CLAIMS

1. A computer program product for transforming instruction specifiers of a computing environment, the computer program product comprising:

a computer readable storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method comprising:

obtaining, by a processor, from a first instruction defined for a first computer architecture, a non-contiguous specifier, the non-contiguous specifier having a first portion and a second portion, wherein the obtaining comprises obtaining the first portion from a first field of the instruction and the second portion from a second field of the instruction, the first field separate from the second field;

generating a contiguous specifier using the first portion and the second portion, the generating using one or more rules based on the opcode of the first instruction; and

using the contiguous specifier to indicate a resource to be used in execution of a second instruction, the second instruction defined for a second computer architecture different from the first computer architecture and emulating a function of the first instruction.

2. The computer program product of claim 1, wherein the processor comprises an emulator, and wherein the first portion includes a first one or more bits, and the second portion includes a second one or more bits, and the generating comprises concatenating the second one or more bits with the first one or more bits to form the contiguous specifier, wherein the second one or more bits are the most significant bits of the contiguous specifier.

3. The computer program product of claim 2, wherein the first field has an operand position associated therewith, and the second one or more bits are a subset of a plurality of bits of the second field, and wherein the obtaining comprises selecting the second one or more bits from the plurality of bits of the second field based on the operand position of the first field.

4. The computer program product of claim 3, wherein the operand position of the first field is as a first operand, and wherein the second one or more bits are selected from a left-most location of the second field.

5. The computer program product of any preceding claim, wherein the first field consists of a register field, the second field consists of an extension field, the first portion consists of a plurality of bits from the register field, the second portion consists of a bit from the extension field in a location of the instruction corresponding to the register field, and the generating comprises concatenating the bit from the extension field with the bits from the register field to provide the contiguous specifier.

6. The computer program product of any preceding claim, wherein the using the contiguous specifier to indicate a resource includes using the contiguous specifier to map to a register to be used by the second instruction.

7. The computer program product of claim 6, wherein the register mapped to by the contiguous specifier has the same value as the contiguous specifier.

8. The computer program product of claim 6, wherein the register mapped to by the contiguous specifier has a different value from the contiguous specifier.

9. The computer program product of any preceding claim, wherein the first computer architecture includes an instruction set comprising first instructions having register fields to access a sub-section of a register space of the first computer architecture, and having second instructions having non-contiguous register fields for accessing the sub-section and remaining subsections of the register space, the first instructions precluded from accessing the remaining subsections.

10. The computer program product of claim 1, wherein the first field consists of a register field, the second field consists of an extension field, the first portion consists of a plurality of bits from the register field, the second portion consists of a bit from the extension field in a location of the instruction corresponding to the register field, and the generating

comprises concatenating the bit from the extension field with the bits from the register field to provide the contiguous specifier, and further comprising:

obtaining, by the processor, from the first instruction, another non-contiguous specifier, the another non-contiguous specifier having another first portion and another second portion, wherein the obtaining comprises obtaining the another first portion from another first field of the instruction and the another second portion from another bit of the extension field, the another first field separate from the first field and the extension field;

generating another contiguous specifier using the another first portion and the another bit, the generating using one or more rules based on the opcode of the first instruction; and

using the another contiguous specifier to indicate a resource to be used in execution of the second instruction.

11. A computer system for transforming instruction specifiers of a computing environment, the computer system comprising:

a memory; and

a processor in communications with the memory, wherein the computer system is configured to perform a method, said method comprising:

obtaining, by a processor, from a first instruction defined for a first computer architecture, a non-contiguous specifier, the non-contiguous specifier having a first portion and a second portion, wherein the obtaining comprises obtaining the first portion from a first field of the instruction and the second portion from a second field of the instruction, the first field separate from the second field;

generating a contiguous specifier using the first portion and the second portion, the generating using one or more rules based on the opcode of the first instruction; and

using the contiguous specifier to indicate a resource to be used in execution of a second instruction, the second instruction defined for a second computer architecture different from the first computer architecture and emulating a function of the first instruction.

12. The computer system of claim 11, wherein the processor comprises an emulator, and wherein the first portion includes a first one or more bits, and the second portion includes a

second one or more bits, and the generating comprises concatenating the second one or more bits with the first one or more bits to form the contiguous specifier, wherein the second one or more bits are the most significant bits of the contiguous specifier.

13. The computer system of claim 12, wherein the first field has an operand position associated therewith, and the second one or more bits are a subset of a plurality of bits of the second field, and wherein the obtaining comprises selecting the second one or more bits from the plurality of bits of the second field based on the operand position of the first field.

14. The computer system of claim 13, wherein the operand position of the first field is as a first operand, and wherein the second one or more bits are selected from a left-most location of the second field.

15. The computer system of any of claims 11 to 14, wherein the first field comprises a register field, the second field comprises an extension field, the first portion comprises a plurality of bits from the register field, the second portion comprises a bit from the extension field in a location corresponding to the register field, and the generating comprises concatenating the bit from the extension field with the bits from the register field to provide the contiguous specifier.

