CLAIMS

1. A data processing system for handling an input/output store instruction, the data processing system comprising:

a data processing unit including a core and system firmware, the data processing unit configured to perform a method, the method comprising:

identifying an input/output function by an address specified using the input/output store instruction, the input/output store instruction specifying at least the input/output function with an offset through the address, at least one of data to be transferred and a pointer to data to be transferred, and a length of the data;

verifying whether access to the input/output function is allowed on an address space and on a guest instance level;

completing the input/output store instruction before an execution of the input/output store instruction in a system nest communicatively coupled to the data processing unit via an aggregation buffer and to at least one external device is completed, the system nest configured to asynchronously load from and store data to the at least one external device;

providing notification through an interrupt, based on detecting an error during an asynchronous execution of the input/output store instruction in the data processing unit, transmitting data of a failed asynchronous execution;

collecting a completion of the input/output store instruction and updating one or more input/output status buffers of multiple input/output status buffers based on received completion messages; and

delaying an execution of the input/output store instruction, until an input/output status buffer of the multiple input/output status buffers is available to store information about a completion of pending input/output store instructions;

wherein the data processing unit further comprises an asynchronous core-nest interface, the asynchronous core-nest interface comprising an input/output status array including the multiple input/output status buffers to collect returned states from the system nest, the aggregation buffer being further communicatively coupled to the asynchronous core-nest interface, and

wherein the aggregation buffer comprises an early completion logic to deliver a free for reuse message after sending a request.

2. The data processing system of claim 1, wherein the method further comprises delaying, by the data processing unit, an execution of at least one of an asynchronous input/output store instruction and a synchronous input/output store instruction, until an input/output status buffer of the multiple input/output status buffers is available to store information about a completion of one or more pending input/output store instructions.

3. The data processing system of claim 1, wherein the data is transferred by the input/output store instruction through an asynchronous transmit mechanism with an early completion message in multiple data packets to the aggregation buffer, based on the length of the data exceeding a defined size.

4. The data processing system of claim 1, wherein the method further comprises collecting message states from the system nest, the message states comprising completion status from the system nest.

5. The data processing system of claim 4, wherein the message states are numbered by an input/output status buffer index.

6. The data processing system of claim 4, wherein the completion status is numbered by an input/output status buffer index.

7. The data processing system of claim 1, wherein the method further comprises collecting message states from an input/output bus controller coupled to the system nest, the message states comprising completion status from the system nest.

8. The data processing system of claim 1, wherein the system firmware comprises an asynchronous input/output driver code to handle the input/output store instruction, and an array

management logic to allocate and deallocate input/output status buffers in the input/output status array and to initiate a start of a new input/output store instruction.

9. The data processing system of claim 8, wherein the core comprises an asynchronous setup code to handle memory requirements for status information of the asynchronous input/output driver code.

10. The data processing system of claim 1, wherein the asynchronous core-nest interface comprises an asynchronous core-nest interface forwarding component to forward data with local completion.

11. A computer program product for handling an input/output store instruction, the computer program product comprising:

at least one computer readable storage medium readable by at least one processing circuit and storing instructions for performing a method comprising:

identifying an input/output function by an address specified using the input/output store instruction, the input/output store instruction specifying at least the input/output function with an offset through the address, at least one of data to be transferred and a pointer to data to be transferred, and a length of the data;

verifying whether access to the input/output function is allowed on an address space and on a guest instance level;

completing, by a data processing unit of a data processing system, the input/output store instruction before an execution of the input/output store instruction in a system nest communicatively coupled to the data processing unit via an aggregation buffer and to at least one external device is completed, the system nest configured to asynchronously load from and store data to the at least one external device;

providing notification through an interrupt, based on detecting an error during an asynchronous execution of the input/output store instruction in the data processing unit, transmitting data of a failed asynchronous execution;

collecting a completion of the input/output store instruction and updating one or more input/output status buffers of multiple input/output status buffers based on received completion messages; and

delaying an execution of the input/output store instruction, until an input/output status buffer of the multiple input/output status buffers is available to store information about a completion of pending input/output store instructions;

wherein the data processing unit further comprises an asynchronous core-nest interface, the asynchronous core-nest interface comprising an input/output status array including the multiple input/output status buffers to collect returned states from the system nest, the aggregation buffer being further communicatively coupled to the asynchronous core-nest interface, and

wherein the aggregation buffer comprises an early completion logic to deliver a free for reuse message after sending a request.

12. The computer program product of claim 11, wherein the method further comprises delaying, by the data processing unit, an execution of at least one of an asynchronous input/output store instruction and a synchronous input/output store instruction, until an input/output status buffer of the multiple input/output status buffers is available to store information about a completion of one or more pending input/output store instructions.

13. A computer-implemented method of handling an input/output store instruction, the computer-implemented method comprising:

identifying an input/output function by an address specified using the input/output store instruction, the input/output store instruction specifying at least the input/output function with an offset through the address, at least one of data to be transferred and a pointer to data to be transferred, and a length of the data;

verifying whether access to the input/output function is allowed on an address space and on a guest instance level;

completing, by a data processing unit of a data processing system, the input/output store instruction before an execution of the input/output store instruction in a system nest communicatively coupled to the data processing unit via an aggregation buffer and to at least

one external device is completed, the system nest configured to asynchronously load from and store data to the at least one external device;

providing notification through an interrupt, based on detecting an error during an asynchronous execution of the input/output store instruction in the data processing unit, transmitting data of a failed asynchronous execution;

collecting a completion of the input/output store instruction and updating one or more input/output status buffers of multiple input/output status buffers based on received completion messages; and

delaying an execution of the input/output store instruction, until an input/output status buffer of the multiple input/output status buffers is available to store information about a completion of pending input/output store instructions;

wherein the data processing unit further comprises an asynchronous core-nest interface, the asynchronous core-nest interface comprising an input/output status array including the multiple input/output status buffers to collect returned states from the system nest, the aggregation buffer being further communicatively coupled to the asynchronous core-nest interface, and

wherein the aggregation buffer comprises an early completion logic to deliver a free for reuse message after sending a request.

14. The computer-implemented method of claim 13, further comprising delaying, by the data processing unit, an execution of at least one of an asynchronous input/output store instruction and a synchronous input/output store instruction, until an input/output status buffer of the multiple input/output status buffers is available to store information about a completion of one or more pending input/output store instructions.