

Description

Background of the invention

This invention relates generally to improved methods and apparatus for generating a memory address in a data processing system and more particularly to improved methods and apparatus for generating a relatively large virtual or logical address in a data processing system employing virtual or mapped data storage. The most direct approach to generating a large address is to provide a data processor having sufficiently wide data paths to accommodate all of the bits of the address. However, the hardware required for such an approach would be quite expensive.

Another known approach is to provide relatively narrow data paths in the data processor and to then provide for making multiple passes of these narrow data paths in order to manipulate addresses wider than the data paths. The difficulty with this multiple pass approach is that the requirement for these multiple passes results in a significant slow-down in operating speed.

One way of overcoming the disadvantages of the multiple pass approach summarized above is displaced in U.S. Patent No. 4,251,860 issued February 17, 1981, G. R. Mitchell et al., inventors. In this patent, an embodiment is disclosed which permits the use of relatively narrow data widths while avoiding the need for multiple passes in order to produce a large virtual address. This is accomplished by splitting the base address registers which contain the data that is used to form the virtual address into segment registers and offset registers. Then, operation is provided so that only the offset portion of a selected base address is required to participate in arithmetic operations in order to form a virtual address, thereby permitting narrow data paths to be used even though the generated virtual address is much wider.

Additional background information regarding addressing techniques can be found in U.S. Patent No. 4,240,139, issued December 16, 1980; in the article "Indexed Mapping Extends Microprocessor Addressing Range", Ian LeMair, *Computer Design*, August 1980, pp. 111—118; in the article "Storage Relocation Translator", P. L. Martinez, *IBM Technical Disclosure Bulletin*, Vol. 23, No. 5, October 1980, pp. 1785—1789; in the article "Relocation Translator Address Bit Scheme", P. L. Martinez, *IBM Technical Disclosure Bulletin*, Vol. 23, No. 6, November 1980, pp. 2199 and 2200; and in the references cited in the above patent and articles.

Thus from one aspect the present invention relates to apparatus for generating a logical address for use by an address translator in accessing a memory, said apparatus comprising means for storing data indicative of a logical address, said data including index selection data and displacement data; a plurality of index registers each including index data; means responsive to said index selection data for selecting one of said indexed registers; an extended address register; means including an adder for adding the index data in a selected index register to said displace-

ment data and for storing the sum in said extended address register; a logical address register having first and second portions; means for storing a predetermined number of the least significant digits of the sum in said extended address register in said first portion of said logical address register; and table storing means for storing a table containing a plurality of address locating entries corresponding to at least one program performable by said system.

In accordance with this aspect of the invention the apparatus also includes dimension override selection means responsive to a predetermined number of the most significant digits of the sum in said extended address register for accessing a corresponding one of the address locating entries in said table and for storing the accessed entry in said second portion of said logical address register; and means for applying the contents of the first and second portions of said logical address register to said address translator.

From another aspect the invention relates to a method of generating a logical address for use or address translator in accessing a memory, said method comprising providing index selection data and displacement data indicative of a logical address; selecting one of a plurality of index values in response to said index selection data; and adding said displacement data to a selected index value to form a sum.

In accordance with this aspect of the invention the method also includes selecting in response to a predetermined number of the most significant digits of said sum an address locating entry from a table of such entries provided for each of a plurality of programs performable by said system; and applying the selected address locating entry along with the remaining digits of said sum to said address translator.

EP—A—82153 (WO-83/00241), forming state of the art according to Art. 54(3) EPC also relates generally to apparatus as defined above. However, in this case the dimension override selection means is responsive to dimension override data in a selected one of said index registers.

Brief description of the drawings

Fig. 1 is a block and schematic diagram illustrating a preferred embodiment of the invention.

Fig. 2 illustrates an example of the operation of Fig. 1 wherein neither indexing nor dimension override are provided.

Fig. 3 is a graphical representation illustrating the results of the example of Fig. 2.

Fig. 4 illustrates an example of the operation of Fig. 1 wherein indexing is provided without dimension override.

Fig. 5 is a graphical representation illustrating the results of the example of Fig. 4.

Fig. 6 illustrates an example of the operation of Fig. 1 wherein both indexing and dimension override are provided.

