

Description

BACKGROUND OF THE INVENTION:

The present invention relates to a data processing technology, and in particular, to a technology effectively applicable to an instruction system in a program controlled system, for example, to an instruction system of a microprocessor having at least two register sets.

In some microcomputers of the Z80 type (Zylog), as shown in FIG. 7, two register sets RSA and RSB are provided so that a task processing can be achieved by use of one of these register sets. According to this provision, for example, when a task is completed by use of the register set RSA, control can be immediately passed to another task processing by changing the register set from the RSA to the RSB with the contents of the registers unchanged. To return control to the original task when the task processing is completed, the register set is changed to the RSA, which enables to use the remaining contents of the registers associated with the original task to be executed. Consequently, this method has advantages that operations such as an operation to save registers are not necessary when a task is changed and that a high execution speed can be developed.

A similar type of data processing system, which includes the features of the first part of claim 1, is disclosed in "Conference proceedings", The 6th Annual Symposium on Computer Architecture, April 23-25, 1979, pages 137-146.

However, there exist some cases where an argument is required to be transferred to execute a new processing when control is passed from a task processing to the another task processing. As shown in FIG. 7, if the system is so controlled to completely change the register set, there arises a disadvantage that the argument cannot be easily transferred.

In contrast, the reduced instruction set computer (RISC) produced by the UC Berkeley is provided with a plurality of register sets RSA, RSB, RSC, etc. in which a fixed number of registers are overlapped with each other as shown in FIG. 8. As a result, operations such as a transfer of arguments can be easily accomplished when control jumps to a subroutine. However, when a plurality of register sets are overlapped with each other, the overlapped registers cannot be used in a case where such operations as a transfer of arguments are not required when a task change takes place, which therefore leads to a disadvantage that the available number of registers is limited.

SUMMARY OF THE INVENTION:

It is therefore an object of the present invention to provide a data processing system which increases the flexibility available at a change of a plurality of register sets, thereby improving the execution speed of the data processing system.

This object is met by a data processing system as characterised in claim 1.

In a data processing system according to the invention and having a plurality of register sets, a new instruction is provided which passes the content of a register specified when a register set is changed to a register in another register set and is enabled to be implemented by hardware. This allows that a register set is changed with or without a transfer of an argument or the like. As a result, a high-speed task change and subroutine call can be effected, and hence the above-mentioned object can be achieved, namely, the flexibility in a change of the register set is improved and the execution speed of the system is improved.

BRIEF DESCRIPTION OF THE DRAWINGS:

The present invention will be apparent from the following detailed description taken in conjunction with the accompanying diagrams in which:

FIGS. 1A - 1G are schematic diagrams illustrating format examples of register set change instructions according to the present invention;

FIG. 2 is a schematic diagram depicting the configuration example of a register bank in FIGS. 4 - 7;

FIGS. 3A - 3D are explanatory diagrams illustrating operations of register set change instructions according to the present invention;

FIGS. 4 - 6 are schematic block diagrams showing microprocessor configuration examples which enables execution of register set change instructions according to the present invention; and

FIGS. 7 - 8 are explanatory diagrams illustrating register change procedures in the conventional microprocessor.

DESCRIPTION OF THE PREFERRED EMBODIMENTS:

FIGS. 1A - 1G are schematic diagrams illustrating examples of formats of register set change instructions according to the present invention.

The format shown in FIG. 1A is a format suitable for an instruction which is used to change a register set (to be referred to as a register bank herebelow) to an adjacent register bank among a plurality of register banks. An operation code specify field OP₁ is used to contain a code for incre-

menting or decrementing a register bank number. A field R following the operation code specify field OP_1 is provided to indicate whether or not a transfer of the register content has been achieved and is constituted from one bit. If this field R is set to "0", when a register bank (for example, BN_1) is changed to the immediately preceding or following register bank BN_{i-1} or BN_{i+1} , respectively, the bank is changed without transferring the contents of any registers in the register bank BN_i .

On the other hand, if the field R is set to "1", for the registers specified by the register specify field Rn-list following the field R, the contents thereof are indicated to be transferred to the corresponding registers of the register bank BN_{i-1} or BN_{i+1} after the register bank is changed.

