ACCESSING DATA IN MULTI-DIMENSIONAL TENSORS

BACKGROUND

[0001] This specification generally relates to using registers for computing memory address values. Registers may be used to handle branch conditions.

SUMMARY

[0001a] According to an aspect, the present disclosure refers to an apparatus for processing a multi-dimensional tensor input, the apparatus comprising: multiple tensor index elements including a tensor index element for each of multiple nested loops used to traverse an N-dimensional tensor, wherein each tensor index element is a respective first hardware register configured to store an index value for a respective nested loop that is used to traverse a respective dimension of the N-dimensional tensor, and wherein the N-dimensional tensor includes data elements arranged across each of the N-dimensions and N is an integer that is equal to or greater than one, wherein each index value is a value that is updated each time an iteration of the respective nested loop for the index value is performed; multiple dimension multiplier elements, wherein each dimension multiplier element is a respective second hardware register configured to store a multiplier value for a respective dimension of the N-dimensional tensor, wherein each multiplier value is a constant value for the respective dimension of the N-dimensional tensor for the multiplier value and remains constant for each iteration of the respective nested loop that is used to traverse the respective dimension; and one or more hardware processors configured to execute one or more instructions of an instruction set executable by the one or more hardware processors, wherein execution of the one or more instructions causes the one or more hardware processors to perform operations comprising: determining memory addresses for locations in memory for storing data values for a sequence of data elements of the N-dimensional tensor by: for each iteration of an inner loop of the nested loops: determining, for each dimension of the N-dimensional tensor, a product of (i) the index value stored in the tensor index element for the nested loop that is used to traverse the dimension and (ii) the multiplier

value for the dimension; and determining a memory address for a data element that corresponds to the iteration of the inner loop based on a sum of the product for each dimension of the N-dimensional tensor; and outputting data indicating the determined memory address for each data element in the sequence of data elements of the N-dimensional tensor.

[0001b] According to another aspect, the present disclosure refers to a system comprising: one or more hardware processors configured to perform linear operations on a N-dimensional tensor, wherein the N-dimensional tensor has data elements arranged across each of the N dimensions, and wherein N is an integer that is equal to or greater than one; multiple tensor index elements including a tensor index element for each of multiple nested loops used to traverse the N-dimensional tensor, wherein each tensor index element is a respective first hardware register configured to store an index value for a respective nested loop that is used to traverse a respective dimension of the N-dimensional tensor, and wherein each index value is a value that is updated each time an iteration of the respective nested loop for the index value is performed; multiple dimension multiplier elements, wherein each dimension multiplier element is a respective second hardware register configured to store a multiplier value for a respective dimension of the N-dimensional tensor, wherein each multiplier value is a constant value for the respective dimension of the N-dimensional tensor for the multiplier value and remains constant for each iteration of the respective nested loop that is used to traverse the respective dimension; and hardware circuitry configured to: determine memory addresses for a sequence of data elements of the N-dimensional tensor by: for each iteration of an inner loop of the nested loops: determining, for each dimension of the N-dimensional tensor, a product of (i) the index value stored in the tensor index element for the nested loop that is used to traverse the dimension and (ii) the multiplier value for the dimension; and determining a memory address for a data element that corresponds to the iteration of the inner loop based on a sum of the product for each dimension of the N—dimensional tensor; and output data indicating the determined memory address for each data element in the sequence of data elements of the N-dimensional tensor

[0002] According to one ~~innovative aspect~~example of the subject matter described in this specification, registers can be used to iterate nested loop structures, to compute corresponding memory addresses, and to allow programmers to handle loop bound conditions. In general, registers may be arranged to maintain several sets of architectural status, including loop bound values, a current loop index variable in each loop, dimension multipliers for computing a memory address value, and program counter values for handling branch loop bounds. The arrangement may be good for processing one or more multi-dimensional data structures. With these architectural status sets, a processor may execute instructions to move index variables to new positions, retrieve memory address values, and to jump to a new branch target.