Fig. 7 is a graphical representation illustrating the results of the example of Fig. 6.

Detailed description of the invention

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

In prior art virtual or logical address systems, such as disclosed in the aforementioned Patent No. 4,251,860, it is typical to consider memory storage as being divided into identifiable regions commonly referred to as blocks or segments, each segment providing enough addresses so that programs or data stored in them will not try to assign the same addresses more than once. These segments are typically referenced by a two-part logical address, the first part identifying a particular one of the segments, and the second part identifying an offset within the segment.

In the embodiment disclosed in the aforementioned Patent No. 4,251,860, the generation of a virtual address from an instruction involves the use of a base register field provided by the instruction to select a base address comprised of segment and offset values which are split up into separate segment and offset registers, these segment and offset values together constituting the selected base address. A displacement field which is also provided by the instruction is then added to the value contained in the selected offset register to provide a resultant offset value which is concatenated with the segment portion of the base address contained in the selected segment register to form the desired virtual address. Since only the offset portion of the selected base address need be added to the displacement field to form the virtual address, much narrower data paths are required than would otherwise be possible.

Although the virtual address generation approach disclosed in the aforementioned Patent No. 4,251,860 permits a large virtual address to be generated using relatively narrow data paths, it will be evident that the virtual address which can be generated by an instruction is limited to the adding of the instruction's displacement field to a particular base address selected by the instruction's base register field. If an overflow is detected, indicating that the resulting address encroaches on the next segment, operation is aborted. It will thus be understood that the address range which each instruction can access using the approach disclosed in Patent No. 4,251,860 is limited to a single segment. The preferred embodiment of the present invention illustrated in Fig. 1 maintains the narrow data path advantages of Patent No. 4,251,860 while, on the other hand, making it possible for the same instruction to generate addresses capable of accessing data in a plurality of different segments, as will become evident from the description which follows.

Thus, referring to Fig. 1, it will be understood that, as typically occurs during data processing operations, an instruction to be processed is fetched and placed in an instruction register 10. As shown, a typical instruction includes an opcode OP which designates the operation to be performed, and an address portion comprised of

an index I and a displacement D which, as will be described hereinafter together cooperate in the generation of a virtual or logical address. Typically, the opcode OP is sent to the data processor's execution unit (not shown) for performance of the operation designated by the opcode on the data in memory corresponding to the generated address. The particular manner in which such execution may be performed is not part of the present invention.

Still with reference to Fig. 1, it will be seen that the displacement D of the instruction in the instruction register 10 is applied to the right input of an arithmetic and logic unit (ALU) 12, while the index selector IS is applied to an index decoder 14 whose output selects a particular one of a plurality of index registers 16. The selected index 16a is applied to the left side of the ALU 12 for addition with the displacement D from the instruction register 10. The resulting sum at the output of the ALU 12 is stored in an extended address register 15 having an offset portion 15a and a dimension override portion 15b. It will be understood that, depending on the address range which it is desired to permit an instruction to access, the extended addressing register 15 is preferably made wide enough to accommodate the largest sum appearing at the output of the ALU 12 without overflow.

As indicated in Fig. 1, the offset portion 15a in the extended address register 15 is the offset value of the desired virtual or logical address and is loaded into the offset portion 20a of a logical address register 20. The dimension override portion 15b is used to derive the segment which is loaded into the segment portion of the logical address register 20. More specifically, as will be understood from Fig. 1, the dimension override portion 15b of the extended address register 15 is applied to a dimension override decoder 23 which also receives a program number signal indicative of a particular one of a plurality of programs being run by the processor. The program number causes this dimension override decoder 23 to select a particular one of a plurality of segment tables (one for each program being run), and the dimension override portion 15b causes the dimension override decoder 23 to select a particular segment entry 48 in the selected segment table. It will be understood that these segment tables 25 may be provided as part of the CPU's main storage or as separate hardware of firmware.

Each of the segment tables 25 in Fig. 1 includes a plurality of segment entries 48 designating the particular storage segments which its respective program will be permitted to access. These segment entries 48 need not correspond to contiguous portions in real memory. Since one or more of these segment entries 48 may be privileged and/or shared by other programs, each segment entry 48 includes a segment check bit 48a which is sent to the CPU to permit appropriate action to be taken in the event the selected segment is invalid or unavailable. Such action may, for example,

include interrupting the program until the selected segment becomes available.

