CLAIMS

1. A method for executing a machine instruction in a central processing unit comprising steps of:

obtaining, by a processor, a machine instruction for execution, the machine instruction being defined for computer execution according to a computer architecture, the machine instruction comprising:

at least one opcode field to provide an opcode, the opcode identifying a Vector Find Element Not Equal operation;

an extension field to be used in designating one or more registers;

a first register field combined with a first portion of the extension field to designate a first register, the first register comprising a first operand;

a second register field combined with a second portion of the extension field to designate a second register, the second register comprising a second operand;

a third register field combined with a third portion of the extension field to designate a third register, the third register comprising a third operand;

a mask field, the mask field comprising one or more controls to be used during execution of the machine instruction; and

executing the machine instruction, the execution comprising:

determining whether the mask field includes a zero element control set to indicate a search for a zero element;

based on the mask field including the zero element control set to indicate the search for a zero element, searching the second operand for a zero element, the searching providing a null index, the null index including one of an index of a zero element found in the search or an indication of no zero elements found;

comparing one or more elements of the second operand with one or more elements of the third operand for inequality, the comparing providing a compare index, the compare index including one of an index of an unequal element based on the comparing finding an unequal element or an indication of no inequality based on the comparing finding no unequal elements; and

providing a result, the result based on whether the search for zero element was performed, wherein the result includes one of:

based on not performing the search for zero element, the result includes the compare index; or

based on performing the search for zero element, the result includes one of the compare index or the null index.

2. The method of claim 1, wherein the result is for an element, the element being a zero element or an unequal element, and the method further comprises:

adjusting the result, the adjusting comprising performing at least one operation on the result to provide an adjusted result, the adjusted result comprising a byte index of a first byte of the element; and

storing the adjusted result in the first operand.

3. The method of claim 2, wherein the machine instruction further comprises another mask field, the another mask field including an element size control, the element size control specifying a size of elements in at least one of the first operand, the second operand, or the third operand, and wherein the size is used in the adjusting.

4. The method of claim 1, wherein the result includes a value indicating a size of the second operand, and the method further comprising storing the result in the first operand.

5. The method of claim 1, wherein the mask field comprises a condition code set control, and wherein the method comprises:

determining whether the condition code set control is set; and

based on the condition code set control being set, setting a condition code for execution of the machine instruction.

6. The method of claim 5, wherein the setting the condition code comprises one of:

setting the condition code to a value indicating detection of a zero element in a lower indexed element than any unequal compares;

setting the condition code to a value indicating a mismatched element, and the mismatched element of the second operand is less than the mismatched element of the third operand;

setting the condition code to a value indicating a mismatched element and the mismatched element of the second operand is greater than the mismatched element of the third operand; and

setting the condition code to a value indicating no mismatch, and based on the zero element control being set, no zero elements were found.

7. The method of claim 1, wherein the executing comprises determining, at runtime, a direction for the comparing, wherein the direction is one of left-to-right or right-to-left, and the determination comprises accessing by the machine instruction a direction control to determine the direction.

8. The method of claim 1, wherein the second operand and the third operand comprise N bytes, and wherein the comparing comprises comparing in parallel the N bytes of the second operand with the N bytes of the third operand, and wherein a size of an element comprises one of one byte, two bytes or four bytes.

9. The method of claim 1, wherein the index of the zero element comprises a byte index, the byte index being an index of a first byte of the zero element.

10. The method of claim 1, wherein the index of the unequal element comprises a byte index, the byte index being an index of a byte of the unequal element.

11. The method of claim 1, wherein at least one of the indication of no zero elements found or the indication of no inequality comprises including in at least one of the null index or the compare index a number representing a size of the second operand.

