Claims

1. An order book management device (10), implemented on an integrated circuit, for aggregating orders identified in input data streams into at least one order book, each order comprising a price and a quantity information, each order book being associated with a tradable object and comprising a sell side and a buy side, each order book side storing a list of limits, each limit being associated with a given price and a quantity, the quantity associated with a limit corresponding to the sum of the quantities of the received orders comprising said given price, wherein each side of the order book comprises :

- a first data structure for maintaining a selected number P of limits corresponding to the P limits having the highest prices among the N limits maintained in the buy side of the book, or the lowest prices among the N limits maintained in the sell side of the order book, the orders in said first data structure being sorted by price depending on the order book side, and
- a second data structure arranged in the form of *a* tree and comprising the N-P limits having the lowest prices among the N limits in the buy side of the book, or the highest prices among the N limits in the sell side of the book,

wherein said order book management device further comprises a first management core (2) for processing each received limit related command comprising a price information, updating said first data structure in response to the processing of said command and/or generating at least one update command to a second management core (3), said second management core (3) being configured to update the second data structure based on said update commands.

2. The order book management device of claim 1, wherein the tree used for said second data structure is a tree comprising a root node (53) and at least one leaf node (55).

3. The order book management device of claim 2, wherein the root node (53) comprises a set of entries, each entry being associated with a reference price and a pointer to a respective leaf node (55), the entries in the root node (53) being sorted by reference prices depending on the order book side, and wherein each leaf node comprises a non-sorted list of limits associated with respective prices, the prices in each leaf node (55) ranging between a first value and a second value, the first value being the price associated with the corresponding pointer (54) in the root node (53) and the second value being related to the price associated with the previous pointer (54) in the root node (53).

4. The order book management device (10) of claim 3, wherein the root node comprises a first entry storing the highest limit in the buy side or the lowest limit in the sell side of the second data structure of the book, the second value of each leaf node being the price associated with said highest or lowest limit in the first entry, if there is no previous pointer to a leaf node in the root node (53).

5. The order book management device of any preceding claim 3~~2~~ and 4~~3~~, wherein the first value is included in the leaf node (55) and the second value is excluded from the leaf node (55).

6. The order book management device of any preceding claims, wherein the first management core (2) comprises an internal cache (21) for storing ~~loading~~ the first data structure of each order book.

7. The order book management device of claim 6, wherein an entry is allocated in said cache for each order book using a "Least Recently Used" (LRU) scheme.

8. The order book management device of any preceding claim 2 to 6, wherein said tree is a B+ tree.

9. The order book management device of any preceding claim 2 to 6, wherein the processing in the first management core (2) is asynchronous with respect to the processing in the second management core (3).

10. The order book management device of any preceding claims, wherein the first management core (2) comprises a processing core (24) configured to process each input command (200) and the associated first data structure in three clock cycles, by processing the limits inside the first data structure in parallel and in a pipelined way.

11. The order book management device of claim 10, wherein the first management core (2) further comprises an array of comparators (23) for comparing the price associated with the order identified in said input data stream with the price in each entry of the first data structure, and updating the first data structure and/or the second data structure based on the results provided by the comparators.

12. The order book management device of any preceding claims 2 to 11, wherein the second management core (3) comprises at least one state machine for processing each leaf node and the root node of said second data structure.

13. The order book management device of claim 12, wherein the integrated circuit comprises at least one processing core, and wherein said state machines are implemented in parallel in said processing cores, a command dispatcher being provided for dispatching the commands generated by the first management core (2) to the processing cores.

14. The order book management device of any preceding claims 1 to 13, where the number of limits P in the first data structure is such that:

P*S = N*W

where N is an integer, S is the size of a limit in bits; and W is the memory word width in bits.

15. An order book management method for aggregating orders identified in input data streams into at least one order book, each order comprising a price and a quantity information, each order book being associated with a tradable object and comprising a sell side and a buy side, each order book side storing a list of limits, each limit being associated with a given price and a quantity, the quantity associated with a limit corresponding to the sum of the quantities of the received orders comprising said given price, wherein it comprises for each order book side :

- maintaining, in a first data structure, a selected number P of limits corresponding to the P limits having the highest prices among the N limits maintained in the buy side of the order book, or the lowest prices among the N limits maintained in the sell side of the order book, the limits in said first data structure being sorted by price depending on the order book side, and
- maintaining a second data structure arranged in the form of *a* tree and comprising the N-P limits having the lowest prices among the N limits of the buy side of the order book, or the highest prices among the N limits in the sell side of the order book,

wherein it further comprises processing each received limit related command comprising a price information, updating said first data structure in response to the processing of said command and/or generating at least one update command to update the second data structure based on said update commands.