Assuming that the segment check bit 48a indicates that the selected segment entry 48 in the selected segment table 25 is valid and available, the dimension override decoder 23 then causes this selected segment entry 48 to be read out into the segment portion 20b of the logical address register 20. The concatenated offset and segment in the respective offset and segment portions 20a and 20b of the logical address register 20 thus constitute the full logical address which is applied to an address translator 30 (which may be of conventional form) for translation or mapping into a physical address which is then applied to the CPU main storage 40.

The versatility and flexibility of the address generation approach of the present invention will now be demonstrated with reference to the examples illustrated in Figs. 2—7 along with Fig. 1. For these examples, it will be assumed that the selected segment entry loaded into the segment portion 20b of the logical address register corresponds to the base address of a segment, that the offset loaded into the offset portion 20a is the offset within this segment, and that the address translator 30 concatenates these portions 20a and 20b to obtain the real address in the CPU main storage 40. However, it is to be understood that the present invention is also applicable for use with other types of address translation techniques for deriving a real address in response to the contents of portions 20a and 20b.

Figs. 2 and 3 illustrate an example where neither indexing nor dimension override is called for by an instruction, as indicated by the selected index register 16 (selected by the index selector 15 in the instruction register 10 in Fig. 1) being all zeroes. Typically, each index register 16 may comprise eight binary coded decimal digits. For the case where the selected index register 16 is all zeroes, as illustrated in Fig. 2, the resulting sum loaded into the extended address register 15 is merely the displacement D in the instruction register 10 in Fig. 1. Ordinarily, the maximum value of the displacement D is not large enough to cause any carry over into the dimension override portion 15b of the extended address register 15 so that the dimension override portion 15b will remain at an all zero value, as illustrated in Fig. 2. Typically, the offset portion 15a of the extended address register may comprise six binary coded decimal digits, while the dimension override portion 15b may comprise two binary coded decimal digits.

The all zero value of the dimension override portions 15b in the extended address register 15 selects (via the dimension override decoder 23) the first one of the entries 48 in the segment table 25 corresponding to the current program number. This first entry, which is loaded into the segment portion 20b of the logical address register 20, is merely the base address of segment B which will be assumed to be assigned to the current program. The resulting address in segment B is diagrammatically illustrated in Fig. 3.

Considering next the example shown in Figs. 4 and 5, this example illustrates a situation where indexing is being called for without dimension override. As shown in Fig. 4 the selected index register 16 is, for example, assumed to be 00012163, and the displacement D is, for example, assumed to be 900054. The sum provided by the ALU 12 is thus 912217. The dimension override portion 15b in the extended address register 15 thus remains all zeroes since the sum 912217 provides no carry over into the dimension override portion 15b. Accordingly, the first entry of the segment table 25 is again selected, causing the segment B base register address to again be loaded into the segment portion 20b of the logical address register 20 so as to point to the segment B base address, while the sum 912217 in the offset portion 15a of the extended address register 15 is loaded into the offset portion 20a of the logical address register 20. A typical resulting address is illustrated in Fig. 5.

The last example to be considered is shown in Figs. 6 and 7 and illustrates the situation where both indexing and dimension override are being called for. This is indicated in Fig. 5 where the index in the selected index register is now assumed to have the larger value 1142131, while the displacement D is again assumed to be 900054. Since the sum now provided at the output of the ALU 12 is 02042185, there is a carry over of 02 into the dimension override portion 15b of the extended address register 15. Thus, the dimension override portion 15b now loads into the segment portion 20b of the logical address register 20, the third entry in the segment table 25 corresponding, for example, to the base address of a segment Q. This segment Q may, for example, contain a portion of a large table or other large data structure which is shared by a number of programs and which is too large for inclusion in any one of the program's assigned segments. A typical resulting logical address in segment Q is illustrated in Fig. 7. Of course, if the check bit 48a of the selected entry 48 indicates that it is invalid or unavailable, then the CPU will abort the operation, which may be accomplished in a conventional manner.

