Systems and Methods for Policy Execution Processing

RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 62/625,634, filed on February 2, 2018. The entire teachings of the above application are incorporated herein by reference.

BACKGROUND

[0002] The present disclosure relates generally to systems and methods for policy execution processing, for example, to enforce security policies. Computer processors, which include electronic circuits configured to execute one or more computer program instructions, are known. Furthermore, it is known to implement security policies to prevent a computer processor from compromising sensitive information.

SUMMARY

[0003] In one aspect, the disclosure is directed to a processing system comprising a host processing domain comprising a host processor. The host processor is configured to receive at least one instruction comprising (i) operand information relating to one or more operands, and (ii) operation information indicative of an operation to be performed on the one or more operands. The host processor is further configured to provide, to a metadata processing domain, (i) the at least one instruction undecoded in accordance with an instruction set architecture implemented by the host processor, and (ii) instruction information including operation output information generated by the host processor executing the operation indicated in the operation information on the one or more operands. The metadata processing domain comprises a write interlock, which is configured to, in response to the host processing domain, obtain operation output information, and place the operation output information into a queue. The metadata processing domain further comprises a tag processing unit configured to receive, from the host processing domain, the at least one instruction or the instruction information, decode the at least one instruction to generate at least one decoded instruction, and use the at least one decoded instruction or the instruction information to obtain one or more input metadata tags associated with the at least one instruction. The tag processing unit is further configured to determine, in accordance with one or more policies being enforced and in accordance with the one or more input metadata tags associated with the at least one instruction, whether the at least one instruction is allowed. The tag processing unit is further configured to, responsive to a determination that the instruction is allowed,

constant value that indicates a write transaction is underway, (ii) perform a second write to write new data, and (iii) perform a third write to write a new tag associated with the new data.

[0019] In another aspect, the disclosure is directed to a computer-based method of processing program instructions, comprising, in a host processing domain, by an application processor, receiving each of the instructions comprising (i) operand information relating to one or more operands, and (ii) operation information indicative of an operation to be performed on the one or more operands. The method further comprises providing, to a metadata processing domain, (i) received instruction undecoded, and (ii) instruction information including operation output information generated by the application processor executing the operation indicated in the operation information on the one or more operands. The method further comprises, in the metadata processing domain, by a write interlock, responsive to the host processing domain, obtaining the operation output information, and placing the operation output information in a queue. The method further comprises, by a tag processing unit, (i) receiving, from the host processing domain, the instruction undecoded and the instruction information, (ii) decoding the instruction to generate a decoded instruction, (iii) using the decoded instruction or the instruction information to obtain one or more input metadata tags associated with the at least one instruction, (iv) determining, in accordance with one or more policies being enforced and in accordance with the one or more input metadata tags associated with the at least one instruction whether the at least one instruction is allowed, and (v) responsive to determining that the instruction is allowed, causing the queue of the write interlock to write to memory the operation output information in a manner that associates the operation output information with at least one output metadata tag.

[0020] The tag processing unit may comprise a rule cache configured to store one or more rule entries of at least one policy of the one or more policies enforced by the metadata processing domain. Determining that the instructions are allowed may comprise determining that the rule cache stores a rule entry matching the one or more input metadata tags associated with the at least one instruction.

[0021] The method may further comprise using, by the tag processing unit, information stored in the rule entry to provide the at least one output metadata tag to be associated with the operation output information.

[0022] The metadata processing domain may comprise a policy execution processor. Determining that the instructions are allowed may comprise (i) determining that the rule cache does not store a rule entry matching the one or more input metadata tags associated with the at least one instruction, (ii) responsive to a determination that the rule cache does not store a rule entry matching the one or more input metadata tags associated with the at least

CLAIMS

What is claimed is:

1. A processing system comprising:

(a) a host processing domain comprising a host processor configured to:

receive at least one instruction comprising (i) operand information relating to one or more operands, and (ii) operation information indicative of an operation to be performed on the one or more operands; and

provide, to a metadata processing domain, (i) the at least one instruction undecoded in accordance with an instruction set architecture implemented by the host processor, and (ii) instruction information including operation output information generated by the host processor executing the operation indicated in the operation information on the one or more operands; and

(b) the metadata processing domain comprising:

(i) a write interlock configured to:

in response to the host processing domain, obtain operation output information; and

place the operation output information into a queue;

(ii) a tag processing unit configured to:

receive, from the host processing domain, the at least one instruction or the instruction information;

decode the at least one instruction to generate at least one decoded instruction;

use the at least one decoded instruction or the instruction information to obtain one or more input metadata tags associated with the at least one instruction;

determine, in accordance with one or more policies being enforced and in accordance with the one or more input metadata tags associated with the at least one instruction, whether the at least one instruction is allowed; and

responsive to a determination that the instruction is allowed, cause the queue of the write interlock to write to memory the operation output information in a manner that associates the operation output information with at least one output metadata tag.

