realized that the actual memory pages employed may be other than the lowermost four pages and that, for example, accesses to pages zero through three could be mapped to pages four through seven or to pages 20 through 24.

Furthermore, the invention is suitable for use with other than four CPUs and can be employed with an arbitrary even or odd number of CPUs. There is no requirement that each CPU be executing the same operating system or controlling program although in many multi-processor systems such is the case. Thus, while the invention has been particularly shown and described with respect to a presently preferred embodiment thereof, it will be understood by those skilled in the art that changes in form and details may be made therein without departing from the scope and spirit of the invention as expressed in the Claims that follow.

CLAIMS

What is claimed is:

1. Apparatus for use in a data processing system of the type that includes n data processing means where n is greater than one, each of the data processing means executing instructions including instructions that generate a first address for accessing a predetermined region of system memory, said apparatus comprising, for each individual one of the data processing means, means for modifying said first address to a second address, said modifying means having a first input for receiving said first address, a second input for receiving an identification of the individual one of the n data processing means, the identification being expressed by a signal line or signal lines that convey logic signals identifying a location wherein each of the individual ones of the processing means is installed within the data processing system, said modifying means having an output for providing the second address therefrom, the second address being a function of the first and the second inputs.

2. Apparatus as set forth in Claim 1 wherein the signal line or signal lines express a binary number for identifying the location where the data processing means is installed.

3. Apparatus as set forth in Claim 1 wherein the predetermined region of system memory is a

predetermined page of system memory comprising a predetermined number of memory storage locations.

4. Apparatus as set forth in Claim 3 wherein the predetermined page is page zero.

5. Apparatus as set forth in Claim 4 wherein the predetermined number of memory storage locations equals 2048 bytes.

6. Apparatus as set forth in Claim 3 wherein n equals up to four, wherein the identification (ID) of each of the data processing means is expressed as a value, wherein the predetermined page is a MEMORY PAGE0, and wherein said means for modifying modifies the first address for accessing one of n MEMORY pages (PAGE0, PAGE 2, PAGE2 or PAGE3) in accordance with the following table:

FIRST

ADDRESS	ID=0	ID=1	ID=2	ID=3
PAGE0	PAGE 0	PAGE 1	PAGE 2	PAGE 3
PAGE1	PAGE 1	PAGE 0	PAGE 1	PAGE 1
PAGE2	PAGE 2	PAGE 2	PAGE 0	PAGE 2
PAGE3	PAGE 3	PAGE 3	PAGE 3	PAGE 0.

7. In a data processing system having a plurality of CPUs coupled to a common memory, a method of providing an executing program access to a page within the common memory, for each of the CPUs the method comprises the steps of:

generating an identification of the CPU as a function of where the CPU is installed within

the data processing system, the identification being expressed on a signal line or signal lines that convey logic signals;

generating with the CPU a first address for identifying a memory location or locations within a page of the common memory;

modifying the first address as a function of the first address and as a function of the identification of the CPU to generate a second address for identifying a memory location that is either within the same page or within another page of the common memory; and

applying the second address to the common memory to access the memory location or locations.

8. A method as set forth in Claim 7 wherein the step of generating a first address includes the steps of:

generating a virtual memory address;

translating the virtual memory address to a physical memory address for generating the first address; and

applying the first address to modifying means for modifying the first address to generate the second address.

SUBSTITUTE SHEET

A90005

14

9. A method as set forth in Claim 7 wherein there are n CPUs and wherein n equals up to four, wherein the identification (ID) of each of the CPUs is expressed as a value, and wherein the step of modifying is accomplished in accordance with the following table for modifying the first address for accessing one of n MEMORY pages (PAGE0, PAGE1, PAGE2 or PAGE3):

FIRST

ADDRESS	ID=0	ID=1	ID=2	ID=3
PAGE0	PAGE 0	PAGE 1	PAGE 2	PAGE 3
PAGE1	PAGE 1	PAGE 0	PAGE 1	PAGE 1
PAGE2	PAGE 2	PAGE 2	PAGE 0	PAGE 2
PAGE3	PAGE 3	PAGE 3	PAGE 3	PAGE 0.

