

WHAT IS CLAIMED IS:

1. A method of operating a graphic processing unit (GPU) comprising:
receiving, by the GPU from a shader compiler, a shader program comprising a preamble code block and a main shader code block, the preamble code block being executable to produce one or more results, the one or more results being a same one or more results for each of a plurality of groups of threads executing the shader program;
executing, by the GPU on a first group of threads of the plurality of groups of threads, the preamble code block to produce the one or more results;
storing, by the GPU, the one or more results of the preamble code block, the one or more results of the preamble code block accessible by each of the plurality of groups of threads; and
upon a determination that storing all of the one or more results of the preamble code block has completed, executing, by the GPU, the main shader code block for each thread of a group of threads of the plurality of groups of threads using the one or more results produced by executing the preamble code block.

2. The method of claim 1, further comprising:
identifying, by the shader compiler running on a processor, a code block of instructions being executable to produce the one or more results being the same one or more results for each of a plurality of groups of threads executing the shader program;
and
grouping, by the shader compiler running on the processor, the code block of instructions into the preamble code block of the shader program.

3. The method of claim 2, wherein the identification of the code block of instructions comprises identification of instructions that evaluate to a constant.

4. The method of claim 1, further comprising:
identifying the preamble code block based on identifying a shader preamble start instruction.

5. The method of claim 1, further comprising:

tracking whether or not that the preamble code block has been executed by the first group of threads allowing the GPU to execute a subsequent group of threads of the plurality of groups of threads.

6. The method of claim 1, further comprising:

tracking whether or not any group of threads of the plurality of groups of threads has executed prior to the first group of threads to determine, at least in part, whether or not the preamble code block has been previously executed.

7. The method of claim 1, further comprising:

tracking whether or not the preamble code block has been executed to determine, at least in part, whether to execute the preamble code block.

8. The method of claim 1, wherein the execution of the preamble code

block is executed by a scalar processor on the GPU.

9. The method of claim 1, wherein the storage of the one or more results is

on a writeable buffer managed cache.

10. The method of claim 1, wherein the storage of the one or more results is

accessible by all processing elements of a shader processor.

11. The method of claim 1, wherein the preamble code block evaluates to

one or more constant results.

12. The method of claim 1, further comprising:

loading, by the GPU, the one or more results of the preamble code block from random access memory (RAM) into a constant memory.

13. An apparatus for processing data, the apparatus comprising:

a graphics processing unit (GPU), the GPU comprising a memory and a shader core, the shader core comprising a control unit, a plurality of processing elements, and a constant memory, wherein the control unit is configured to:

receive, from a shader compiler, a shader program comprising a preamble code block and a main shader code block, the preamble code block being executable to produce one or more results, the one or more results being a same one or more results for each of a plurality of groups of threads executing the shader program;

direct at least one of the plurality of processing elements to execute, on a first group of threads of the plurality of groups of threads, the preamble code block to produce the one or more results;

store the one or more results of the preamble code block in the constant memory, the one or more results of the preamble code block accessible by each of the plurality of groups of threads; and

upon a determination that storing all of the one or more results of the preamble code block has completed, direct at least one of the plurality of processing elements to execute the main shader code block for each thread of a group of threads of the plurality of groups of threads using the one or more results produced by executing the preamble code block.

14. The apparatus of claim 13, further comprising a processor, wherein the processor is configured to:

identify a code block of instructions being executable to produce the one or more results being the same one or more results for each of a plurality of groups of threads executing the shader program; and

group the code block of instructions into the preamble code block of the shader program.

15. The apparatus of claim 14, wherein the identification of the code block of instructions comprises identification of instructions that evaluate to a constant.

16. The apparatus of claim 13, wherein the control unit is further configured to:

identify the preamble code block based on identifying a shader preamble start instruction.

17. The apparatus of claim 13, wherein the control unit is further configured to:

track whether or not that the preamble code block has been executed by the first group of threads allowing the GPU to execute a subsequent group of threads of the plurality of groups of threads.

18. The apparatus of claim 13, wherein the control unit is further configured to:

track whether or not any group of threads of the plurality of groups of threads has executed prior to the first group of threads to determine, at least in part, whether or not the preamble code block has been previously executed.

19. The apparatus of claim 13, wherein the control unit is further configured to:

track whether or not the preamble code block has been executed to determine, at least in part, whether to execute the preamble code block.

20. The apparatus of claim 13, wherein the execution of the preamble code block is executed by a scalar processor of the plurality of processing elements.

21. The apparatus of claim 13, wherein the stored one or more results is accessible by all processing elements of the shader core.

22. An apparatus configured to operate a graphic processing unit (GPU), the apparatus comprising:

means for receiving, from a shader compiler, a shader program comprising a preamble code block and a main shader code block, the preamble code block being executable to produce one or more results, the one or more results being a same one or more results for each of a plurality of groups of threads executing the shader program;

means for executing, on a first group of threads of the plurality of groups of threads, the preamble code block to produce the one or more results;

means for storing the one or more results of the preamble code block, the one or more results of the preamble code block accessible by each of the plurality of groups of threads; and

means for executing upon a determination that storing all of the one or more results of the preamble code block has completed, the main shader code block for each thread of a group of threads of the plurality of groups of threads using the one or more results produced by executing the preamble code block.

23. The apparatus of claim 22, further comprising:

means for identifying a code block of instructions being executable to produce the one or more results being the same one or more results for each of a plurality of groups of threads executing the shader program; and

means for grouping the code block of instructions into the preamble code block of the shader program.

24. The apparatus of claim 22, further comprising:

means for identifying the preamble code block based on identifying a shader preamble start instruction.

25. The apparatus of claim 22, further comprising:

means for tracking whether or not that the preamble code block has been executed by the first group of threads allowing the GPU to execute a subsequent group of threads of the plurality of groups of threads.

26. The apparatus of claim 22, further comprising:

means for tracking whether or not any group of threads of the plurality of groups of threads has executed prior to the first group of threads to determine, at least in part, whether or not the preamble code block has been previously executed.

27. The apparatus of claim 22, further comprising:

means for tracking whether or not the preamble code block has been executed to determine, at least in part, whether to execute the preamble code block.

28. The apparatus of claim 22, wherein the preamble code block evaluates to one or more constant results.

29. The apparatus of claim 22, further comprising:

means for loading the one or more results of the preamble code block from random access memory (RAM) into a constant memory.

30. A non-transitory computer-readable storage medium including instructions stored thereon that, when executed, cause at least one processor of to:

- receive, from a shader compiler, a shader program comprising a preamble code block and a main shader code block, the preamble code block being executable to produce one or more results, the one or more results being a same one or more results for each of a plurality of groups of threads executing the shader program;
- execute, on a first group of threads of the plurality of groups of threads, the preamble code block to produce the one or more results;
- store the one or more results of the preamble code block in a constant memory, the one or more results of the preamble code block accessible by each of the plurality of groups of threads; and
- upon a determination that storing all of the one or more results of the preamble code block has completed, execute the main shader code block for each thread of a group of threads of the plurality of groups of threads using the one or more results produced by executing the preamble code block.