2. The processing system of claim 1, wherein the tag processing unit comprises a rule cache configured to store one or more rule entries of at least one policy of the one or more policies enforced by the metadata processing domain.

3. The processing system of claim 2, wherein the determination that the instruction is allowed comprises:

determine that the rule cache stores a rule entry matching the one or more input metadata tags associated with the at least one instruction.

4. The processing system of claim 3, wherein the tag processing unit is configured to use information stored in the rule entry to provide the at least one output metadata tag to be associated with the operation output information.

5. The processing system of claim 2, wherein the metadata processing domain comprises a policy execution processor, and wherein the determination that the instruction is allowed comprises:

determine that the rule cache does not store a rule entry matching the one or more input metadata tags associated with the at least one instruction;

responsive to a determination that the rule cache does not store a rule entry matching the one or more input metadata tags associated with the at least one instruction, provide, to the policy execution processor, the one or more input metadata tags associated with the at least one instruction; and

receive, from the policy execution processor, the at least one output metadata tag to be associated with the operation output information; and

wherein the policy execution processor is configured to:

receive, from the tag processing unit, the one or more input metadata tags associated with the at least one instruction;

execute policy code against the one or more input metadata tags associated with the at least one instruction to determine whether the at least one instruction is allowed; and

responsive to a determination that the at least one instruction is allowed, install, into the rule cache, a rule entry based on the one or more input metadata tags associated with the at least one instruction and the at least one output metadata tag.

6. The processing system of claim 5, wherein the policy execution processor is configured to execute a secure boot operation, the policy execution processor comprising a boot Read Only Memory (ROM) that stores one or more public keys, and stores code that can (i) read an image from an external memory device, authenticate and decrypt the image using the one or more public keys, and enable the host processor to continue its boot process upon successful authentication and decryption.

7. The processing system of claim 6, wherein the secure boot operation comprises:

at reset, the host processor remains held in a reset state; and

the policy execution processor:

(i) starts execution at its reset vector;

(ii) boots policy software into its own memory space;

(iii) configures one or more memory fabric protection configuration registers to define memory regions that each initiator can access, to protect a region of memory to hold a policy data segment;

(iv) initializes the policy data segment;

(v) copies a boot-loader for the host processor from the external memory device into main memory; and

(vii) release the host processor from the reset state.

8. The processing system of claim 1, wherein:

the host processor is further configured to provide, to the metadata processing domain, update information indicative of one or more updates to the host processor's state as a result of executing the at least one instruction; and

the metadata processing domain is further configured to, responsive to a determination that the at least one instruction is allowed, use the update information to update a shadow register configured to store a shadow copy of the host processing domain as of a most-recently-allowed instruction; and

wherein the at least one instruction comprises a first instruction, the instruction information comprises first instruction information, and the one or more input metadata tags comprise one or more first input metadata tags, and wherein the tag processing unit is further configured to:

receive, from the host processing domain, second instruction information relating to a second instruction executed by the host processor;

use the second instruction information to obtain one or more second input metadata tags associated with the second instruction;

determine, in accordance with the one or more policies being enforced and in accordance with the one or more second metadata tags associated with the second instruction, whether the second instruction is allowed; and

responsive to a determination that the second instruction is not allowed, communicate one or more rollback signals to the host processing domain to restore a state of the host processing domain to the shadow copy of the host processing domain.

9. The processing system of claim 1, wherein the one or more input metadata tags are inaccessible to the host processor.

10. A processing system of claim 1 wherein the host processor is an application processor, and wherein

(a) in an application processing domain, the application processor is configured to:

implement a first instruction set architecture,

the received at least one instruction being formatted in accordance with the first instruction set architecture; and

provide, to the metadata processing domain, the received at least one instruction in an instruction stream; and

(b) in the metadata processing domain, the tag processing unit further comprising at least one decode table, the tag processing unit being configured to extract the operand information and the operation information from the received at least one instruction.

11. The processing system of claim 10, wherein the tag processing unit comprises at least one look-up table configured to determine, based on the received at least one instruction, instruction care bit information indicative of one or more care bits, and operation group information indicative of an operation group.