16. The computer system of any of claims 11 to 15, wherein the using the contiguous specifier to indicate a resource includes using the contiguous specifier to map to a register to be used by the second instruction.

17. The computer system of claim 16, wherein the register mapped to by the contiguous specifier has one of: the same value as the contiguous specifier or has a different value from the contiguous specifier.

18. A method of transforming instruction specifiers of a computing environment, the method comprising:

obtaining, by a processor, from a first instruction defined for a first computer architecture, a non-contiguous specifier, the non-contiguous specifier having a first portion

and a second portion, wherein the obtaining comprises obtaining the first portion from a first field of the instruction and the second portion from a second field of the instruction, the first field separate from the second field;

generating a contiguous specifier using the first portion and the second portion, the generating using one or more rules based on the opcode of the first instruction; and

using the contiguous specifier to indicate a resource to be used in execution of a second instruction, the second instruction defined for a second computer architecture different from the first computer architecture and emulating a function of the first instruction.

19. The method of claim 18, wherein the processor comprises an emulator, and wherein the first portion includes a first one or more bits, and the second portion includes a second one or more bits, and the generating comprises concatenating the second one or more bits with the first one or more bits to form the contiguous specifier, wherein the second one or more bits are the most significant bits of the contiguous specifier.

20. The method of claim 18 or claim 19, wherein the first field comprises a register field, the second field comprises an extension field, the first portion comprises a plurality of bits from the register field, the second portion comprises a bit from the extension field in a location corresponding to the register field, and the generating comprises concatenating the bit from the extension field with the bits from the register field to provide the contiguous specifier.

CLAIMS

1. A computer program product for emulating instructions in a computing environment, the computer program product comprising:

a computer readable storage medium readable by a processing circuit and storing instructions for execution by the processing circuit for performing a method comprising:

determining from a first instruction defined for a first computer architecture that the first instruction includes a non-contiguous register operand specifier having a first portion and a second portion, non-contiguous with the first portion;

obtaining the non-contiguous register operand specifier from the first instruction comprising obtaining the first portion from a first field of the first instruction and the second portion from a second field of the first instruction, the first field separated from the second field by at least one intervening field, wherein a portion of the opcode of the first instruction specifies the first field and the second field used to designate the non-contiguous specifier;

generating a contiguous register operand specifier using the first portion and the second portion obtained from the first instruction, the generating using one or more rules depending on the format of the instruction specified by the opcode of the first instruction;

using the contiguous register operand specifier in place of the non-contiguous register operand specifier to indicate a resource to be used in execution of a second instruction, the second instruction defined for a second computer architecture different from the first computer architecture and emulating a function of the first instruction; and

executing the second instruction to emulate the function of the first instruction, the executing using the resource indicated by the contiguous register operand specifier without further regard to the non-contiguous register operand specifier.

2. The computer program product of claim 1, wherein the first portion includes a first one or more bits, and the second portion includes a second one or more bits, and the generating comprises concatenating the second one or more bits with the first one or more bits to form the contiguous register operand specifier, wherein the second one or more bits are the most significant bits of the contiguous register operand specifier.

3. The computer program product of claim 2, wherein the first field has an operand

position associated therewith, and the second one or more bits are a subset of a plurality of bits of the second field, and wherein the obtaining comprises selecting the second one or more bits from the plurality of bits of the second field based on the operand position of the first field.

4. The computer program product of claim 3, wherein the operand position of the first field is as a first operand, and wherein the second one or more bits are selected from a left- most location of the second field.

5. The computer program product of any one of the preceding claims, wherein the first field consists of a register field, the second field consists of an extension field, the first portion consists of a plurality of bits from the register field, the second portion consists of a bit from the extension field in a location of the instruction corresponding to the register field, and the generating comprises concatenating the bit from the extension field with the bits from the register field to provide the contiguous register operand specifier.

6. The computer program product of any one of the preceding claims, wherein the using the contiguous register operand specifier to indicate a resource includes using the contiguous register operand specifier to map to a register to be used by the second instruction.

7. The computer program product of claim 6, wherein the register mapped to by the contiguous register operand specifier has the same value as the contiguous register operand specifier.

8. The computer program product of claim 6, wherein the register mapped to by the contiguous register operand specifier has a different value from the contiguous register operand specifier.

9. The computer program product of any one of the preceding claims, wherein the first computer architecture includes an instruction set comprising first instructions having register fields to access a sub-section of a register space of the first computer architecture, and having second instructions having non-contiguous register fields for accessing the sub-section and remaining subsections of the register space, the first instructions precluded from accessing

the remaining subsections.

