CLAIMS

What is claimed is:

1. A method comprising:

receiving an instruction for execution by a microprocessor;

dispatching the instruction to an issue queue without assigning a register tag to the instruction;

determining that the instruction is ready to issue;

assigning, in response to determining that the instruction is ready to issue, an available register tag to the instruction;

issuing the instruction;

determining that a register tag is not available for a next to complete (NTC) instruction;

performing, in response to determining that a flush criteria has been met, a partial flush of younger instructions from the microprocessor pipeline to reclaim register tags, wherein the younger instructions were received by the microprocessor after the NTC instruction, wherein the partial flush comprises flushing two or more younger instructions from the microprocessor pipeline without flushing one or more other younger instructions from the microprocessor pipeline;

assigning one of the reclaimed register tags to the NTC instruction; and

issuing the NTC instruction.

2. The method of claim 1, wherein the instruction is dependent on a producer instruction, and wherein determining that the instruction is ready to issue comprises:

determining that a ready bit for the instruction is set.

3. The method of claim 2, wherein a source register tag for the instruction identifies an entry in a register file that contains a register location that the instruction needs to execute.

4. The method of claim 2, wherein the ready bit identifies whether the producer instruction has already executed.

5. The method of claim 1, the method further comprising:

comparing, after issuing the instruction, an ITAG for the instruction to source ITAGs of other instructions in the issue queue;

identifying, based on the comparing, a second instruction that is dependent on the instruction; and

writing the register tag assigned to the instruction to a source register tag field for an entry for the second instruction in the issue queue, to an entry for the second instruction in a mapper, and to an entry for the second instruction in a save and restore buffer.

6. A system comprising a microprocessor configured to perform a method comprising:

receiving, in a microprocessor pipeline, an instruction for execution;

dispatching the instruction to an issue queue without assigning a register tag to the instruction;

determining that the instruction is ready to issue;

determining that a register tag is not available for the instruction;

determining whether a flush criteria having a plurality of flush criterions has been satisfied;

performing, in response to determining that the flush criteria has been satisfied, a flush of one or more younger instructions from the microprocessor pipeline to reclaim one or more register tags, wherein the one or more younger instructions were received by the microprocessor after the instruction, wherein a type of flush to be performed is dependent upon which flush criterion are satisfied, wherein the types of flush include a full flush of younger instructions in the microprocessor pipeline and a partial flush of younger instructions in the microprocessor pipeline;

assigning a reclaimed register tag to the instruction; and

issuing the instruction.

7. The system of claim 6, wherein the instruction is dependent on a producer instruction, and wherein determining that the instruction is ready to issue comprises:

determining that a ready bit for the instruction is set.

8. The system of claim 7, wherein a source register tag for the instruction identifies an entry in a register file that contains a register location that the instruction needs to execute.

9. The system of claim 7, wherein the ready bit identifies whether the producer instruction has already executed.

10. The system of claim 6, wherein the method further comprises:

comparing, after issuing the instruction, an ITAG for the instruction to source ITAGs of other instructions in the issue queue;

identifying, based on the comparing, a second instruction that is dependent on the instruction; and

writing the register tag assigned to the instruction to a source register tag field for an entry for the second instruction in the issue queue, to an entry for the second instruction in a mapper, and to an entry for the second instruction in a save and restore buffer.

11. The system of claim 6, wherein determining that the instruction is ready to issue comprises:

determining that the instruction is a next to complete instruction; and

determining that all instructions that the instruction is dependent on have issued.

12. The method of claim 1, wherein determining that the flush criteria has been met comprises:

determining that the NTC instruction has not received a register tag for a number of processor cycles; and

determining that the number of processor cycles exceeds a threshold.

13. The method of claim 1, wherein determining that the flush criteria has been met comprises:

determining a rate at which instructions are being issued; and

determining that the rate is below a threshold.

14. The method of claim 1, wherein determining that the flush criteria has been met comprises:

determining a number of register tags that are in a free list; and

determining that the number of register tags that are in the free list is below a threshold.

15. The method of claim 1, wherein performing the partial flush comprises:

flushing out N younger instructions starting from a head pointer—N.

16. The system of claim 6, wherein the method further comprises:

maintaining, by the microprocessor, a free list of register tags that are available to be assigned to instructions and a reserve register tag for assigning in response to detecting that there are no available tags in the free list for a next to complete (NTC) instruction;

receiving a second instruction, wherein the second instruction is the NTC instruction;

determining that there is no available register tag in the free list of register tags for the second instruction;

assigning the reserve register tag to the second instruction; and

issuing the second instruction.

17. A method comprising:

receiving, in a microprocessor pipeline, an instruction for execution by a microprocessor;

dispatching the instruction to an issue queue without assigning a register tag to the instruction;

maintaining, by the microprocessor, a free list of register tags that are available to be assigned to instructions and a reserve register tag for assigning in response to detecting that there are no available tags in the free list for a next to complete (NTC) instruction;

determining that the instruction is the NTC instruction;

determining that a register tag is not available in the free list for the instruction;

assigning the instruction to a reserve entry in a register file; and

issuing the instruction.

18. The method of claim 17, the method further comprising:

receiving a second instruction for execution by the microprocessor;

determining that the second instruction is the NTC instruction;

determining that a register tag is not available in the free list for the instruction and that the reserve register tag is not available;

performing a flush of younger instructions from the microprocessor pipeline to reclaim one or more register tags;

assigning a reclaimed register tag to the second instruction; and

issuing the second instruction.

19. The method of claim 18, wherein performing the flush of the younger instructions from the microprocessor pipeline comprises:

determining that a flush criteria having a plurality of flush criterions has been satisfied, wherein a type of flush to be performed is dependent upon which flush criterion are satisfied.

20. The method of claim 18, wherein performing the flush of the younger instructions from the microprocessor pipeline comprises:

performing a partial flush of younger instructions from the microprocessor pipeline, wherein the younger instructions were received by the microprocessor after the second instruction, wherein the partial flush comprises flushing one or more younger instructions from the microprocessor pipeline without flushing one or more other younger instructions from the microprocessor pipeline.