12. The processing system of claim 10, wherein the tag processing unit further comprises a buffered interface configured to store the instructions from the application processor when the tag processing unit is stalled, and to provide stored instructions for use by the tag processing unit when the instruction stream is stalled.

13. The processing system of claim 10, wherein the tag processing unit further comprises a tag map table (TMT) configured to convert one or both of a physical instruction

address and a data memory page address, into one or both of (i) a corresponding associated tag addresses and (ii) directly into a tag.

14. The processing system of claim 13, wherein the TMT is configured to accommodate an immediate value tag, specified for a particular size of a memory region, the particular size being selected from a range of memory region sizes.

15. The processing system of claim 10, wherein the metadata processing domain further comprises a Metadata Tag Interlock Queue configured to:

(i) perform a first write to change a tag address to a predetermined constant value that indicates a write transaction is underway;

(ii) perform a second write to write new data; and

(iii) perform a third write to write a new tag associated with the new data.

16. A computer-based method of processing program instructions, comprising:

(a) in a host processing domain, by an application processor:

receiving each of the instructions comprising (i) operand information relating to one or more operands, and (ii) operation information indicative of an operation to be performed on the one or more operands; and

providing, to a metadata processing domain, (i) received instruction undecoded, and (ii) instruction information including operation output information generated by the application processor executing the operation indicated in the operation information on the one or more operands; and

(b) in the metadata processing domain:

by a write interlock:

responsive to the host processing domain, obtaining the operation output information; and

placing the operation output information in a queue; and

by a tag processing unit:

receiving, from the host processing domain, the instruction undecoded and the instruction information;

decoding the instruction to generate a decoded instruction;

using the decoded instruction or the instruction information to obtain one or more input metadata tags associated with the at least one instruction;

determining, in accordance with one or more policies being enforced and in accordance with the one or more input metadata tags associated with the at least one instruction whether the at least one instruction is allowed; and

responsive to determining that the instruction is allowed, causing the queue of the write interlock to write to memory the operation output information in a manner that associates the operation output information with at least one output metadata tag.

17. The method of claim 16, wherein the tag processing unit comprises a rule cache configured to store one or more rule entries of at least one policy of the one or more policies enforced by the metadata processing domain, and determining that the instructions are allowed comprises:

determining that the rule cache stores a rule entry matching the one or more input metadata tags associated with the at least one instruction.

18. The method of claim 17, further comprising, by the tag processing unit, using information stored in the rule entry to provide the at least one output metadata tag to be associated with the operation output information.

19. The method of claim 16, wherein the metadata processing domain comprises a policy execution processor, and determining that the instructions are allowed comprises:

determining that the rule cache does not store a rule entry matching the one or more input metadata tags associated with the at least one instruction;

responsive to a determination that the rule cache does not store a rule entry matching the one or more input metadata tags associated with the at least one instruction, providing, to the policy execution processor, the one or more input metadata tags associated the at least one instruction; and;

receiving, from the policy execution processor, the at least one output metadata tag to be associated with the operation output information; and

further comprising, by the policy execution processor:

receiving, from the tag processing unit, the one or more input metadata tags associated with the at least one instruction;

executing policy code against the one or more input metadata tags associated with the at least one instruction to determine whether the at least one instruction is allowed; and

responsive to a determination that the at least one instruction is allowed, installing into the rule cache, a rule entry based on the one or more input metadata tags associated with the at least one instruction and the at least one output metadata tag.

20. The method of claim 16, further comprising:

by the host processor, providing to the metadata processing domain, update information indicative of one or more updates to the host processor's state as a result of executing the at least one instruction; and

by the metadata processing domain, responsive to a determination that the at least one instruction is allowed, using the update information to update a shadow register configured to store a shadow copy of the host processing domain as of a most-recently-allowed instruction; and

wherein the at least one instruction comprises a first instruction, the instruction information comprises first instruction information, and the one or more input metadata tags comprise one or more first input metadata tags, further comprising, by the tag processing unit:

receiving, from the host processing domain, second instruction information relating to a second instruction executed by the host processor;

using the second instruction information to obtain one or more second input metadata tags associated with the second instruction;

determining, in accordance with the one or more policies being enforced and in accordance with the one or more second metadata tags associated with the second instruction, whether the second instructions is allowed; and

responsive to a determination that the second instruction is not allowed, communicating one or more rollback signals to the host processing domain to restore a state of the host processing domain to the shadow copy of the host processing domain.