CLAIMS:

1. A method comprising:

identifying a first plurality of instructions which require permutations depending on a non-orthogonal state (NOS) from a second plurality of instructions which are not dependent on the NOS;

marking each of the first plurality of instructions;

detecting a current NOS; and

dynamically patching the marked first plurality of instructions according to the current NOS, wherein the identifying and the marking are performed during compile time and the detecting and the patching are performed during shader execution time.

2. The method as in claim 1 wherein the first plurality of instructions comprise a plurality of shader instructions for implementing a graphics shader.

3. The method as in claim 1 or 2 wherein marking each of the first plurality of instructions comprises setting one or more bits within each of the first plurality of instructions to indicate NOSs on which the first plurality of instructions depend.

4. The method as in claim 2 further comprising:

compiling shader program code to a patchable kernel including the first plurality of instructions and the second plurality of instructions.

5. The method as in claim 4 wherein the patchable kernel is updated during shader execution time by performing dynamically patching the marked first plurality of instructions for the current NOS.

6. The method as in claim 2 wherein the graphics shader is selected from a group consisting of a pixel shader, a vertex shader, a geometry shader, a hull shader, and a domain shader.

7. The method as in any one of claims 1 to 6 further comprising:

executing the patched first plurality of instructions to implement a graphics shader adapted for the current NOS.

8. An apparatus comprising:

a shader compiler with instruction marking logic to identify a first plurality of instructions which require permutations depending on a non-orthogonal state (NOS) from a second plurality of instructions which are not dependent on the NOS and to mark each of the first plurality of instructions; and

instruction patch logic to detect a current NOS during shader execution time and dynamically patch the marked first plurality of instructions according to the current NOS.

9. The apparatus as in claim 8 wherein the first plurality of instructions comprise a plurality of shader instructions for implementing a graphics shader.

10. The apparatus as in claim 9 wherein detecting the current NOS and dynamically patching the marked first plurality of instructions for the current NOS are performed at a time at or after a shader execution function is called by an application.

11. The apparatus as in any one of claims 8 to 10 wherein marking each of the first plurality of instructions comprises setting one or more bits within each of the first plurality of instructions to indicate NOSs on which the first plurality of instructions depend.

12. The apparatus as in claim 9 wherein the shader compiler compiles shader program code to a patchable kernel including the first plurality of instructions and the second plurality of instructions.

13. The apparatus as in claim 12 wherein the patchable kernel is updated during shader execution time by performing dynamically patching the marked first plurality of instructions for the current NOS.

14. The apparatus as in claim 9 wherein the graphics shader is selected from a group consisting of a pixel shader, a vertex shader, a geometry shader, a hull shader, and a domain shader.

15. The apparatus as in any one of claims 8 to 14 further comprising:

a plurality of execution units (EUs) to execute the patched first plurality of instructions to implement a graphics shader adapted for the current NOS.

16. A computer program, causing a computer to perform procedures to to :

identify a first plurality of instructions which require permutations depending on a non-orthogonal state (NOS) from a second plurality of instructions which are not dependent on the NOS;

mark each of the first plurality of instructions;

detecting a current NOS; and

dynamically patch the marked first plurality of instructions according to the current NOS, wherein the identifying and the marking are performed during compile time and the detecting and the patching are performed at a time at or after a shader execution function is called by an application.

17. The computer program as in claim 16 wherein the first plurality of instructions comprise a plurality of shader instructions for implementing a graphics shader.

18. The computer program as in claim 16 or 17 wherein the procedure to mark each of the first plurality of instructions comprises a procedure to set one or more bits within each of the first plurality of instructions to indicate NOSs on which the first plurality of instructions depend.

19. The computer program as in claim 17 comprising an additional procedure to:

compile shader program code to a patchable kernel including the first plurality of instructions and the second plurality of instructions.

20. The computer program as in claim 19 wherein the patchable kernel is updated during shader execution time by performing the procedure to dynamically patch the marked first plurality of instructions for the current NOS.

21. The computer program as in claim 17 wherein the graphics shader is selected from a group consisting of a pixel shader, a vertex shader, a geometry shader, a hull shader, and a domain shader.

22. The computer program as in any one of claims 16 to 21 comprising an additional procedure to:

execute the patched first plurality of instructions to implement a graphics shader adapted for the current NOS.

23. A computer readable storage medium storing the computer program as in any one of claims 16 to 22.