[0003] In general, one example ~~innovative aspect~~ of the subject matter described in this specification can be embodied in an apparatus for processing an instruction for accessing a *N*-dimensional tensor. The apparatus includes multiple tensor index elements and multiple dimension multiplier elements, where each of the dimension multiplier elements has a corresponding tensor index element. The apparatus includes one or more processors configured to obtain an instruction to access a particular element of a *N*-dimensional tensor, where the *N*-dimensional tensor has multiple elements arranged across each of the *N* dimensions, and where *N* is an integer that is equal to or greater than one; determine, using one or more tensor index elements of the multiple tensor index elements and one or more dimension multiplier elements of the multiple dimension multiplier elements, an address of the particular element; and output data indicating the determined address for accessing the particular element of the *N*-dimensional tensor.

CLAIMS

1. An apparatus for processing ~~an instruction for accessing~~ a ~~*N*~~multi-dimensional tensor input, the apparatus comprising:

~~multiple tensor index elements;~~

multiple tensor index elements including a tensor index element for each of multiple nested loops used to traverse an N-dimensional tensor, wherein each tensor index element is a respective first hardware register configured to store an index value for a respective nested loop that is used to traverse a respective dimension of the N-dimensional tensor, and wherein the N-dimensional tensor includes data elements arranged across each of the N-dimensions and N is an integer that is equal to or greater than one, wherein each index value is a value that is updated each time an iteration of the respective nested loop for the index value is performed;

multiple dimension multiplier elements, wherein each ~~of the~~ dimension multiplier ~~elements has a corresponding tensor index element~~element is a respective second hardware register configured to store a multiplier value for a respective dimension of the N-dimensional tensor, wherein each multiplier value is a constant value for the respective dimension of the N-dimensional tensor for the multiplier value and remains constant for each iteration of the respective nested loop that is used to traverse the respective dimension; and

one or more hardware processors configured to~~:~~

~~obtain an instruction to access a particular element of a *N*-dimensional tensor, wherein the *N*-dimensional tensor has multiple elements arranged across each of the *N* dimensions, and wherein *N* is an integer that is equal to or greater than one;~~

~~determine, using one or more tensor index elements of the multiple tensor index elements and~~ execute one or more instructions of an instruction set executable by the one or more ~~dimension multiplier elements of the multiple dimension multiplier elements, an address of the particular element; and~~hardware processors, wherein execution of the one or more instructions causes the one or more hardware processors to perform operations comprising:

~~output~~ determining memory addresses for locations in memory for storing data values for a sequence of data elements of the N-dimensional tensor by:

for each iteration of an inner loop of the nested loops:

determining, for each dimension of the N-dimensional tensor, a product of (i) the index value stored in the tensor index element for the nested loop that is used to traverse the dimension and (ii) the multiplier value for the dimension; and

determining a memory address for a data element that corresponds to the iteration of the inner loop based on a sum of the product for each dimension of the N-dimensional tensor; and

outputting data indicating the determined memory address for ~~accessing the particular~~each data element in the sequence of data elements of the *N*-dimensional tensor.

2. The apparatus of claim 1,

wherein determining the memory address ~~of the particular~~for a data element ~~is an address offset from another element~~ that corresponds to the iteration of the inner loop based on a sum of the product for each dimension of the *N*-dimensional tensor~~,~~ ~~and~~

~~wherein determining the address of the particular element~~ comprises~~:~~ ~~for each tensor index element~~ determining a sum of (i) the ~~one or more tensor index elements, multiplying a value stored in the respective tensor index element by a value stored in the corresponding tensor index element~~sum of the ~~one or more multiple tensor index elements to generate a respective multiplied value; and~~product for each dimension of the N-dimensional tensor and (ii) a base memory address.

~~determining the address offset by determining a sum of the multiplied values.~~

3. The apparatus of claim 1,

wherein ~~the multiple tensor index elements are arranged into *M* groups of tensor index elements, *M* being a positive integer greater than one, and~~

~~wherein~~outputting data indicating the determined memory address for each ~~group of tensor index elements is assigned to a distinct and different multi-dimensional tensor~~data element comprises sequentially outputting data indicating the determined memory address for each data element in sequence as the memory addresses are determined.

4. The apparatus of claim 1, wherein the one or more processors are further configured to:

increment the index value for the inner loop by a first incremental value each time the inner loop completes; and

increment an index value for a second loop in which the inner loop is nested by a second incremental value each time the second loop completes.