Moreover, in this embodiment, though not limited to, it is assumed that each register bank comprises 16 registers, that the register specify field Rn-list is constituted from 16 bits, and that a register to which the content is to be transferred can be specified according to the bit correspondence. For example, in the example of FIG. 1A, it is indicated that the contents of the first and fourth registers R_1 and R_4 , respectively in the register bank in use are to be passed to the first and fourth registers in a register destination bank after the change.

FIG. 3A is a diagram showing a case where by using this instruction, the register bank is changed (incremented) from BN_1 to BN_2 . However, in this case, an instruction (represented as IBNR) is used to increment the bank number. To restore the register bank from BN_2 to BN_1 , an instruction (represented as DBNR) to decrement the bank number need only be used, where the field R and the register specify field Rn-list need only be set as described above.

Next, an instruction format shown in FIG. 1B will be described. This format is suitable for an instruction (represented as CBNR) which changes the register bank in use to a register bank at a location separated therefrom (i.e. not adjacent thereto).

This CBNR instruction is so structured to have an immediate data specify field Imm to specify a register bank number of the change destination between the field R and the register specify field Rn-list of the instruction format of FIG. 1A. With this CBNR instruction, the register bank is changed, for example, from BN_0 to BN_2 . Furthermore, in this operation, by specifying arbitrary registers in the bank, for example, registers R_1 and R_4 as shown in FIG. 3B, the register bank can be changed with or without a transfer of the contents of the registers. The presence or absence of the register transfer is determined by an information in the field R, whereas the content of the instruction is determined by an information in the operation

specify field OP_2 .

Although not limited to, in this embodiment, the immediate data specify field Imm is set to have a length of eight bits due to the relationships with respect to the operation word length. As a result, up to 256 register banks can be handled.

In addition, when a register bank change is achieved with a CBNR instruction, the operation to return to the original register bank will be facilitated if the original register bank number before the change is saved in a stack area or the like. Consequently, in such a case, with an instruction (represented as PBNR) having a format identical to the format shown in FIG. 1A, the original register bank number is fetched from the stack, which enables to restore the original register bank without using the CBNR instruction described above.

Next, the instruction format shown in FIG. 1C will be described. This format comprises an operation code specify field OP_3 and a register specify field Rn-list. A field R dedicated to indicate whether or not a transfer of the register content takes place is not provided. The information related to the present or absence of the transfer of the register content is contained in the operation specify field OP_3 . Although not limited to, the operation code specify field OP_3 is constituted from eight bits. When this instruction is executed, the register set is changed as shown in FIG. 3A.

Next, the instruction format shown in FIG. 1D will be described. This format comprises an operation code specify field OP_4 , an immediate data specify field Imm, and a register specify field Rn-list. A field R dedicated to indicate whether or not the transfer of the register content takes place is not provided. The information related to the presence or absence of the transfer of the register content is contained in the operation specify field OP_4 . When this instruction is executed, a change of the register set is accomplished as shown in FIG. 3B.

Next, the instruction format shown in FIG. 1E will be described. This format comprises an operation code specify field OP_5 , a transfer presence/absence specify field R, an immediate data specify field Imm, a transfer source register specify field Rn-list (source), and a transfer destination register specify field Rn-list (destination). The transfer destination register specify field Rn-list (destination) is divided into 16 fields RL_0 - RL_{15} corresponding to the number of registers in the register bank. To each field of the fields RL_0 - RL_{15} , an area for four bits is allocated to indicate a register number of the transfer destination. For example, the field RL_2 specifies the seventh register, namely, register R_7 in the transfer destination register bank. Furthermore, the field RL_{15} specifies the 12th register, namely, register RL_{12} in the

transfer destination register bank. FIG. 3C is a diagram illustrating an example of a case where the register bank is changed according to the instruction shown in FIG. 1E. Among the registers in the register bank BN_0 , the registers R_2 and R_{15} to be transferred are specified by the information in the transfer source register specify field $Rn\text{-list}$ (source). The bank number BN_2 of the register bank after the change is specified by the information in the immediate data specify field Imm . By using the instruction format shown in FIG. 1E, a transfer between arbitrary registers becomes possible between arbitrary register banks.

