

CLAIMS

1. A system configured for lock-free conveyance of variable value updates as a unitary set, the system comprising:
 - a memory containing a set of variables having respective non-updated values, the set including an update status indication variable (Var1) and at least one other variable;
 - a processor in operable communication with the memory, the processor configured to perform steps for conveying variable value updates as a unitary set without lock reliance, the steps including a values placing step and a values retrieving step;wherein the values placing step comprises performing the following actions in sequence: replacing a non-updated value in the update status indication variable (Var1) with a different value (Z) that indicates updating is underway, executing a first release barrier, replacing the non-updated value of each other variable with a respective corresponding update value, executing a second release barrier, and replacing the value (Z) in the update status indication variable (Var1) with a different value that is also an update value;
 - wherein the values retrieving step comprises performing the following actions in sequence: reading from the update status indication variable (Var1) into a first destination status variable (DestVar1A), executing a first acquire barrier, reading from other variables of the set into respective destination variables, executing a second acquire barrier, reading from the update status indication variable (Var1) into a second destination status variable (DestVar1B), and repeating the foregoing retrieving step actions if the first destination status variable (DestVar1A) has the value (Z) or if the first destination status variable (DestVar1A) and the second destination status variable (DestVar1B) contain different values from each other;
 - whereby the system conveys the update values through the variables to respective destination variables without lock reliance, and the system conveys the update values as a unitary set in that after the values placing step and the values retrieving step are completed the values in the destination variables will not include both a non-updated value and an update value.
2. The system of claim 1, wherein the value (Z) is different from all valid non-

updated values and is also different from all valid update values.

3. The system of claim 1, wherein one of the non-updated values represents an address of a free space area and the corresponding update value represents an address of an allocated item, and wherein another of the non-updated values represents a size of the free space area and the corresponding update value represents a size of the allocated item.

4. The system of claim 1, wherein the destination variables include at least one of the following: a location in a random access memory, or a register in a processor.

5. The system of claim 1, wherein at least one of the following holds: at least one of the release barriers is implemented as part of a full memory barrier, or at least one of the acquire barriers is implemented as part of a full memory barrier.

6. The system of claim 1, wherein the values placing step and the values retrieving step are each performed without relying on any full memory barrier, a full memory barrier being a memory barrier which operates both as a release barrier and as an acquire barrier.

7. The system of claim 1, wherein the variables are each naturally aligned.

8. A method supporting lock-free conveyance of variable value updates as a unitary set, the method comprising a values placing step which performs the following actions in sequence:

replacing a non-updated value in an update status indication variable (Var1) of a set of variables with a different value (Z) that indicates updating is underway;

executing a first release barrier;

replacing non-updated values of other variables of the set with respective update values;

executing a second release barrier; and

replacing the value (Z) in the update status indication variable (Var1) with a different value that is also an update value;

whereby the method places update values in the variables without lock reliance, and the method places the update values as a unitary set in that after the values placing step actions are completed the values in the variables will not include both a non-updated value and an update value.

9. The method of claim 8, further comprising a values retrieving step which performs the following actions in sequence:

reading from the update status indication variable (Var1) into a first destination

status variable (DestVar1A);
executing a first acquire barrier;
reading from other variables of the set into respective destination variables;
executing a second acquire barrier;
reading from the update status indication variable (Var1) into a second destination status variable (DestVar1B); and
repeating the foregoing retrieving step actions if the first destination status variable (DestVar1A) has the value (Z) or if the first destination status variable (DestVar1A) and the second destination status variable (DestVar1B) contain different values from each other;
whereby the method retrieves the update values from the variables into respective destination variables without lock reliance, and the method conveys the update values as a unitary set in that after the values placing step and the values retrieving step are completed the values in the destination variables will not include both a non-updated value and an update value.

10. The method of claim 9, wherein the values placing step is performed by a thread (P) and the values retrieving step is performed by a different thread (R).

11. The method of claim 9, wherein one of the non-updated values represents an address of a free space area and the corresponding update value represents an address of an allocated item, and wherein another of the non-updated values represents a size of the free space area and the corresponding update value represents a size of the allocated item.

12. The method of claim 8, wherein replacing the non-updated value in the update status indication variable (Var1) with the value (Z) comprises zeroing the update status indication variable (Var1).

13. The method of claim 8, wherein the set of variables consists of two variables, wherein the method executes in a device containing a digital item, and wherein each of the variables in the set represents a respective characteristic of the digital item.

14. The method of claim 8, wherein the values placing step is performed without implementing any of the release barriers as part of a full memory barrier.

15. The method of claim 8, wherein each replacing action is individually atomic.