5. The apparatus of claim ~~further comprising multiple~~1, wherein the one or more processors are further configured to:

receive an instruction to update the index value for the inner loop;

after receiving the instruction to update the index value for the inner loop, determining that a difference between the index value for the inner loop and a tensor bound ~~elements, wherein each of the multiple~~ value stored in a tensor bound ~~elements has a corresponding tensor index~~ element ~~in~~for the ~~multiple tensor~~inner loop satisfies a threshold and in response:

increment the index ~~elements.~~

value for the inner loop by a first incremental value.

~~5. The apparatus of claim 4, wherein the instruction represents an instruction for processing a nested loop that includes a first loop and a second loop that is an outer loop of the first loop, wherein the first loop is iterated using a first index variable, and wherein the second loop is iterated using a second index variable.~~

6. The apparatus of claim ~~5, wherein the one or more processors are further configured to:~~

~~set values stored in the multiple tensor index elements, the multiple dimension multiplier elements, and the multiple tensor bound elements to an initialization value.~~

~~7. The apparatus of claim 5, wherein the one or more processors are further configured to:~~

~~set a value stored in a first tensor bound element of the multiple tensor bound elements to an upper bound value of the first index variable for iterating the first loop;~~

~~set a value stored in a second tensor bound element of the multiple tensor bound elements to an upper bound value of the second index variable for iterating the second loop;~~

~~set a value stored in a first dimension multiplier element of the multiple dimension multiplier elements to a predetermined value; and~~

~~set a value stored in a second dimension multiplier element of the multiple dimension multiplier elements to the upper bound value of the first index variable.~~

~~8. The apparatus of claim 7, wherein determining the address of the element of the *N*-dimensional tensor comprises:~~

~~setting a value stored in a first tensor index element of the multiple tensor index elements to a current value of the first index variable;~~

~~setting a value stored in a second tensor index element of the multiple tensor index elements to a current value of the second index variable;~~

~~multiplying the value stored in the first dimension multiplier element by the value stored in the first tensor index element to obtain a first multiplied value;~~

~~multiplying the value stored in the second dimension multiplier element by the value stored in the second tensor index element to obtain a second multiplied value; and~~

~~determining the address by at least summing the first multiplied value and the second multiplied value~~1~~.~~

~~9. The apparatus of claim 8~~, wherein the one or more processors are further configured to:

receive an instruction to update the index value ~~stored in~~for the ~~first tensor index element~~inner loop;

after receiving the instruction to update the index value ~~stored in~~for the ~~first tensor index element~~inner loop, determining that a difference between the index value ~~stored in~~for the ~~first tensor index element~~inner loop and ~~the~~ a tensor bound value stored in ~~the first tensor bound element satisfies a threshold; and~~

~~in response to determining that the difference between the value stored in the first tensor index element and the value stored in the first~~a tensor bound element ~~satisfies the threshold, updating the value stored in the first tensor index element~~for the inner loop~~.~~

~~10. The apparatus of claim 9,~~

~~wherein the first index variable is incremented by a first incremental value each time the first loop completes, and~~

~~wherein updating the value stored in the first tensor index element comprises incrementing the value stored in the first tensor index element by the first incremental value.~~

~~11. The apparatus of claim 8, wherein the one or more processors are further configured to:~~

~~receive an instruction to update the value stored in the first tensor index element; after receiving the instruction to update the value stored in the first tensor index element, determining that a difference between the value stored in the first tensor index element and the value stored in the first tensor bound element~~ does not satisfy a threshold~~;~~ and in response:

~~in response to determining that the difference between the value stored in the first tensor index element and the value stored in the first tensor bound element does not satisfy the threshold:~~

~~reset the value stored in the first tensor index element; and~~

~~update the value stored in the~~ reset the index value for the inner loop to an initial value for the inner loop; and

increment an index value for a second ~~tensor index element.~~

~~12. The apparatus of claim 11,~~

~~wherein~~loop in which the ~~second index variable~~inner loop is ~~incremented~~nested by a ~~second incremental value each time the second loop completes, and wherein updating the value stored in the second tensor index element comprises incrementing the value stored in the second tensor index element by the~~ second incremental value.

~~13~~7. The apparatus of claim 1, wherein the one or more processors include one or more arithmetic logic units.

