CLAIMS

What is claimed is:

1. A general purpose graphics processor comprising:

a compute cluster including multiple processing resources coupled with a cache memory, at least one processing resource including a matrix accelerator, the matrix accelerator configured to perform a matrix multiply operation on multiple elements of a first matrix and a second matrix, wherein the first matrix is a sparse matrix, wherein elements of the first matrix are compressed into a first compressed representation, and wherein the first compressed representation includes a set of non-zero value elements and metadata to identify the non-zero value elements;

wherein the first compressed representation is to be stored to the cache memory in a second compressed representation that includes a compressed set of non-zero value elements; and

wherein the at least one processing resource is configured to:

load the second compressed representation from the cache memory and store the second compressed representation to a memory within the at least one processing resource in the first compressed representation;

load selected elements of the second matrix from the cache memory and store the selected elements to the memory within the at least one processing resource, wherein the selected elements of the second matrix correspond with non-zero value elements of the first matrix stored within the first compressed representation and are selected based on the metadata for the non-zero value elements;

perform the matrix multiply operation on elements from the first compressed representation and the selected elements of the second matrix; and

write output of the matrix multiply operation to the memory within the at least one processing resource.

2. The general purpose graphics processor as in claim 1, wherein the cache memory is a level two cache memory.

3. The general purpose graphics processor as in claim 2, wherein a compressor is to compress data stored to the level two cache memory.

4. The general purpose graphics processor as in claim 1, wherein the memory within the processing resource includes a level one cache memory

5. The general purpose graphics processor as in claim 1, wherein the memory within the processing resource includes a register file.

6. The general purpose graphics processor as in claim 1, wherein the memory within the processing resource includes a memory within the matrix accelerator.

7. The general purpose graphics processor as in claim 1, wherein the first matrix includes weight data associated with a neural network.

8. The general-purpose graphics processor as in claim 7, wherein the second matrix includes input activation data associated with the neural network.

9. The general-purpose graphics processor as in claim 8, wherein the output of the matrix multiply operation includes output activation data associated with the neural network.

10. The general-purpose graphics processor as in claim 9, wherein the output of the matrix multiply operation is a dense matrix.

11. The general-purpose graphics processor as in claim 1, wherein the matrix accelerator includes a systolic array of processing elements.

12. The general-purpose graphics processor as in claim 1, wherein the cache memory couples with a compressor to compress the first compressed representation into the second compressed representation and a decompressor to decompress the second compressed representation into the first compressed representation.

13. A method comprising:

on a general-purpose graphics processor:

performing a matrix multiply operation on multiple elements of a first matrix and a second matrix via a compute cluster including multiple processing resources coupled with a cache memory, at least one processing resource including a matrix accelerator, wherein the first matrix is a sparse matrix, wherein elements of the first matrix are encoded into a sparse encoding, and wherein the sparse encoding includes a set of non-zero value elements and metadata to identify a location for the non-zero value elements;

storing the sparse encoding to the cache memory, the cache memory coupled with a compressor to compress the sparse encoding within the cache memory into a compressed representation of the sparse encoding that includes a compressed version of the set of non-zero value elements; and

via the at least one processing resource:

loading the compressed representation of the sparse encoding from the cache memory;

decompressing the compressed representation of the sparse encoding into the sparse encoding;

storing the sparse encoding to a memory within the at least one processing resource;

loading elements of the second matrix from the cache memory;

storing the selected elements to the memory within the at least one processing resource, wherein the selected elements of the second matrix correspond with non-zero value elements of the first matrix stored within the sparse encoding and are selected based on the metadata to indicate the locations for the non-zero elements;

performing the matrix multiply operation on the non-zero value elements from the sparse encoding and the selected elements of the second matrix; and

writing output of the matrix multiply operation to the memory within the at least one processing resource.

14. The method as in claim 13, wherein the cache memory is a level two cache memory and the compressor is to compress data stored to the level two cache memory.

15. The method as in claim 14, wherein the memory within the processing resource includes a level one cache memory.

16. The method as in claim 13, wherein the memory within the processing resource includes a memory within the matrix accelerator.

17. A data processing system comprising:

a memory device; and

a general purpose graphics processor comprising

a compute cluster including multiple processing resources coupled with a cache memory, at least one processing resource including a matrix accelerator, the matrix accelerator configured to perform a matrix multiply operation on multiple elements of a first matrix and a second matrix, wherein the first matrix is a sparse matrix, wherein elements of the first matrix are encoded into a sparse encoding, and wherein the sparse encoding includes a set of non-zero value elements and metadata to identify a location for the non-zero value elements;

wherein the sparse encoding is to be stored to the cache memory;

wherein the cache memory couples with a compressor to compress the sparse encoding into a compressed representation of the sparse encoding in association with a write the cache memory, wherein the compressed representation of the sparse encoding includes a compressed version of the set of non-zero value elements; and

wherein the at least one processing resource is configured to:

load the compressed representation of the sparse encoding from the cache memory, decompress the compressed representation of the sparse encoding into the sparse encoding, and store the compressed representation to a memory within the at least one processing resource;

load selected elements of the second matrix from the cache memory and store the selected elements to the memory within the at least one processing resource,

wherein the selected elements of the second matrix correspond with non-zero value elements of the first matrix stored within the compressed representation and are selected based on the metadata to indicate the locations for the non-zero value elements;

perform the matrix multiply operation on elements from the compressed representation and the selected elements of the second matrix; and

write output of the matrix multiply operation to the memory within the at least one processing resource.

18. The data processing system as in claim 17, wherein the cache memory is a level two cache memory and the compressor is to compress data stored to the level two cache memory.

19. The data processing system as in claim 17, wherein the memory within the processing resource includes a level one cache memory.

20. The data processing system as in claim 17, wherein the memory within the processing resource includes a register file.

21. The data processing system as in claim 17, wherein the memory within the processing resource includes a memory within the matrix accelerator.

22. The data processing system as in claim 17, wherein the first matrix includes weight data associated with a neural network and the second matrix includes input activation data associated with the neural network.

23. The data processing system as in claim 22, wherein output of the matrix multiply operation includes output activation data associated with the neural network.

24. The data processing system as in claim 23, wherein the output of the matrix multiply operation is a dense matrix.

25. The data processing system as in claim 17, wherein the matrix accelerator includes a systolic array of processing elements.