It will be understood from the example of Figs. 6 and 7 that, since the value loaded into the dimension override portion 15b of the extended address register 15 is the carry over from the sum of the displacement D and the selected index register 16, there is no need to restrict or abort the translation when overflow occurs, as is the case, for example, in the aforementioned Patent No. 4,251,860, thereby permitting a greatly extended range of addresses to be accessed by an instruction.

Although the present invention has been described in connection with a particular illustrative embodiment, it will be understood that many variations in construction, arrangement and use are possible within the scope of the invention. For example, the present invention is applicable for use with mapped memory as well as virtual memory and various other types of memory organizations.

The present invention is thus to be considered as including all possible modifications and variations encompassed within the scope of the appended claims.

Claims

1. Apparatus for generating a logical address for use by an address translator (30) in accessing a memory (40), said apparatus comprising means (10) for storing data indicative of a logical address, said data including index selection data (IS) and displacement data (D); a plurality of index registers (16) each including index data; means (14) responsive to said index selection data for selecting one of said index registers; an extended address register (15); means including an adder (12) for adding the index data in a selected index register to said displacement data and for storing the sum in said extended address register; a logical address register (20) having first (20a) and second (20b) portions; means for storing a predetermined number of the least significant digits of the sum in said extended address register in said first portion of said logical address register; and table storing means (25) for storing a table containing a plurality of address locating entries corresponding to at least one program performable by said system; characterised in that said apparatus also includes dimension override selection means (23) responsive to a predetermined number of the most significant digits of the sum in said extended address register (15) for accessing a corresponding one of the address locating entries in said table and for storing the accessed entry in said second portion (20b) of said logical address register (20); and means for applying the contents of the first (20a) and second (20b) portions of said logical address register (20) to said address translator (30).

2. Apparatus in accordance with Claim 1, wherein said predetermined number of said most significant digits concatenated with said predetermined number of said least significant digits constitute the sum in said extended address register (15).

3. Apparatus in accordance with Claim 2, wherein said displacement data (D) has a value which is not large enough by itself to cause a carry over into said predetermined number of said most significant digits in said extended address register (15).

4. Apparatus in accordance with any of the preceding Claims, wherein said table storing means (25) provides for storing a respective plurality of address locating entries for each of a plurality of programs performable by said system and wherein said dimension override selection means (23) is also responsive to a program indication for accessing a corresponding address locating entry in a table corresponding to a particular program.

5. Apparatus in accordance with any of the preceding Claims, wherein said means (10) for storing data indicative of a logical address is an

instruction register additionally containing operation data indicating a type of operation to be performed on data located at said logical address.

6. Apparatus in accordance with any of the preceding Claims, wherein said memory (40) includes a plurality of segments, and wherein each address locating entry in a table is indicative of the base address of a particular segment.

7. Apparatus in accordance with Claim 6, wherein a table corresponding to a program includes address locating entries corresponding to the base addresses of segments assigned to the program, and wherein the table also includes at least one address locating entry corresponding to the base address of a segment which is shared with another program.

8. Apparatus in accordance with Claim 7, wherein each address locating entry includes check data indicating the status thereof.

9. Apparatus in accordance with any of Claims 6 to 8, wherein said predetermined number of the least significant digits in said extended address register (15) determines the resultant location accessed in the segment indicated by the selected address locating entry.

10. A method of generating a logical address for use by an address translator (30) in accessing a memory (40), said method comprising providing index selection data (IS) and displacement data (D) indicative of a logical address; selecting one of a plurality of index values in response to said index selection data; and adding said displacement data to a selected index value to form a sum; characterised in that said method also includes selecting response to a predetermined number of the most significant digits of said sum an address locating entry from a table of such entries provided for each of a plurality of programs performable by said system; and applying the selected address locating entry along with the remaining digits of said sum to said address translator (30).

11. A method in accordance with Claim 10, wherein said displacement data has a value which is not large enough by itself to cause a carry over into said predetermined number of most significant digits.

12. A method in accordance with Claim 11, wherein said memory (40) includes a plurality of segments, and wherein each data locating entry in a table is indicative of the base address of a particular segment.

13. A method in accordance with Claim 12, wherein a table corresponding to a program includes address locating entries corresponding to the base addresses of segments assigned to the program, and wherein the table also includes at least one address locating entry corresponding to the base address of a segment which is shared with another program.

