CLAIMS

1. A method of processing in a data processing system including a system memory, the method comprising:

a processor core processing a translation load instruction, wherein the translation load instruction includes a protection field specifying a desired access protection to be specified in a translation entry for a memory page, and wherein processing the translation load instruction includes:

calculating an effective address within the memory page;

determining that the translation entry containing the desired access protection is stored within at least one translation lookaside buffer of the data processing system;

and based on determining that the translation entry containing the desired access protection is stored within the at least one translation lookaside buffer, ending processing of the translation load instruction without further processing of the translation load instruction;

wherein the translation load instruction is unrelated to a data load, data store, or instruction fetch access.

2. The method of claim 1, wherein:

the at least one translation lookaside buffer includes a first translation lookaside buffer in the processor core of a processing unit;

the translation load instruction includes a load field;

the method further comprises the processor core, based on the load field being set, installing the translation entry for the effective address in the first translation lookaside buffer in the processor core.

3. The method of claim 2, wherein:

the at least one translation lookaside buffer includes a plurality of different translation lookaside buffers in the processing unit, said plurality of different translation lookaside buffers including the first translation lookaside buffer;

and the load field comprises one or more bits, each of the one or more bits individually specifies into which of the plurality of different translation lookaside buffers the translation entry is to be installed.

4. The method of claim 1, wherein the system memory stores a page frame table containing a plurality of page table entries, and wherein page table entries accessed from the page frame table by a memory management unit are cacheable in the at least one translation lookaside buffer.

5. The method of claim 1, wherein:

the protection field in the translation load instruction is a first protection field;

the translation entry includes a second protection field;

and the method further comprises the processor core invoking a protection fault handler based on a mismatch of access protections specified in the first and second protection fields.

6. The method of claim 5, and further comprising:

determining whether a modification of access protections requested by the translation load instruction is legal; and

responsive to determining the modification of access protections requested by the translation load instruction is legal, updating the second protection field in the translation entry.

7. The method of claim 1, wherein:

the data processing system includes a nest memory management unit coupled to a system fabric of the data processing system;

and the processing comprises the processor core processing the translation load instruction based on receiving an interrupt initiated by the nest memory management unit.

8. A processing unit for a data processing system, the processing unit comprising:

a processor core including:

a memory management unit that translates effective addresses to real addresses;

and an execution unit that executes instructions;

wherein the processor core is configured to perform:

processing a translation load instruction, wherein the translation load instruction includes a protection field specifying a desired access protection to be specified in a translation entry for a memory page, and wherein processing the translation load instruction includes:

calculating an effective address within the memory page;

determining whether the translation entry containing the desired access protection is stored within at least one translation lookaside buffer of the data processing system;

based on determining the translation entry containing the desired access protection is not stored within the at least one translation lookaside buffer, loading the translation entry with the desired access protection in the at least one translation lookaside buffer;

and based on determining the translation entry containing the desired access protection is stored within the at least one translation lookaside buffer, ending processing of the translation load instruction without further processing of the translation load instruction;

wherein the translation load instruction is unrelated to a data load, data store, or instruction fetch access.

9. The processing unit of claim 8, wherein:

the at least one translation lookaside buffer includes a first translation lookaside buffer in the processor core;

the translation load instruction includes a load field;

the processor core is configured, based on the load field being set, to install the translation entry for the effective address in the first translation lookaside buffer in the processor core.

10. The processing unit of claim 9, wherein:

the at least one translation lookaside buffer includes a plurality of different translation lookaside buffers in the processing unit, said plurality of different translation lookaside buffers including the first translation lookaside buffer;

and the load field comprises one or more bits, each of the one or more bits individually specifies into which of the plurality of different translation lookaside buffers the translation entry is to be installed.

11. The processing unit of claim 8, wherein a system memory of the data processing system stores a page frame table containing a plurality of page table entries, and wherein page table entries accessed from the page frame table by the memory management unit are cacheable in the at least one translation lookaside buffer.

12. The processing unit of claim 8, wherein:

the protection field in the translation load instruction is a first protection field;

the translation entry includes a second protection field; and

the processor core invokes a protection fault handler based on a mismatch of access protections specified in the first and second protection fields.

13. The processing unit of claim 12, wherein the processor core is further configured to perform:

determining whether a modification of access protections requested by the translation load instruction is legal; and

responsive to determining the modification of access protections requested by the translation load instruction is legal, updating the second protection field in the translation entry.

14. The processing unit of claim 8, wherein:

the data processing system includes a nest memory management unit coupled to a system fabric of the data processing system; and

the processor core processes the translation load instruction based on receiving, via the system fabric, an interrupt initiated by the nest memory management unit.

15. A data processing system including a plurality of processing units according to claim 8 and a system fabric coupling the plurality of processing units.

16. A program product, comprising:

a computer-readable storage device;

and a design structure stored in the computer-readable storage device for designing, manufacturing, or testing an integrated circuit, the design structure comprising:

a processing unit for a data processing system, the processing unit including:

a processor core including:

a memory management unit that translates effective addresses to real addresses; and

an execution unit that executes instructions;

wherein the processing unit is configured to perform:

processing a translation load instruction, wherein the translation load instruction includes a protection field specifying a desired access protection to be specified in a translation entry for a memory page, and wherein processing the translation load instruction includes:

calculating an effective address within the memory page;

and determining whether the translation entry containing the desired access protection is stored within at least one translation lookaside buffer of the data processing system;

based on determining the translation entry containing the desired access protection is not stored within the at least one translation lookaside buffer, loading the translation entry with the desired access protection in the at least one translation lookaside buffer;

and based on determining the translation entry containing the desired access protection is stored within the at least one translation lookaside buffer, ending processing of the translation load instruction without further processing of the translation load instruction;

wherein the translation load instruction is unrelated to a data load, data store, or instruction fetch access.

17. The program product of claim 16, wherein:

the at least one translation lookaside buffer includes a first translation lookaside buffer in the processor core;

the translation load instruction includes a load field;

the processor core is configured, based on the load field being set, to install the translation entry for the effective address in the first translation lookaside buffer in the processor core.

18. The program product of claim 16, wherein a system memory of the data processing system stores a page frame table containing a plurality of page table entries, and wherein page table entries accessed from the page frame table by the memory management unit are cacheable in the at least one translation lookaside buffer.

19. The program product of claim 16, wherein:

the protection field in the translation load instruction is a first protection field;

the translation entry includes a second protection field; and

the processor core invokes a protection fault handler based on a mismatch of access protections specified in the first and second protection fields.