Next the instruction format shown in FIG. 1F will be described. This format comprises an operation code specify field OP_6 , an immediate data specify field Imm , a transfer source register specify field $Rn\text{-list}$ (source), and a transfer destination register specify field $Rn\text{-list}'$ (destination). Unlike the format shown in FIG. 1E, the transfer destination register specify field $Rn\text{-list}'$ is divided into two fields RLa and RLb . To each field of the fields RLa and RLb , four bits are allocated to specify a register number in the register bank after the change. This instruction format is especially effective in a case where the number of the registers to be transferred is limited to two or less. This is because the total bit length of the instruction format can be minimized. For example, the transfer destination register specify field $Rn\text{-list}$ (destination) of FIG. 1E is constituted from 64 bits; whereas, the transfer destination register specify field $Rn\text{-list}'$ (destination) of FIG. 1F can be constituted from only eight bits. Moreover, in the format of FIG. 1F, an information to establish a correspondence between a transfer source register number and a transfer destination register number is required to be included in the operation specify field OP_6 . That is, the correspondence is established such that the transfer destination of the register R_2 specified by the transfer source register specify field $Rn\text{-list}$ (source) is specified by the field RLa and the transfer destination of the register R_{15} is specified by the register RLb . FIG. 3C shows an example of a case where the register bank is changed according to an instruction shown in FIG. 1F.

Next, the instruction format shown in FIG. 1G will be described. This format comprises an operation code specify field OP_7 , an immediate data specify field Imm , and a transfer source register specify field $Rn\text{-list}$ (source). FIG. 3D shows an example of a case where the register bank is changed according to the instruction shown in FIG. 1G. In this instruction format, the transfer destination specify field is not disposed. If there exist registers to be transferred when the register bank is changed, the registers are specified beginning from the register R_0 having the smallest register

bank number in the new register bank BN_2 . Such a specify sequence, is determined by the information in the operation code specify field OP_7 .

Referring now to FIG. 4, a description will be given of an example of a hardware configuration of a microprocessor which enables an execution of a register bank change instruction configured in the format described above.

The microprocessor of this embodiment is provided with a control section of the microprogram controlled system. That is, an LSI chip constituting the microprocessor includes a micro read only memory (ROM) 2 in which a microprogram is stored. The micro ROM 2 is accessed by use of a micro address decoder 5 so as to sequentially read the microinstructions constituting the microprogram.

An address generated by a micro address generate circuit 4 based on an operation code of a macro instruction fetched into an instruction register 3 is supplied to the micro address decoder 5. By decoding the address, the first instruction of a group of microinstructions for executing the macro instruction is read and then the control signals are generated for the components such as an execution unit 6 comprising various temporary registers $REG_1 - REG_n$, a data buffer DB, an arithmetic logic unit ALU, and the like. Among the group of microinstructions associated with the macro instruction, the second and subsequent microinstructions are read, when a code of the next address field of the microinstruction previously read is supplied to the micro address decoder 5, based on the next address of the previous microinstruction and the address from the micro address generate circuit 4. With the control signals generated after a series of micro instructions are read, the execution unit 6 is controlled and, for example, a macro instruction such as an add instruction represented by ADD is executed.

In this embodiment, there is provided a memory access control circuit 7 having a function to control a group of general purpose registers 12 including 16 register banks $BN_0 - BN_{15}$ and a group of control registers 11. The 16 register banks $BN_0 - BN_{15}$ are, although not limited to, each comprise 32-bit registers 0 - 15, for example, as shown in FIG. 2. The control register is provided with a status register SR, a stack pointer SP, and a bank number register BNR keeping the register bank number currently in use. In this embodiment, the group of general-purpose registers 12 including the register banks $BN_0 - BN_{15}$ and the group of control registers 11 are constituted from a random access memory (RAM), which is connected via an internal bus 10 to a data buffer DB in the execution unit 6.

Moreover, the instruction register 3 comprises a first-in first-out (FIFO) memory. A macro instruction stored in an external memory is read in the byte-by-byte basis when the memory access control circuit 7 accesses an address bus 8 and then the macro instruction is supplied via the data bus 9 to the instruction register 3 comprising an FIFO memory. In this manner, a plurality of operation codes and operands are fetched into the instruction register in advance.

As described above, in the microprocessor of this embodiment, each task can be processed by use of one of the 16 register banks $BN_0 - BN_{15}$.

When a register bank change instruction IBNR or DBNR shown in FIG. 1A is stored in the instruction register 3 of the microprocessor, a corresponding micro instruction is read from the micro ROM 2. The memory access control circuit 7 is controlled by the microcode, for example, if the register bank being used by the memory access control circuit 7 up to the point is BN_i , the change is performed to effect the subsequent processing by use of the BN_{i+1} or BN_{i-1} . If the bit of the field R is "1", for the registers specified by Rn-list, a processing is achieved in which the contents of the registers are moved to the corresponding registers in another bank or to the registers particularly specified therefor. Moreover, the bank number in the bank number register BNR is changed.