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

13. 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 executing a machine instruction in a central processing unit comprising the steps of:

obtaining, by a processor, a machine instruction for execution, the machine instruction being defined for computer execution according to a computer architecture, the machine instruction comprising:

at least one opcode field to provide an opcode, the opcode identifying a Vector Find Element Not Equal operation;

an extension field to be used in designating one or more registers;

a first register field to designate a first register, the first register comprising a first operand;

a second register field to designate a second register, the second register comprising a second operand;

a third register field to designate a third register, the third register comprising a third operand; and

a mask field, the mask field comprising one or more controls to be used during execution of the machine instruction; and

executing the machine instruction, the execution comprising:

identifying the first register based on a combination of the first register field and a first portion of the extension field, identifying the second register based on a combination of the second register field and a second portion of the extension field, and identifying the third register based on a combination of the third register field and a third portion of the extension field;

searching the second operand for a zero element, the searching providing a null index, the null index including one of an index of a zero element found in the search or an indication of no zero elements found;

comparing one or more elements of the second operand with one or more elements of the third operand for inequality, the comparing providing a compare index separate from the null index, the compare index including one of an index of an unequal element based on the comparing finding an unequal element or an indication of no inequality based on the comparing finding no unequal elements; and

selecting between the null index and the compare index to provide a result, the result being the null index or the compare index, and the selecting being based on whether an unequal element or zero element is found.

2. The method of claim 1, wherein the result is a byte index of an element, the element being a zero element or an unequal element, and the method further comprises:

adjusting the result, the adjusting comprising performing at least one operation on the result to provide an adjusted result, the adjusted result comprising an index of a first byte of the element; and

storing the adjusted result in the first operand.

3. The method of claim 2, wherein the machine instruction further comprises another mask field, the another mask field including an element size control, the element size control specifying a size of elements in at least one of the first operand, the second operand, or the third operand, and wherein the size is used in the adjusting.

4. The method of claim 1, wherein the result indicates a size of the second operand, and the method further comprises storing the result in the first operand.

5. The method of claim 1, wherein the mask field comprises a condition code set control, and wherein the method comprising:

determining whether the condition code set control is set; and

based on the condition code set control being set, setting a condition code for execution of the machine instruction.

6. The method of claim 5, wherein the setting the condition code comprises one of:

setting the condition code to a value indicating detection of a zero element in a lower indexed element than any unequal compares;

setting the condition code to a value indicating a mismatch between an element of the second operand and an element of the third operand, and the element of the second operand is less than the mismatched element of the third operand;

setting the condition code to a value indicating a mismatch between an element of the second operand and an element of the third operand, and the element of the second operand is greater than the element of the third operand; and

setting the condition code to a value indicating no mismatch, and no zero elements were found.

7. The method of claim 1, wherein the executing comprises determining, at runtime, a direction for the comparing, wherein the direction is one of left-to-right or right-to-left, and the determination comprises accessing by the machine instruction a direction control to determine the direction.

8. The method of claim 1, wherein the second operand and the third operand comprise N bytes, and wherein the comparing comprises comparing in parallel the N bytes of the second operand with the N bytes of the third operand, and wherein a size of an element comprises one of one byte, two bytes or four bytes.

9. The method of claim 1, wherein the index of the zero element comprises a byte index, the byte index being an index of a first byte of the zero element.

10. The method of claim 1, wherein the index of the unequal element comprises a byte index, the byte index being an index of a byte of the unequal element.

11. The method of claim 1, wherein at least one of the indication of no zero elements found or the indication of no inequality comprises including in at least one of the null index or the compare index a number representing a size of the second operand.

12. The method of claim 1, wherein the combination of the first register field and the first portion of the extension field comprises a combination of bits of the first register field and at least one bit of the first portion of the extension field to form an identifier to identify the first register, the at least one bit of the first portion of the extension field being specific to identification of the first register based on combining the at least one bit with the bits of the first register field.

13. The method of claim 1, wherein the selecting is based on either no unequal element being detected in a lower indexed element than a zero element or no zero element being detected in a lower indexed element than an unequal element, or the selecting is based on neither a zero element nor an unequal element being found.

14. The method of claim 1, wherein the selecting selects a lowest index of the null index or the compare index.

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

16. 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.