~~14. The apparatus of claim 1, wherein each of the tensor index elements and the dimension multiplier elements is a register~~8~~.~~

~~15. The apparatus of claim 1, further comprising multiple branch target elements, wherein each of the multiple branch target elements has a corresponding tensor index element in the multiple tensor index elements, and wherein each of the multiple branch target elements is configured to store an address for an instruction to be executed upon a completion of an iteration of a loop that corresponds to the respective branch target element.~~

~~16~~. A system comprising:

one or more hardware processors configured to perform linear ~~algebra~~ operations on a *N*-dimensional tensor, wherein the *N*-dimensional tensor has ~~multiple elements arranged across each of the *N* dimensions, and wherein *N* is an integer that is equal to or greater than one;~~

~~multiple tensor index elements;~~

~~multiple dimension multiplier elements, wherein each of the dimension multiplier elements has a corresponding tensor index element; and~~

~~circuitry configured to:~~

~~obtain an instruction to access a particular element of the *N*-dimensional tensor;~~

~~determine, using one or more tensor index elements of the multiple tensor index elements and one or more dimension multiplier elements of the multiple dimension multiplier elements, an address of the particular element; and~~

~~output, to the one or more processors, data indicating the determined address for accessing the particular element of the *N*-dimensional tensor~~data~~.~~

~~17. The system of claim 16,~~

~~wherein the address of the particular element is an address offset from another element of the *N*-dimensional tensor, and~~

~~wherein determining the address of the particular element comprises:~~

~~for each tensor index element of the one or more tensor index elements, multiplying a value stored in the respective tensor index element by a value stored in the corresponding tensor index element of the one or more multiple tensor index elements to generate a respective multiplied value; and~~

~~determining the address offset by determining a sum of the multiplied values.~~

~~18. A computer-implemented method, comprising:~~

~~obtaining an instruction to access a particular element of a *N*-dimensional tensor, wherein the *N*-dimensional tensor has multiple~~ elements arranged across each of the *N* dimensions, and wherein *N* is an integer that is equal to or greater than one;

~~determining, using one or more tensor index elements of~~ multiple tensor index elements ~~and one or more dimension multiplier elements of multiple dimension multiplier elements, an address of the particular~~ including a tensor index element~~; and outputting data indicating the determined address~~ for ~~accessing the particular element~~each of multiple nested loops used to traverse the *N*-dimensional tensor~~.~~

~~19. The method of claim 18,~~

wherein ~~the address of the particular element is an address offset from another element of the *N*-dimensional tensor, and~~

wherein determining the address of the particular element comprises:

for each tensor index element of the one or more tensor index elements, multiplying a value stored in the is a respective tensor index element by a value stored in the corresponding tensor index element of the one or more multiple tensor index elements to generate first hardware register configured to store an index value for a respective multiplied value; and

determining the address offset by determining a sum of the multiplied values.

20. The method of claim 18, wherein the instruction represents an instruction for processing a nested loop that includes a first is used to traverse a respective dimension of the N-dimensional tensor, and wherein each index value is a value that is updated each time an iteration of the respective nested loop and a for the index value is performed;

multiple dimension multiplier elements, wherein each dimension multiplier element is a respective second hardware register configured to store a multiplier value for a respective dimension of the N-dimensional tensor, wherein each multiplier value is a constant value for the respective dimension of the N-dimensional tensor for the multiplier value and remains constant for each iteration of the respective nested loop that is an outerused to traverse the respective dimension; and

hardware circuitry configured to:

determine memory addresses for a sequence of data elements of the N-dimensional tensor by:

for each iteration of an inner loop of the first loop, wherein the first loop is iterated using a first nested loops:

determining, for each dimension of the N-dimensional tensor, a product of (i) the index value stored in the tensor index variable,element for the nested loop that is used to traverse the dimension and (ii) the multiplier value for the dimension; and

determining a memory address for a data element that corresponds to the iteration of the inner loop based on a sum of the product for each dimension of the N—dimensional tensor; and

output data indicating the determined memory address for each data element in the sequence of data elements of the N-dimensional tensor.

9. The system of claim 8, wherein determining the memory address for a data element that corresponds to the iteration of the inner loop based on a sum of the product for each dimension of the N-dimensional tensor comprises determining a sum of (i) the sum of the product for each dimension of the N-dimensional tensor and (ii) a base memory address.