On the other hand, when a corresponding micro instruction is read after a register bank change instruction CBNR such as shown in FIG. 1B is stored in the instruction register 3, the register bank number specified by the immediate data specify field Imm is written in the bank number register BNR and hence the bank number is changed. At the same time, the specified registers are transferred. Moreover, the bank number in use contained in the bank number register BNR up to the point is stored together with the content of the program counter and the like in a stack area of the external memory.

When the register bank change instruction PBNR is loaded, the saved original register number is read from the stack area specified by a stack pointer in the control register group 11 and is loaded into the bank register BNR, and at the same time, the bank in use in the general-purpose register group 12 is changed and the necessary contents of registers are transferred.

Consequently, by use of the instruction CBNR described above, at a subroutine call for example, another task processing can be executed by effecting a jump from the main routine to a subroutine with a transfer of data such as arguments; moreover, by use of the PBNR instruction, control can be rapidly returned to the original position of the main routine, thereby continuing the original pro-

cessing.

FIG. 5 is a schematic diagram showing another configuration example of a microprocessor which enables an execution of a register bank change instruction according to the present invention. The different point, when compared with the microprocessor shown in FIG. 4, is configuration means of the control register group. The control register group 11 is disposed in the RAM in FIG. 4, whereas the control group of FIG. 5 is constituted from a plurality of registers as dedicated store means. Like the temporary registers $REG_1 - REG_n$ in the same type of configuration, these registers may be disposed in the execution unit 6.

FIG. 6 is a diagram illustrating another configuration example of a microprocessor which enables an execution of a register bank change instruction according to the present invention. The difference, when compared with the microprocessor of FIG. 5, is configuration means of the register banks $BN_0 - BN_{15}$. The register banks $BN_0 - BN_{15}$ are disposed in the RAM in FIG. 5, whereas the register banks $BN_0 - BN_{15}$ of FIG. 6 are constituted from a plurality of registers as dedicated store means. Like the temporary registers $REG_1 - REG_n$ in the same type of configuration, may be disposed in the execution unit 6. By disposing the register banks $BN_0 - BN_{15}$ in the execution unit 6, the data buffer DB for keeping the contents of the register banks $BN_0 - BN_{15}$ need not be disposed in the execution unit 6. Moreover, a register access controller 7' is disposed in place of the memory access controller 7.

In addition, the register bank change instruction proposed by the present invention is not limited to the formats of FIGS. 1A - 1G, for example, the field lengths of the register specify field Rn-list and the immediate data specify field Imm can be naturally altered according to the capacity of the RAM (12) as the general-purpose register group and the number of registers of each register bank BN_i .

According to the present invention, a microprocessor having a plurality of register sets is provided with a new instruction used with a register specification for changing a register set to transfer the content of the register to a register in another register set, and hence a change of a register set can be achieved with or without a transfer of data such as arguments. This enables to effect a high-speed task change and subroutine call, which leads to the effect that the flexibility in the register set change is improved and the execution speed of the system is increased.

While the invention of the present inventor has been concretely described with reference to the particular embodiments, it is not to be restricted by those embodiments but only by the appended claims. For example, the register banks $BN_0 - BN_{15}$ may be so configured to be disposed exter-

nally with respect to the microprocessor 1. Moreover, in the description above, although the invention of the present inventor has been described in a case where the invention is applied to a microprocessor, namely, a utilization field which is the background of the invention, it is not restricted by that utilization field, that is, the present invention can be used for a program controlled data processing system in general such as a computer and a mini-computer.

