Claims

1. Processor for processing an audio signal ~~(200)~~, comprising:

an analyzer ~~(202)~~ for deriving a window control signal ~~(204)~~ from the audio signal ~~(200)~~ indicating a change from a first asymmetric window ~~(1400)~~ to a second window ~~(1402)~~ or for indicating a change from a third window ~~(1450)~~ to a fourth asymmetric window ~~(1452)~~, wherein the second window ~~(1402)~~ is shorter than the first window ~~(1400)~~, or wherein the third window ~~(1450)~~ is shorter than the fourth window ~~(1452)~~;

a window constructor ~~(206)~~ for constructing the second window ~~(1402)~~ using a first overlap portion ~~(800)~~ of the first asymmetric window ~~(1400)~~, wherein the window constructor ~~(206)~~ is configured to determine a first overlap portion ~~(1000)~~ of the second window ~~(1402)~~ using a truncated first overlap portion of the first asymmetric window, or wherein the window constructor is configured to calculate a second overlap portion ~~(1330)~~ of the third window ~~(1450)~~ using a truncated second overlap portion ~~(814)~~ of the fourth asymmetric window ~~(1452)~~; and

a windower ~~(208)~~ for applying the first and second windows or the third and fourth windows to obtain windowed audio signal portions ~~(210)~~.

2. Processor of claim 1,

wherein the first and second windows are analysis windows or the third and fourth windows are synthesis windows,

wherein the processor further comprises an audio encoder ~~(110)~~ for further processing samples windowed by the first and second windows, or wherein the processor further comprises an overlap-adder ~~(174)~~ for overlap-adding samples windowed by the third and fourth windows.

3. Processor of claim 1 or 2,

wherein the window constructor ~~(206)~~ is configured to derive the first overlap portion ~~(1000)~~ of the second window by truncating the first overlap portion of the first window and by fading-in the truncated portion, or

wherein the window constructor ~~(206)~~ is configured to derive the second overlap portion of the third window by truncating the second overlap portion of the fourth window and by fading-out the truncated portion.

4. Processor of claim 3,

wherein the window constructor ~~(206)~~ is configured for performing the fade-in or the fade-out using a sine fade-in function or a sine fade-out function.

5. Processor of claim 3 or 4,

wherein the window constructor ~~(206)~~ is configured to calculate the fade-in or fade-out using an overlap portion of any other window used by the processor.

6. Processor of claim 5,

wherein the window constructor ~~(206)~~ is configured to calculate the fade-in or fade-out ~~(304)~~ using a shortest overlap portion ~~(401)~~ of all overlap portions used.

7. Processor of one of the preceding claims, further comprising a memory ~~(300)~~ having stored thereon, for a certain sampling rate, the first overlap portion ~~(800)~~ of the first asymmetric window, a second overlap portion ~~(802)~~ of the first asymmetric window and a third overlap portion for a further window shorter than the first window,

wherein the window constructor ~~(206)~~ is configured for retrieving ~~(308)~~ the first overlap portion of the first asymmetric window from the memory ~~(300)~~, for truncating ~~(302)~~ the first overlap portion to a length shorter than the length of the first overlap portion, for retrieving ~~(312)~~ the third overlap portion, and

for multiplying ~~(304)~~ the truncated first portion by the third overlap portion to generate the first overlap portion of the second window; or

wherein the window constructor ~~(206)~~ is configured

for retrieving the second overlap portion of the fourth asymmetric window from the memory ~~(300)~~,

for truncating the second overlap portion retrieved to a length shorter than the length of the second overlap portion ~~(302)~~,

for retrieving the third overlap portion ~~(312)~~; and

for multiplying the truncated second overlap portion by the third overlap portion to generate the second overlap portion of the third window.

8. Processor of claim 7,

wherein the memory ~~(300)~~ has furthermore stored a fourth overlap portion ~~(402)~~ of an even further window, the even further window having a length between a length of the first window and a length of the further window.

9. Processor of claim 8,

wherein the window constructor ~~(206)~~ is configured to construct, depending on the window control signal ~~(204)~~, a sequence comprising the first window ~~(1400)~~, the second window ~~(1402)~~, an additional window constructed using the third overlap portion and the fourth overlap portion or using the third overlap portion only, and a further additional window ~~(1410)~~ using the third overlap portion and the second overlap portion of the first window.