14. A method in accordance with Claim 13, wherein said remaining digits of said sum determines the resultant location accessed in the segment indicated by the selected address locating entry.



The present invention is thus to be considered as including all possible modifications and variations encompassed within the scope of the appended claims.

Claims

1. Apparatus for generating a logical address for use by an address translator (30) in accessing a memory (40), said apparatus comprising means (10) for storing data indicative of a logical address, said data including index selection data (IS) and displacement data (D); a plurality of index registers (16) each including index data; means (14) responsive to said index selection data for selecting one of said index registers; an extended address register (15); means including an adder (12) for adding the index data in a selected index register to said displacement data and for storing the sum in said extended address register; a logical address register (20) having first (20a) and second (20b) portions; means for storing a predetermined number of the least significant digits of the sum in said extended address register in said first portion of said logical address register; and table storing means (25) for storing a table containing a plurality of address locating entries corresponding to at least one program performable by said system; characterised in that said apparatus also includes dimension override selection means (23) responsive to a predetermined number of the most significant digits of the sum in said extended address register (15) for accessing a corresponding one of the address locating entries in said table and for storing the accessed entry in said second portion (20b) of said logical address register (20); and means for applying the contents of the first (20a) and second (20b) portions of said logical address register (20) to said address translator (30).

2. Apparatus in accordance with Claim 1, wherein said predetermined number of said most significant digits concatenated with said predetermined number of said least significant digits constitute the sum in said extended address register (15).

3. Apparatus in accordance with Claim 2, wherein said displacement data (D) has a value which is not large enough by itself to cause a carry over into said predetermined number of said most significant digits in said extended address register (15).

4. Apparatus in accordance with any of the preceding Claims, wherein said table storing means (25) provides for storing a respective plurality of address locating entries for each of a plurality of programs performable by said system and wherein said dimension override selection means (23) is also responsive to a program indication for accessing a corresponding address locating entry in a table corresponding to a particular program.

5. Apparatus in accordance with any of the preceding Claims, wherein said means (10) for storing data indicative of a logical address is an

instruction register additionally containing operation data indicating a type of operation to be performed on data located at said logical address.

6. Apparatus in accordance with any of the preceding Claims, wherein said memory (40) includes a plurality of segments, and wherein each address locating entry in a table is indicative of the base address of a particular segment.

7. Apparatus in accordance with Claim 6, wherein a table corresponding to a program includes address locating entries corresponding to the base addresses of segments assigned to the program, and wherein the table also includes at least one address locating entry corresponding to the base address of a segment which is shared with another program.

8. Apparatus in accordance with Claim 7, wherein each address locating entry includes check data indicating the status thereof.

9. Apparatus in accordance with any of Claims 6 to 8, wherein said predetermined number of the least significant digits in said extended address register (15) determines the resultant location accessed in the segment indicated by the selected address locating entry.

10. A method of generating a logical address for use by an address translator (30) in accessing a memory (40), said method comprising providing index selection data (IS) and displacement data (D) indicative of a logical address; selecting one of a plurality of index values in response to said index selection data; and adding said displacement data to a selected index value to form a sum; characterised in that said method also includes selecting in response to a predetermined number of the most significant digits of said sum an address locating entry from a table of such entries provided for each of a plurality of programs performable by said system; and applying the selected address locating entry along with the remaining digits of said sum to said address translator (30).

11. A method in accordance with Claim 10, wherein said displacement data has a value which is not large enough by itself to cause a carry over into said predetermined number of most significant digits.

12. A method in accordance with Claim 11, wherein said memory (40) includes a plurality of segments, and wherein each data locating entry in a table is indicative of the base address of a particular segment.

13. A method in accordance with Claim 12, wherein a table corresponding to a program includes address locating entries corresponding to the base addresses of segments assigned to the program, and wherein the table also includes at least one address locating entry corresponding to the base address of a segment which is shared with another program.

14. A method in accordance with Claim 13, wherein said remaining digits of said sum determines the resultant location accessed in the segment indicated by the selected address locating entry.