Claims

1. A data processing system for processing a plurality of tasks each involving the contents of one of a plurality of register groups ($BN_0, BN_1, \dots$), wherein a change from one register group to another register group is effected in accordance with a register change instruction, characterized in that said register change instruction includes information (Rn-list(SOURCE)) for specifying at least one register ($R_0, R_1, \dots$) in said one register group, and that the contents in the specified at least one register in said one register group are transferred to said other register group in response to said register change instruction.
2. The system of claim 1, wherein said register change instruction further includes information (Rn-list(DESTINATION)) for specifying at least one register ($RL_0, RL_1, \dots$) in said other register group into which the contents of said at least one register ($R_0, R_1, \dots$) are to be transferred.
3. The system of claim 1 or 2, wherein said register change instruction further includes an indication (R) as to whether the contents of any of the registers ($R_0, R_1, \dots$) of said one register group are to be transferred to another register group.
4. The system of any of claims 1 to 3, wherein said register change instruction contains a number of bits each corresponding to one register ($R_0, R_1, \dots$) in a register group ($BN_0, BN_1, \dots$) and indicating whether the contents of the respective register are to be transferred.
5. The system of any of claims 1 to 4, wherein said register change instruction further contains information (Imm) for specifying a bank number of the other register group.

Patentansprüche

1. Datenverarbeitungssystem zum Verarbeiten mehrerer Aufgaben, deren jede mit dem Inhalt

einer von mehreren Registergruppen ($BN_0, BN_1, \dots$) arbeitet, wobei gemäß einem Registerwechselbefehl ein Wechsel von einer Registergruppe zu einer anderen erfolgt,

dadurch gekennzeichnet, daß der Registerwechselbefehl Informationen (Rn-list(SOURCE)) zum Spezifizieren mindestens eines Registers ($R_0, R_1, \dots$) in der besagten einen Registergruppe enthält, und daß der Inhalt des spezifizierten mindestens einen Register in der besagten einen Registergruppe entsprechend dem Registerwechselbefehl auf die andere Registergruppe übertragen wird.

2. System nach Anspruch 1, wobei der Registerwechselbefehl ferner Informationen (Rn-list(DESTINATION)) zum Spezifizieren mindestens eines Registers ($RL_0, RL_1, \dots$) in der besagten anderen Registergruppe enthält, auf die der Inhalt des besagten mindestens einen Registers ($R_0, R_1, \dots$) übertragen werden soll.
3. System nach Anspruch 1 oder 2, wobei der Registerwechselbefehl ferner eine Angabe (R) enthält, ob der Inhalt eines der Register ($R_0, R_1, \dots$) der besagten einen Registergruppe auf eine andere Registergruppe übertragen werden soll.
4. System nach einem der Ansprüche 1 bis 3, wobei der Registerwechselbefehl eine Anzahl von Bits enthält, deren jedes einem Register ($R_0, R_1, \dots$) in einer Registergruppe ($BN_0, BN_1, \dots$) entspricht und angibt, ob der Inhalt des jeweiligen Registers zu übertragen ist.
5. System nach einem der Ansprüche 1 bis 4, wobei der Registerwechselbefehl ferner eine Information (Imm) zum Spezifizieren einer Banknummer der anderen Registergruppe enthält.

Revendications

1. Système de traitement de données pour le traitement d'une pluralité de tâches mettant chacune en oeuvre le contenu d'un premier groupe d'une pluralité de groupes de registres ($BN_0, BN_1, \dots$), dans lequel une commutation d'un groupe de registres à un autre groupe de registres est exécutée conformément à une instruction de changement de registre, caractérisé en ce que ladite instruction de changement de registre comprend une information (Rn-list(SOURCE)) servant à spécifier au moins un registre ($R_0, R_1, \dots$) dans ledit premier groupe de registres, et que le contenu situé dans ledit au moins un registre spécifié

nally with respect to the microprocessor 1. Moreover, in the description above, although the invention of the present inventor has been described in case where the invention is applied to a microprocessor, namely, a utilization field which is the background of the invention, it is not restricted by that utilization field, that is, the present invention can be used for a program controlled data processing system in general such as a computer and a mini-computer.

Claims

1. A data processing system for processing a plurality of tasks each involving the contents of one of a plurality of register groups ($BN_0, BN_1, \dots$), wherein a change from one register group to another register group is effected in accordance with a register change instruction, characterized in that said register change instruction includes information (Rn-list(SOURCE)) for specifying at least one register ($R_0, R_1, \dots$) in said one register group, and that the contents in the specified at least one register in said one register group are transferred to said other register group in response to said register change instruction.
2. The system of claim 1, wherein said register change instruction further includes information (Rn-list(DESTINATION)) for specifying at least one register ($RL_0, RL_1, \dots$) in said other register group into which the contents of said at least one register ($R_0, R_1, \dots$) are to be transferred.
3. The system of claim 1 or 2, wherein said register change instruction further includes an indication (R) as to whether the contents of any of the registers ($R_0, R_1, \dots$) of said one register group are to be transferred to another register group.
4. The system of any of claims 1 to 3, wherein said register change instruction contains a number of bits each corresponding to one register ($R_0, R_1, \dots$) in a register group ($BN_0, BN_1, \dots$) and indicating whether the contents of the respective register are to be transferred.
5. The system of any of claims 1 to 4, wherein said register change instruction further contains information (Imm) for specifying a bank number of the other register group.