10. A multi-CPU data processing system including a plurality of CPUs each of which is coupled to a common bus and, through the common bus, to a main memory, each of the CPUs comprising:

means for generating an identification of the CPU as a function of where the CPU is installed within the data processing system, the identification being expressed on a signal line or signal lines that convey logic signals;

means for providing a program that is operating upon the CPU to access a page within the main memory, the providing means comprising,

SUBSTITUTE SHEET

A90005

15

means for generating a first address for identifying a memory location or locations within a page of the main memory;

means, having inputs coupled to the one or more signal lines and to the generated first address, for modifying the first address as a function of the first address and as a function of the identification of the CPU to generate a second address for identifying a memory location or locations that are either within the same page or within another page of the main memory; and

means for applying the second address to the main memory to access the memory location or locations.

SUBSTITUTE SHEET

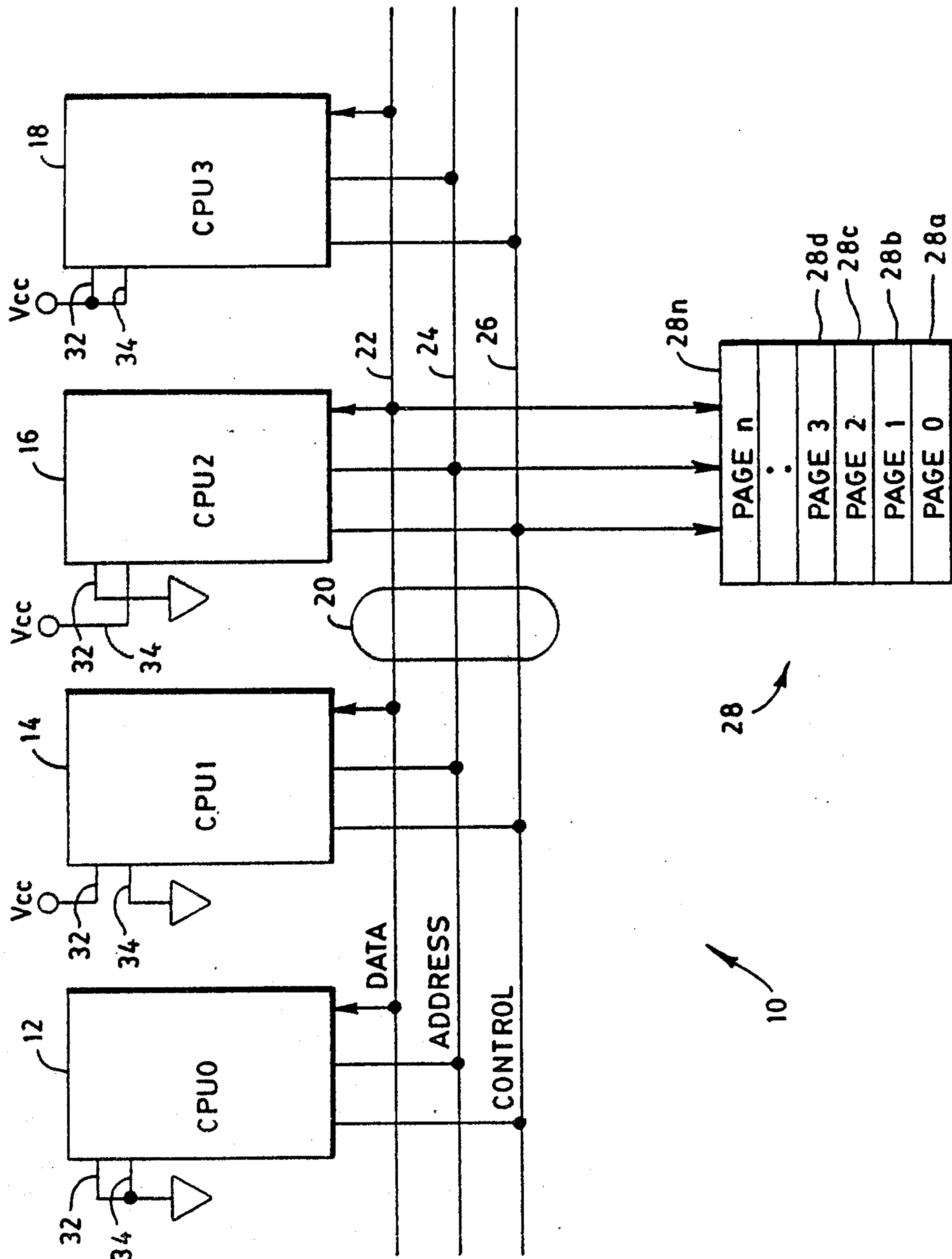


FIG. 1

2081013

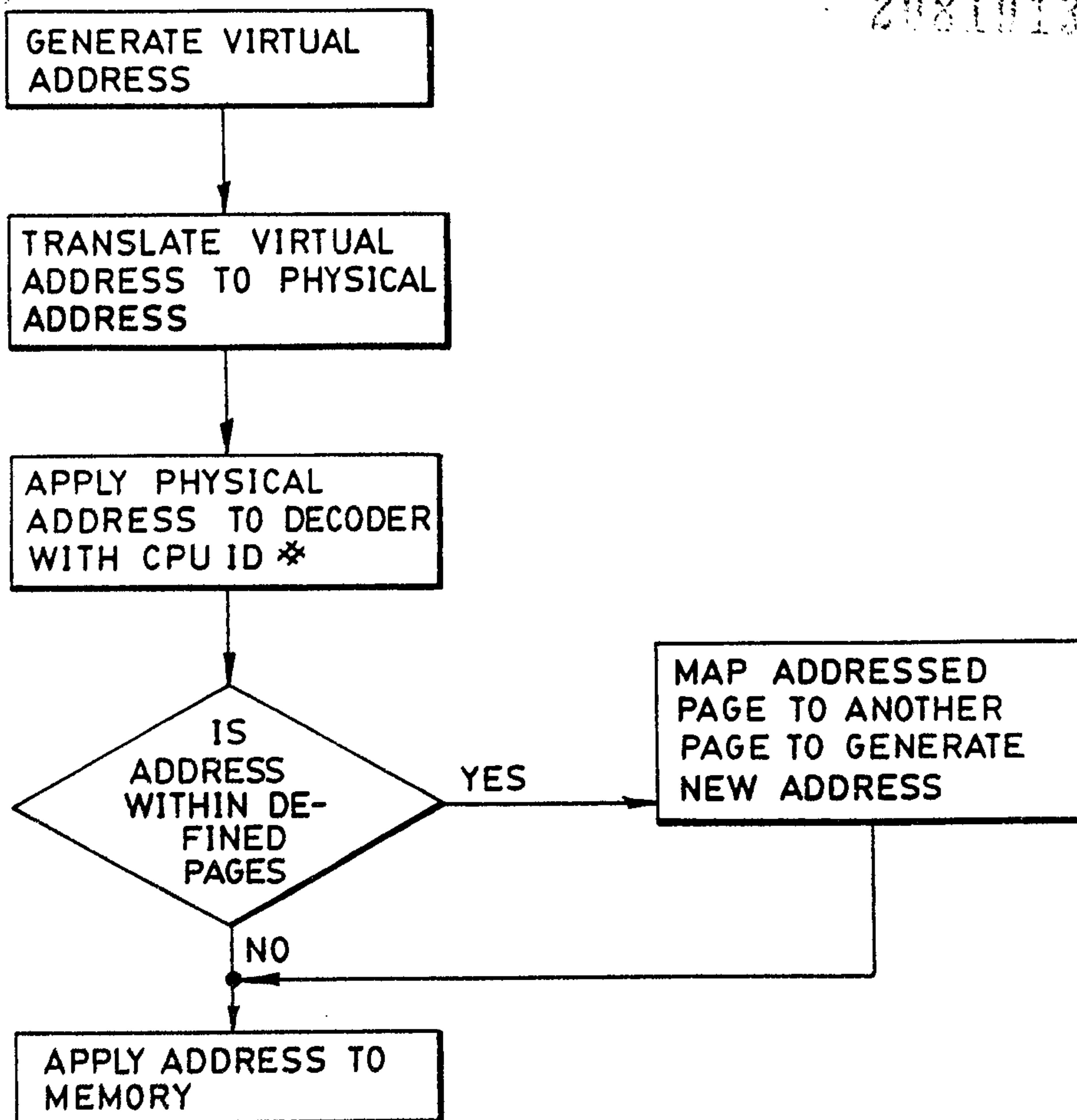


FIG. 3

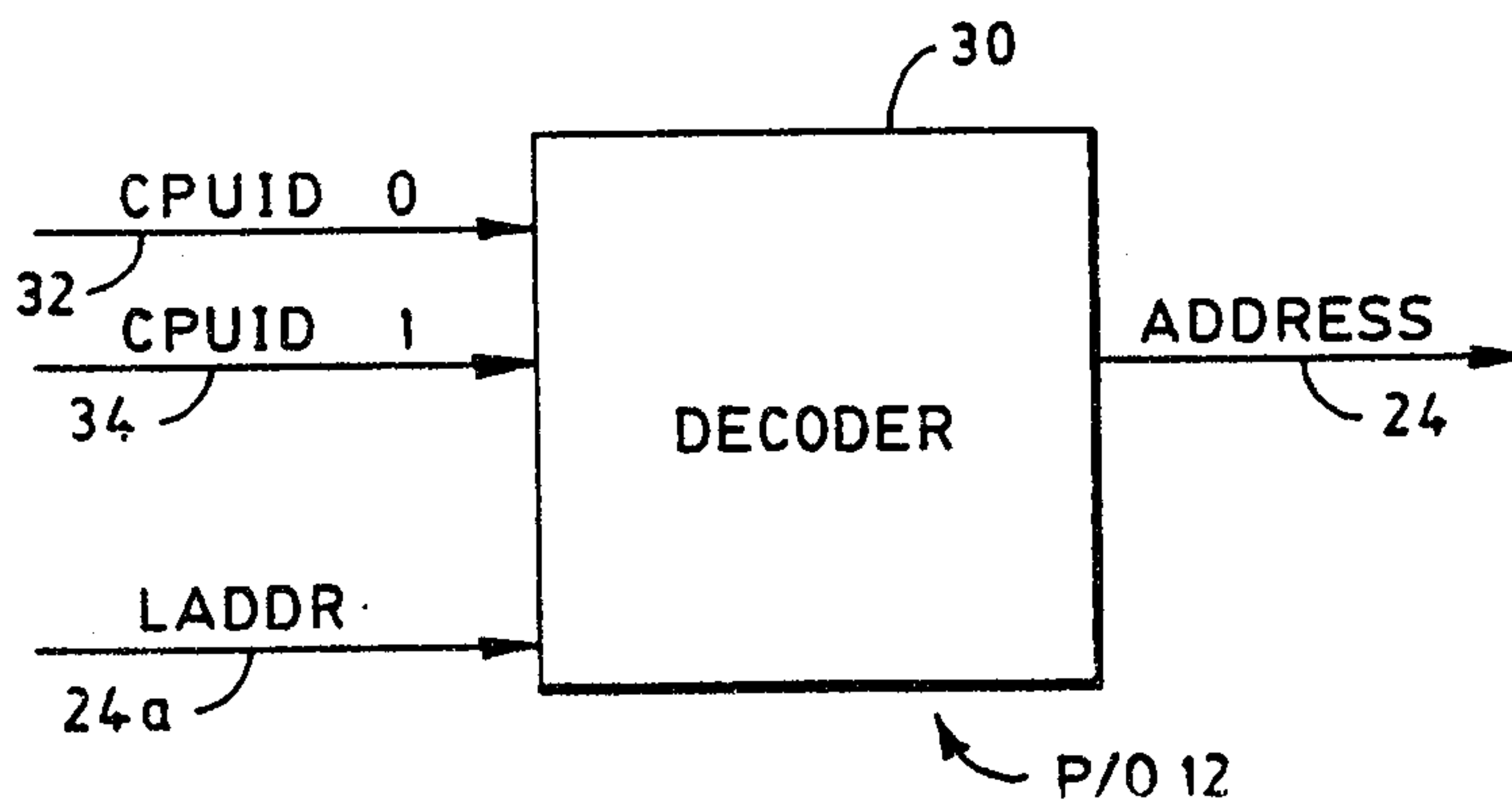


FIG. 2