Patentansprüche

1. Gerät zum Erzeugen einer logischen Adresse für einen Adressenumsetzer (30) zum Zugriff zu einem Speicher (40); mit einer Einrichtung (10) zum Speichern von Daten, die eine logische Adresse anzeigen und die Indexauswahldaten (IS) und Distanzdaten (D) enthalten;

mit einer Mehrzahl von Indexregistern (16), von denen jedes Indexdaten enthält;

mit einer Einrichtung (14), die auf die Indexauswahldaten anspricht, um eines der Indexregister auszuwählen;

mit einem erweiterten Adressregister (15);

mit einer Einrichtung, die ein Addierwerk (12) zum Addieren der Indexdaten zu den Distanzdaten in einen ausgewählten Indexregister aufweist und die Summe in dem erweiterten Adressregister abspeichert;

mit einem logischen Adressregister (20), das erste (20a) und zweite (20b) Abschnitte aufweist;

mit einer Einrichtung zum Abspeichern einer vorbestimmten Anzahl von Ziffern niedrigster Wertigkeit der im erweiterten Adressregister stehenden Summe in dem ersten Abschnitt des logischen Adressregisters; und

mit einer Tabellen-Speichereinrichtung (25) zum Speichern einer Tabelle, die eine Mehrzahl von Adressdaten entsprechend mindestens einem, durch das System verarbeitbaren Programms enthält;

gekennzeichnet durch eine Dimensionsübersteuerungs-Auswahleinrichtung (23), die auf eine vorbestimmte Anzahl von Ziffern höchster Wertigkeit der in dem erweiterten Adressregister (15) stehenden Summe anspricht, um ein entsprechendes Adressdatum der Tabelle anzusteuern und dieses in dem zweiten Abschnitt (20b) des logischen Adressregisters (20) zu speichern; und eine Einrichtung zum Zuführen der Inhalte der ersten (20a) und zweiten (20b) Abschnitte des logischen Adressregisters (20) an den Adressenumsetzer (30).

2. Gerät nach Anspruch 1, dadurch gekennzeichnet, daß die vorbestimmte Anzahl von Ziffern höchster Wertigkeit mit der damit zusammenhängenden vorbestimmten Anzahl von Ziffern niedrigster Wertigkeit die Summe in dem erweiterten Adressregister (15) bilden.

3. Gerät nach Anspruch 2, dadurch gekennzeichnet, daß die Distanzdaten (D) einen Wert haben, der selbst nicht groß genug ist, um einen Übertrag in der vorbestimmten Anzahl von Ziffern höchste Wertigkeit in dem erweiterten Adressregister (15) zu bewirken.

4. Gerät nach einem oder mehreren der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Tabellen-Speichereinrichtung (25) zur Speicherung einer entsprechenden Anzahl Adressdaten für jedes einer Mehrzahl von durch das System verarbeitbaren Programmen eingerichtet ist; und daß die Dimensionsübersteuerungs-Auswahleinrichtung (23) auch auf eine Programmanzeige anspricht, um ein bestimmtes Adressdatum in einer Tabelle ent-

sprechend einem bestimmten Programm anzusteuern.

5. Gerät nach einem oder mehreren der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Einrichtung (10) zum Speichern der eine logische Adresse anzeigenden Daten ein Befehlsregister ist, das zusätzlich Befehlsdaten enthält, die die Art eines Befehls angeben, der in Verbindung mit den an der logischen Adresse befindlichen Daten durchgeführt werden soll.

6. Gerät nach einem oder mehreren der vorstehenden Ansprüche, dadurch gekennzeichnet, daß der Speicher (40) eine Mehrzahl von Segmenten enthält und daß jedes Adressdatum in einer Tabelle die Basisadresse eines bestimmten Segmentes angibt.

7. Gerät nach Anspruch 6, dadurch gekennzeichnet, daß eine einem Programm entsprechende Tabelle Adressdaten enthält, die den Basisadressen der dem Programm zugeordneten Segmente entsprechen, und daß die Tabelle darüber hinaus mindestens ein Adressdatum enthält, das der Basisadresse eines Segments entspricht, das mit einem anderen Programm gemeinsam benutzt wird.

8. Gerät nach Anspruch 7, dadurch gekennzeichnet, daß jedes Adressdatum Prüfdaten zur Angabe des Status desselben enthält.

