CLAIMS:

1. An apparatus comprising:

logic, the logic at least partially comprising hardware logic, to cause one or more threads executing on a processor to stop at an instruction level granularity in response to a request to pre-empt the one or more threads,

wherein context data of the one or more threads are to be copied to memory in response to completion of any outstanding pending cycles of the one or more threads at the instruction level granularity, wherein no new threads are to be spawned in response to the request to pre-empt the one or more threads, wherein the one or more threads are to be stopped after completion of the outstanding pending cycles of the one or more threads at the instruction level granularity, wherein context data of the one or more threads are to be copied to the memory after the one or more threads have been stopped, wherein the context data is to originate from one of three sources comprising: thread spawner logic, an execution unit complex, or a Shared Local Memory (SLM).

2. The apparatus of claim 1, wherein the SLM is to store the context data prior to copying the context data to the memory.

3. The apparatus of claim 2, comprising logic to copy the context data from the SLM to the memory.

4. The apparatus of claim 3, wherein the logic to copy the context data is to comprise a Direct Memory Access (DMA) logic.

5. The apparatus of claim 1, wherein the context data of the one or more threads are to be copied to the memory in response to one or more software instructions.

6. The apparatus of claim 1, wherein the logic is to cause all threads executing on the processor to stop at the instruction level granularity in response to the request.

7. The apparatus of claim 1, wherein a Graphics Processing Unit (GPU) is to comprise the processor, wherein the one or more threads or a thread corresponding to the pre-emption request are to be executed as General-Purpose Computation on Graphics Processing Unit (GPGPU) on the GPU.

8. The apparatus of claim 1, comprising thread spawner logic to spawn the one or more threads.

9. The apparatus of claim 1, wherein the processor is to comprise the logic.

10. The apparatus of claim 1, wherein one or more of the processor having one or more processor cores, the logic, or the memory are on a single integrated circuit die.

11. A non-transitory computer-readable medium comprising one or more instructions that when executed on a processor configure the processor to perform one or more operations to:

cause one or more threads executing on the processor to stop at an instruction level granularity in response to a request to pre-empt the one or more threads,

wherein context data of the one or more threads are to be copied to memory in response to completion of any outstanding pending cycles of the one or more threads at the instruction level granularity, wherein no new threads are to be spawned in response to the request to pre-empt the one or more threads, wherein the one or more threads are to be stopped after completion of the outstanding pending cycles of the one or more threads at the instruction level granularity, wherein context data of the one or more threads are to be copied to the memory after the one or more threads have been stopped, wherein the context data is to originate from one of three sources comprising: thread spawner logic, an execution unit complex, or a Shared Local Memory (SLM).

12. The non-transitory computer-readable medium of claim 11, further comprising one or more instructions that when executed on the processor configure the processor to perform one or more operations to cause the context data of the one or more threads to be copied to the memory in response to one or more software instructions.

13. The non-transitory computer-readable medium of claim 11, further comprising one or more instructions that when executed on the processor configure the processor to perform one or more operations to cause all threads executing on the processor to stop at the instruction level granularity in response to the request.

14. The non-transitory computer-readable medium of claim 11, wherein a Graphics Processing Unit (GPU) is to comprise the processor, wherein the one or more threads or a thread corresponding to the pre-emption request are to be executed as General-Purpose Computation on Graphics Processing Unit (GPGPU) on the GPU.

15. The non-transitory computer-readable medium of claim 11, further comprising one or more instructions that when executed on the processor configure the processor to perform one or more operations to cause spawning of the one or more threads.

16. A method comprising:

causing one or more threads executing on a processor to stop at an instruction level granularity in response to a request to pre-empt the one or more threads,

wherein context data of the one or more threads are copied to memory in response to completion of any outstanding pending cycles of the one or more threads at the instruction level granularity, wherein no new threads are spawned in response to the request to pre-empt the one or more threads, wherein the one or more threads are stopped after completion of the outstanding pending cycles of the one or more threads at the instruction level granularity, wherein context data of the one or more threads are copied to the memory after the one or more threads have been stopped, wherein the context data originates from one of three sources comprising: thread spawner logic, an execution unit complex, or a Shared Local Memory (SLM).

17. The method of claim 16, further comprising causing the context data of the one or more threads to be copied to the memory in response to one or more software instructions.

18. The method of claim 16, further comprising causing all threads executing on the processor to stop at the instruction level granularity in response to the request.

19. The method of claim 16, wherein a Graphics Processing Unit (GPU) is to comprise the processor, the method further comprising executing the one or more threads or a thread corresponding to the pre-emption request as General-Purpose Computation on Graphics Processing Unit (GPGPU) on the GPU.

20. The method of claim 16, further comprising causing spawning of the one or more threads.

21. A system comprising:

a processor having one or more processor cores;

memory to store data, corresponding to at least one frame of a scene, to be accessed by at least one of the one or more processor cores;

a display device to present the at least one frame of the scene;

logic, the logic at least partially comprising hardware logic, to cause one or more threads executing on a processor to stop at an instruction level granularity in response to a request to pre-empt the one or more threads,

wherein context data of the one or more threads are to be copied to the memory in response to of completion any outstanding pending cycles of the one or more threads at the instruction level

granularity, wherein no new threads are to be spawned in response to the request to pre-empt the one or more threads, wherein the one or more threads are to be stopped after completion of the outstanding pending cycles of the one or more threads at the instruction level granularity, wherein context data of the one or more threads are to be copied to the memory after the one or more threads have been stopped, wherein the context data is to originate from one of three sources comprising: thread spawner logic, an execution unit complex, or a Shared Local Memory (SLM).

22. The system of claim 21, wherein the SLM is to store the context data prior to copying the context data to the memory.

23. The system of claim 21, wherein the context data of the one or more threads are to be copied to the memory in response to one or more software instructions.

24. The system of claim 21, wherein the logic is to cause all threads executing on the processor to stop at the instruction level granularity in response to the request.

25. The system of claim 21, wherein a Graphics Processing Unit (GPU) is to comprise the processor, wherein the one or more threads or a thread corresponding to the pre-emption request are to be executed as General-Purpose Computation on Graphics Processing Unit (GPGPU) on the GPU.