10. The computer program product of claim 1, wherein the first field consists of a register field, the second field consists of an extension field, the first portion consists of a plurality of bits from the register field, the second portion consists of a bit from the extension field in a location of the instruction corresponding to the register field, and the generating comprises concatenating the bit from the extension field with the bits from the register field to provide the contiguous register operand specifier, and further comprising:

obtaining, by the processor, from the first instruction, another non-contiguous register operand specifier, the another non-contiguous register operand specifier having another first portion and another second portion, wherein the obtaining comprises obtaining the another first portion from another first field of the instruction and the another second portion from another bit of the extension field, the another first field separate from the first field and the extension field;

generating another contiguous register operand specifier using the another first portion and the another bit, the generating using one or more rules based on the opcode of the first instruction; and

using the another contiguous register operand specifier to indicate a resource to be used in execution of the second instruction.

11. A computer system for emulating instructions in a computing environment, the computer system comprising:

a memory; and

a processor in communications with the memory, wherein the computer system is configured to perform a method, said method comprising:

determining from a first instruction defined for a first computer architecture that the first instruction includes a non-contiguous register operand specifier having a first portion and a second portion, non-contiguous with the first portion;

obtaining the non-contiguous register operand specifier from the first instruction comprising obtaining the first portion from a first field of the first instruction and the second portion from a second field of the first instruction, the first field separated from the second field by at least one intervening field, wherein a portion of the opcode of the first instruction specifies the first field and the second field used to designate the non-contiguous specifier;

generating a contiguous register operand specifier using the first portion and the second portion obtained from the first instruction, the generating using one or more rules depending on the format of the instruction specified by the opcode of the first instruction;

using the contiguous register operand specifier in place of the non-contiguous register operand specifier to indicate a resource to be used in execution of a second instruction, the second instruction defined for a second computer architecture different from the first computer architecture and emulating a function of the first instruction; and

executing the second instruction to emulate the function of the first instruction, the executing using the resource indicated by the contiguous register operand specifier without further regard to the non-contiguous register operand specifier.

12. The computer system of claim 11, wherein the first portion includes a first one or more bits, and the second portion includes a second one or more bits, and the generating comprises concatenating the second one or more bits with the first one or more bits to form the contiguous register operand specifier, wherein the second one or more bits are the most significant bits of the contiguous register operand specifier.

13. The computer system of claim 12, wherein the first field has an operand position associated therewith, and the second one or more bits are a subset of a plurality of bits of the second field, and wherein the obtaining comprises selecting the second one or more bits from the plurality of bits of the second field based on the operand position of the first field.

14. The computer system of claim 13, wherein the operand position of the first field is as a first operand, and wherein the second one or more bits are selected from a left-most location of the second field.

15. The computer system of any one of claims 11 to 14, wherein the first field comprises a register field, the second field comprises an extension field, the first portion comprises a plurality of bits from the register field, the second portion comprises a bit from the extension field in a location corresponding to the register field, and the generating comprises concatenating the bit from the extension field with the bits from the register field to provide the contiguous register operand specifier.

16. The computer system of any one of claims 11 to 15, wherein the using the

contiguous register operand specifier to indicate a resource includes using the contiguous register operand specifier to map to a register to be used by the second instruction.

17. The computer system of claim 16, wherein the register mapped to by the contiguous register operand specifier has one of: the same value as the contiguous register operand specifier or has a different value from the contiguous register operand specifier.

18. A method of emulating instructions in a computing environment, the method compnsmg:

determining from a first instruction defined for a first computer architecture that the first instruction includes a non-contiguous register operand specifier having a first portion and a second portion, con-contiguous with the first portion;

obtaining the non-contiguous register operand specifier from the first instruction comprising obtaining the first portion from a first field of the first instruction and the second portion from a second field of the first instruction, the first field separated from the second field by at least one intervening field, wherein a portion of the opcode of the first instruction specifies the first field and the second field used to designate the non-contiguous specifier;

generating a contiguous register operand specifier using the first portion and the second portion obtained from the first instruction, the generating using one or more rules depending on the format of the instruction specified by the opcode of the first instruction;

using the contiguous register operand specifier in place of the non-contiguous register operand specifier to indicate a resource to be used in execution of a second instruction, the second instruction defined for a second computer architecture different from the first computer architecture and emulating a function of the first instruction; and

executing the second instruction to emulate the function of the first instruction, the executing using the resource indicated by the contiguous register operand specifier without further regard to the non-contiguous register operand specifier.

19. The method of claim 18, wherein the first portion includes a first one or more bits, and the second portion includes a second one or more bits, and the generating comprises concatenating the second one or more bits with the first one or more bits to form the contiguous register operand specifier, wherein the second one or more bits are the most

significant bits of the contiguous register operand specifier.

20. The method of claim 18 or claim 19, wherein the first field comprises a register field, the second field comprises an extension field, the first portion comprises a plurality of bits from the register field, the second portion comprises a bit from the extension field in a location corresponding to the register field, and the generating comprises concatenating the bit from the extension field with the bits from the register field to provide the contiguous register operand specifier.