9. Gerät nach einem oder mehreren der Ansprüche 6 bis 8, dadurch gekennzeichnet, daß die vorbestimmte Anzahl von Ziffern niedrigster Wertigkeit in dem erweiterten Adressregister (15) die angesteuerte Adresse in dem Segment bestimmt, das durch das ausgewählte Adressdatum bezeichnet wird.

10. Verfahren zum Erzeugen einer logischen Adresse für einen Adressenumsetzer (30) zum Zugriff zu einem Speicher (40);

bei dem eine logische Adresse durch Indexauswahldaten (IS) und Distanzdaten (D) bestimmt wird;

bei dem einer von mehreren Indexwerten abhängig von den Indexauswahldaten ausgewählt wird; und

bei dem die Distanzdaten einem ausgewählten Indexwert hinzuaddiert werden, um eine Summe zu bilden;

dadurch gekennzeichnet, daß anhängig von einer vorbestimmten Anzahl von Ziffern höchster Wertigkeit der Summe ein Adressdatum aus einer Tabelle mit solchen Daten ausgewählt wird, wobei für jedes einer Mehrzahl von durch das System verarbeitbaren Programmen eine Tabelle vorgesehen ist; und

daß das ausgewählte Adressdatum zusammen mit den übrigen Ziffern der Summe dem Adressenumsetzer (30) zugeführt wird.

11. Verfahren nach Anspruch 10, dadurch gekennzeichnet, daß die Distanzdaten einen Wert haben, der selbst nicht groß genug ist, um einen Übertrag in der vorbestimmten Anzahl von Ziffern höchste Wertigkeit zu bewirken.

12. Verfahren nach Anspruch 11, dadurch gekennzeichnet, daß der Speicher (40) eine Mehrzahl von Segmenten enthält und daß jedes



Adressdatum in einer Tabelle die Basisadresse eines bestimmten Segments angibt.

13. Verfahren nach Anspruch 12, dadurch gekennzeichnet, daß eine einem Programm entsprechende Tabelle Adressdaten enthält, die den Basisadressen der dem Programm zugeordnete Segmente entsprechen, und daß die Tabelle darüber hinaus mindestens eine Adressdatum enthält, das der Basisadresse eines Segmentes entspricht, das mit einem anderen Programm gemeinsam benutzt wird.

14. Verfahren nach Anspruch 13, dadurch gekennzeichnet, daß die übrigen Zifferen der Summe die resultierende Adresse in dem Segment bezeichnen, das durch das ausgewählte Adressdatum angegeben wird.

Revendications

1. Appareil de génération d'une adresse logique destinée à être utilisée dans un traducteur d'adresse (30) pour accéder à une mémoire (40), cet appareil comprenant des moyens (10) pour stocker des données indiquant une adresse logique, ces données comprenant des données de sélection de repère (IS) et des données de déplacement (D); un certain nombre de registres de repères (16) comprenant chacun des données de repères des moyens (14) répondant aux données de sélection de repères pour sélectionner l'un des registres de repères; un registre d'adresses élargies (15); des moyens comprenant un additionneur (12) pour additionner les données de repères contenues dans un registre de repères sélectionné, aux données de déplacement, et pour stocker la somme dans le registre d'adresses élargies; un registre d'adresses logiques (20) comprenant une première partie (20a) et une seconde partie (20b); des moyens pour stocker un nombre prédéterminé des derniers chiffres significatifs de la somme dans le registre d'adresses élargies de la première partie du registre d'adresses logiques; et des moyens de stockage de tableau (25) pour stocker un tableau contenant un certain nombre d'entrées de localisation d'adresse correspondant à au moins un programme pouvant être effectué par le système; caractérisé en ce que l'appareil comprend également des moyens de sélection de dépassement de dimensions (23) répondant à un nombre prédéterminé des chiffres les plus significatifs de la somme dans le registre d'adresses élargies (15) pour accéder à l'une correspondante des entrées de localisation d'adresse dans le tableau, et pour stocker l'entrée à laquelle on a accédé dans la seconde partie (20b) du registre d'adresses logiques (20); et des moyens pour appliquer le contenu des première (20a) et seconde (20b) parties du registre d'adresses logiques (20), au traducteur d'adresse (30).

