CLAIMS

1. A method for managing a dynamic address translation (DAT) table, the method comprising steps of:

determining, by the processor, from an opcode of a machine executable instruction to be executed, that the instruction is a compare and replace DAT table entry instruction, the instruction comprising an R_1 field and an R_2 field; and

executing, by the processor, the instruction, the executing comprising:

based on a type field, adding an index value of an odd register to a value of an even register to form an address of a second operand, the odd register and even register being an even-odd register pair designated by the R_2 field of the instruction;

obtaining an original second operand from a memory location specified by the formed address;

comparing a first operand and the obtained original second operand, the first operand contained in an even-numbered (even R_1) register of an even-odd register pair designated by the R_1 field of the instruction;

based on the first operand being equal to the original second operand, storing the contents of the even R_1 register in the second operand location as the new second operand.

2. The method of Claim 1, further comprising selectively clearing all local Translation Lookaside Buffer (TLB) entries corresponding to entries formed from DAT using translation table entries based on a translation table entry designated by a value of the original second operand.

3. The method of claim 1, comprising storing the original second operand into the even R_1 register based on the first operand being not equal to the original second operand.

4. The method of claim 2, wherein a translation table entry selectively cleared corresponds to lower level translation table entries associated with the original second operand.

5. The method according to claim 1, further comprising:

based on a first operand compare value being equal to the second operand, setting a first condition code value; and

based on a first operand compare value being not-equal to the second operand, setting a second condition code value.

6. The method of claim 2 wherein the TLB entries cleared are limited to entries formed to translate addresses in an address space specified by the instruction.

7. The method of claim 2, further comprising:

based on a mask field of the instruction having a first value, selectively clearing only a TLB of the processor executing the instruction of a plurality of processors of a configuration; and based on a mask field of the instruction having a second value, selectively clearing the TLB of a plurality of processors of the configuration.

8. A system comprising means adapted for carrying out all the steps of the method according to any preceding method claim.

9. A computer program comprising instructions for carrying out all the steps of the method according to any preceding method claim, when said computer program is executed on a computer system.

CLAIMS

1. A method for managing a dynamic address translation (DAT) table, the method comprising steps of:

determining, by the processor, from an opcode of a machine executable instruction to be executed, that the instruction is a compare and replace DAT table entry instruction, the instruction comprising an R_1 field and an R_2 field; and

executing, by the processor, the instruction, the executing comprising:

based on a type field, adding an index value of an odd register to a value of an even register to form an address of a second operand, the odd register and even register being an even-odd register pair designated by the R_2 field of the instruction;

obtaining an original second operand from a memory location specified by the formed address;

comparing a first operand and the obtained original second operand, the first operand contained in an even-numbered (even R_1) register of an even-odd register pair designated by the R_1 field of the instruction; and

based on the first operand being equal to the original second operand;

storing the contents of the odd-numbered (odd R_1) register of the even-odd register pair designated by the R_1 field of the instruction in the second operand location as the new second operand; and

selectively clearing all local translation lookaside buffer (TLB) entries corresponding to entries formed from DAT using translation table entries based on a translation table entry designated by a value of the original second operand.

2. The method of claim 1, comprising storing the original second operand into the even R_1 register based on the first operand being not equal to the original second operand.

3. The method of claim 1, wherein a translation table entry selectively cleared corresponds to lower level translation table entries associated with the original second operand.

4. The method according to claim 1, further comprising:

based on a first operand compare value being equal to the second operand, setting a first condition code value; and

based on a first operand compare value being not-equal to the second operand, setting a second condition code value.

5. The method of claim 1 wherein the TLB entries cleared are limited to entries formed to translate addresses in an address space specified by the instruction.

6. The method of claim 1, further comprising:

based on a mask field of the instruction having a first value, selectively clearing only a TLB of the processor executing the instruction of a plurality of processors of a configuration; and based on a mask field of the instruction having a second value, selectively clearing the TLB of a plurality of processors of the configuration.

7. A system comprising means adapted for carrying out all the steps of the method according to any preceding method claim.

8. A computer program comprising instructions for carrying out all the steps of the method according to any preceding method claim, when said computer program is executed on a computer system.