10. The ~~the~~ system of claim 8, wherein outputting data indicating the determined memory address for each data element comprises sequentially outputting data indicating the determined memory address for each data element in sequence as the memory addresses are determined.

11. The system of claim 8, wherein the circuitry is further configured to:

increment the index value for the inner loop by a first incremental value each time the inner loop completes; and

increment an index value for a second loop ~~is iterated using~~in which the inner loop is nested by a second ~~index variable~~incremental value each time the second loop completes.

12. The system of claim 8, wherein the circuitry is further configured to:

receive an instruction to update the index value for the inner loop;

after receiving the instruction to update the index value for the inner loop, determining that a difference between the index value for the inner loop and a tensor bound value stored in a tensor bound element for the inner loop satisfies a threshold and in response:

increment the index value for the inner loop by a first incremental value.

13. The system of claim 8, wherein the circuitry is further configured to:

receive an instruction to update the index value for the inner loop;

after receiving the instruction to update the index value for the inner loop, determining that a difference between the index value for the inner loop and a tensor bound value stored in a tensor bound element for the inner loop does not satisfy a threshold and in response:

reset the index value for the inner loop to an initial value for the inner loop; and

increment an index value for a second loop in which the inner loop is nested by a second incremental value.

14. The system of claim 8, wherein the one or more processors include one or more arithmetic logic units.

15. A method performed by data processing apparatus comprising one or more hardware processors that execute one or more instructions of an instruction set executable by the one or more hardware processors, the method comprising:

determining, by the one or more hardware processors, memory addresses for a sequence of data elements of an N-dimensional tensor, wherein the N-dimensional tensor includes data elements arranged across each of the N-dimensions and N is an integer that is equal to or greater than one, the determining comprising:

for each iteration of an inner loop of multiple nested loops used to traverse the N-dimensional tensor:

determining, by the one or more hardware processors and for each dimension of the N-dimensional tensor, a product of (i) an index value stored in an tensor index element for the nested loop that is used to traverse the dimension and (ii) a multiplier value for the dimension that is stored in a dimension multiplier element for the dimension, wherein:

each nested loop has a corresponding index value stored in a corresponding tensor index element, wherein each tensor index element is a respective first hardware register and each index value is a value that is updated each time an iteration of the nested loop corresponding to the index value is performed; and

each nested loop has a corresponding multiplier value stored in a corresponding dimension multiplier element, wherein each dimension multiplier element is a respective second hardware register and each multiplier value is a constant value for the dimension for the multiplier value that remains constant for each iteration of the nested loop corresponding to the multiplier value; and

determining, by the one or more hardware processors, a memory address for a data element that corresponds to the iteration of the inner loop based on a sum of the product for each dimension of the N-dimensional tensor; and

outputting, by the one or more hardware processors, data indicating the determined memory address for each data element in the sequence of data elements of the N-dimensional tensor.

16. The method of claim 15, wherein determining the memory address for a data element that corresponds to the iteration of the inner loop based on a sum of the product for each dimension of the N-dimensional tensor comprises determining a sum of (i) the sum of the product for each dimension of the N-dimensional tensor and (ii) a base memory address.

17. The method of claim 15, wherein outputting data indicating the determined memory address for each data element comprises sequentially outputting data indicating the determined memory address for each data element in sequence as the memory addresses are determined.

18. The method of claim 15, further comprising:

incrementing the index value for the inner loop by a first incremental value each time the inner loop completes; and

incrementing an index value for a second loop in which the inner loop is nested by a second incremental value each time the second loop completes.

19. The method of claim 15, further comprising:

receiving an instruction to update the index value for the inner loop;

after receiving the instruction to update the index value for the inner loop, determining that a difference between the index value for the inner loop and a tensor bound value stored in a tensor bound element for the inner loop satisfies a threshold and in response:

incrementing the index value for the inner loop by a first incremental value.

20. The method of claim 15, further comprising:

receiving an instruction to update the index value for the inner loop;

after receiving the instruction to update the index value for the inner loop, determining that a difference between the index value for the inner loop and a tensor bound value stored in a tensor bound element for the inner loop does not satisfy a threshold and in response:

resetting the index value for the inner loop to an initial value for the inner loop; and

incrementing an index value for a second loop in which the inner loop is nested by a second incremental value.