2. Appareil selon la revendication 1, caractérisé en ce que le nombre prédéterminé des chiffres les plus significatifs, enchainés avec le

nombre prédéterminé des chiffres les moins significatifs, constituent la somme se trouvant dans le registre d'adresses élargies (15).

3. Appareil selon la revendication 2, caractérisé en ce que les données de déplacement (D) présentent une valeur qui n'est pas suffisamment grande en elle-même pour produire un passage dans le nombre prédéterminé des chiffres les plus significatifs du registre d'adresses élargies (15).

4. Appareil selon l'une quelconque des revendications précédentes, caractérisé en ce que les moyens de stockage de tableau (25) assurent le stockage d'un nombre correspondant d'entrées de localisation d'adresse pour chacun de plusieurs programmes pouvant être effectuées par le système, et en ce que les moyens de sélection de dépassement de dimensions (23) répondent également à une indication de programme pour accéder à une entrée de localisation d'adresse correspondante dans un tableau correspondant à un programme particulier.

5. Appareil selon l'une quelconque des revendications précédentes, caractérisé en ce que les moyens (10) pour stocker les données indiquant une adresse logique, sont constitués par un registre d'instruction contenant en outre des données d'opération indiquant un type d'opération à effectuer sur les données situées à l'adresse logique.

6. Appareil selon l'une quelconque des revendications précédentes, caractérisé en ce que la mémoire (40) comprend un certain nombre de segments, et en ce que chaque entrée de localisation d'adresse dans un tableau indique l'adresse de base d'un segment particulier.

7. Appareil selon la revendication 6, caractérisé en ce qu'un tableau correspondant à un programme comprend des entrées de localisation d'adresse correspondant aux adresses de base des segments assignés au programme, et en ce que le tableau comprend également au moins une entrée de localisation d'adresse correspondant à l'adresse de base d'un segment partagé avec un autre programme.

8. Appareil selon la revendication 7, caractérisé en ce que chaque entrée de localisation d'adresse comprend des données de vérification indiquant l'état de celle-ci.

9. Appareil selon l'une quelconque des revendications 6 à 8, caractérisé en ce que le nombre prédéterminé des chiffres les moins significatifs dans le registre d'adresses élargies (15), détermine la localisation résultante à laquelle on accède dans le segment indiqué par l'entrée de localisation d'adresses sélectionnée.

10. Procédé de génération d'une adresse logique destinée à être utilisée par un traducteur d'adresse (30) pour accéder à une mémoire (40), ce procédé consistant à utiliser des données de sélection de repères (IS) et des données de déplacement (D) indiquant une adresse logique; à sélectionner l'une de plusieurs valeurs de repères en réponse aux données de sélection de repères et à ajouter des données

de déplacement à une valeur de repère sélectionnée pour former une somme; caractérisé en ce que le procédé consiste également à sélectionner, en réponse à un nombre prédéterminé des chiffres les plus significatifs de la somme; une entrée de localisation d'adresse provenant d'un tableau de ces entrées utilisées pour chacun de plusieurs programmes pouvant être effectuées par le système; et à appliquer l'entrée de localisation d'adresse sélectionnée en même temps que les chiffres restant de la somme, au traducteur d'adresse (30).

11. Procédé selon la revendication 10, caractérisé en ce que les données de déplacement présentent une valeur qui n'est pas suffisamment grande en elle-même pour produire un passage dans le nombre prédéterminé des chiffres les plus significatifs.

12. Procédé selon la revendication 11, caracté-

risé en ce que la mémoire (40) comprend un certain nombre de segments, et en ce que chacune des entrées de localisation de données d'un tableau indique l'adresse de base d'un segment particulier.

13. Procédé selon la revendication 12, caractérisé en ce qu'un tableau correspondant à un programme comprend des entrées de localisation d'adresse correspondant aux adresses de base des segments assignés au programme, et en ce que le tableau comprend également au moins une entrée de localisation d'adresse correspondant à l'adresse de base d'un segment partagé avec un autre programme.

14. Procédé selon la revendication 13, caractérisé en ce que les chiffres restants de la somme déterminent la position résultante à laquelle on accède dans le segment indiqué par l'entrée de localisation d'adresse sélectionnée.

5

10

15

20

25

30

35

40

45

50

55

60

65