10. Processor of one of the preceding claims,

wherein the window constructor ~~(206)~~ is configured to determine the first overlap portion of the second window ~~(1402)~~ using the truncated first overlap portion of the first asymmetric window ~~(1400)~~ being truncated to a length of a second overlap portion of the first asymmetric window, or

to determine the second overlap portion of the third window using a second overlap portion of the fourth window truncated to a length of the first overlap portion of the fourth asymmetric window.

11. Processor of one of the preceding claims,

wherein the window constructor ~~(206)~~ is configured to determine the second window using the first overlap portion of the second window ~~(1402)~~ and a second overlap portion ~~(1002)~~ of the second window ~~(1402)~~ corresponding to a first overlap portion ~~(1022)~~ of a further window following the second window, or

wherein the window constructor ~~(206)~~ is configured to construct the third window by using a first overlap portion ~~(1331)~~ of the third windowe corresponding to a second overlap portion ~~(1342)~~ of a further window preceding the third window ~~(1450)~~.

12. Processor of one of the preceding claims,

wherein the window constructor ~~(206)~~ is configured to truncate the first overlap portion of the first asymmetric window or the second overlap portion of the fourth asymmetric window to a truncation length being shorter or equal than a window length of the second or third window less a length of the first overlap portion of a further window following the second window or a length of a second overlap portion of a further window preceding the third window ~~(706, 710)~~.

13. Processor of claim 12,

wherein, when the truncation length is smaller than the window length less the length of the first overlap portion of the further window or the second overlap portion of the window, the window constructor ~~(206)~~ is configured to insert ~~(712)~~ zeroes ~~(1131, 1133)~~ before or subsequent to the first and second overlap portions of the second or third window, and wherein the window constructor is furthermore configured to insert a number of "1" values ~~(714; 1132)~~ between the first and second overlap portions of the second window or the third window.

14. Processor of one of claims 1 to 3,

wherein the first asymmetric window has a first overlap portion, a second overlap portion, a first high value part between the first and second overlap portion and a second low value part subsequent to the second overlap portion, wherein the values in the high value part are greater than 0.9 and the values in the low value part are lower than 0.1, and

wherein the length of the second overlap portion is lower than a length of the first overlap portion.

15. Processor of one of the preceding claims, which is configured to operate at a plurality of different sampling rates, and

wherein the processor is configured to store, for each sampling rate, the first and second overlap portions of the first or fourth window ~~(403, 404)~~, a symmetric overlap portion of a further window ~~(402)~~, and a further symmetric overlap portion ~~(401)~~ of an even further window being shorter than the further window; and

wherein the symmetric overlap portion and the further symmetric overlap portion are stored as an ascending or a descending portion only, and wherein the window constructor ~~(206)~~ is configured to derive a descending or an ascending portion from the stored ascending or descending portion by arithmetic or logic operations.

16. Processor of one of the preceding claims,

wherein the first window is configured for a transform length of 20 ms, wherein the window constructor is configured for further using further windows for transform length of 10 ms or 5 ms, and

wherein the second window is a transition window from the transform length of 20 ms to the transform length of 10 ms or 5 ms, or

wherein the fourth window is configured for the transform length of 20 ms, and wherein the third window is a transition window from the transform length of 5 ms to 20 ms or from the transform length of 10 ms to 20 ms.

17. Method of processing an audio signal ~~(200)~~, comprising:

deriving ~~(202)~~ a window control signal ~~(204)~~ from the audio signal ~~(200)~~ indicating a change from a first asymmetric window ~~(1400)~~ to a second window ~~(1402)~~ or for indicating a change from a third window ~~(1450)~~ to a fourth asymmetric window ~~(1452)~~, wherein the second window ~~(1402)~~ is shorter than the first window ~~(1400)~~, or wherein the third window ~~(1450)~~ is shorter than the fourth window ~~(1452)~~;

constructing ~~(206)~~ the second window ~~(1402)~~ using a first overlap portion ~~(800)~~ of the first asymmetric window ~~(1400)~~, wherein the window constructor ~~(206)~~ is configured to determine a first overlap portion ~~(1000)~~ of the second window ~~(1402)~~ using a truncated first overlap portion of the first asymmetric window, or wherein the window constructor is configured to calculate a second overlap portion ~~(1330)~~ of the third window ~~(1450)~~ using a truncated second overlap portion ~~(814)~~ of the fourth asymmetric window ~~(1452)~~; and

~~a windower (208) for~~ applying the first and second windows or the third and fourth windows to obtain windowed audio signal portions ~~(210)~~.

18. Computer program for performing, when running on a computer or a processor, the method of claim 17.