Patentansprüche

1. Datenverarbeitungssystem zum Verarbeiten mehrerer Aufgaben, deren jede mit dem Inhalt

einer von mehreren Registergruppen ($BN_0, BN_1, \dots$) arbeitet, wobei gemäß einem Registerwechselbefehl ein Wechsel von einer Registergruppe zu einer anderen erfolgt,

dadurch gekennzeichnet, daß der Registerwechselbefehl Informationen (Rn-list(SOURCE)) zum Spezifizieren mindestens eines Registers ($R_0, R_1, \dots$) in der besagten einen Registergruppe enthält, und daß der Inhalt des spezifizierten mindestens einen Register in der besagten einen Registergruppe entsprechend dem Registerwechselbefehl auf die andere Registergruppe übertragen wird.

2. System nach Anspruch 1, wobei der Registerwechselbefehl ferner Informationen (Rn-list(DESTINATION)) zum Spezifizieren mindestens eines Registers ($RL_0, RL_1, \dots$) in der besagten anderen Registergruppe enthält, auf die der Inhalt des besagten mindestens einen Registers ($R_0, R_1, \dots$) übertragen werden soll.
3. System nach Anspruch 1 oder 2, wobei der Registerwechselbefehl ferner eine Angabe (R) enthält, ob der Inhalt eines der Register ($R_0, R_1, \dots$) der besagten einen Registergruppe auf eine andere Registergruppe übertragen werden soll.
4. System nach einem der Ansprüche 1 bis 3, wobei der Registerwechselbefehl eine Anzahl von Bits enthält, deren jedes einem Register ($R_0, R_1, \dots$) in einer Registergruppe ($BN_0, BN_1, \dots$) entspricht und angibt, ob der Inhalt des jeweiligen Registers zu übertragen ist.
5. System nach einem der Ansprüche 1 bis 4, wobei der Registerwechselbefehl ferner eine Information (Imm) zum Spezifizieren einer Banknummer der anderen Registergruppe enthält.

Revendications

1. Système de traitement de données pour le traitement d'une pluralité de tâches mettant chacune en oeuvre le contenu d'un premier groupe d'une pluralité de groupes de registres ($BN_0, BN_1, \dots$), dans lequel une commutation d'un groupe de registres à un autre groupe de registres est exécutée conformément à une instruction de changement de registre,

caractérisé en ce que ladite instruction de changement de registre comprend une information (Rn-list(SOURCE)) servant à spécifier au moins un registre ($R_0, R_1, \dots$) dans ledit premier groupe de registres, et que le contenu situé dans ledit au moins un registre spécifié



dans ledit premier groupe de registres est transféré dans ledit autre groupe de registres en réponse à ladite instruction de changement de registre.

2. Système selon la revendication 1, dans lequel ladite instruction de changement de registre comprend en outre une information (Rn-list-(DESTINATION)) servant à spécifier au moins un registre (RL₀, RL₁, ...) dans ledit autre groupe de registres dans lequel le contenu dudit au moins un registre (R₀, R₁, ...) doit être transféré.
3. Système selon la revendication 1 ou 2, dans lequel ladite instruction de changement de registre comprend en outre une indication (R) mentionnant si le contenu de l'un quelconque des registres (R₀, R₁, ...) dudit premier groupe de registres doit être transféré à un autre groupe de registres.
4. Système selon l'une quelconque des revendications 1 à 3, dans lequel ladite instruction de changement de registre contient un certain nombre de bits qui correspondent chacun à un registre (R₀, R₁, ...) dans un groupe de registres (BN₀, BN₁, ...) et indiquant si le contenu du registre respectif doit être transféré.
5. Système selon l'une quelconque des revendications 1 à 4, dans lequel ladite instruction de changement de registre contient en outre une information (Imm) servant à spécifier un numéro de bloc de l'autre groupe de registres.

5

10

15

20

25

30

35

40

45

50

55

