3-ANNEX C (informative)

THE ENCODING PROCESS

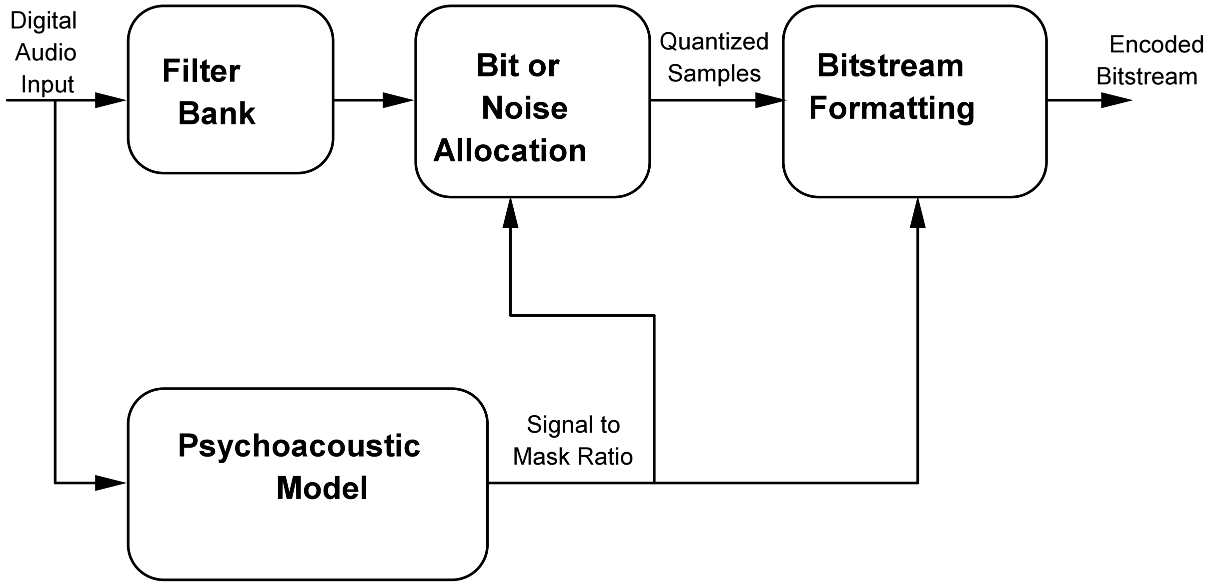
3-C.1 Encoder

3-C.1.1 Overview

For each of the Layers, an example of one suitable encoder with the corresponding flow-diagram is given in this annex. In subsequent clauses the analysis subband filter and the layer-specific encoding techniques are described. In Annex D two examples of psychoacoustic models, which are common to all layers, are described. A short introduction describes the overall philosophy.

INTRODUCTION

The MPEG-Audio algorithm is a psychoacoustic algorithm. The figure below shows the primary parts of a psychoacoustic algorithm.



The four primary parts of the psychoacoustic encoder are:

1) The Filterbank:
The filterbank does a time to frequency mapping. There are two filterbanks used in the MPEG-Audio algorithm, each providing a specific mapping in time and frequency. These filterbanks are critically sampled (i.e. there are as many samples in the analyzed domain as there are in the time domain). These filterbanks provide the primary frequency separation for the encoder, and the reconstruction filters for the decoder. The output samples of the filterbank are quantized.

2) The Psychoacoustic Model:
The psychoacoustic model calculates a just noticable noise-level for each band in the filterbank. This noise level is used in the bit or noise allocation to determine the actual quantizers and quantizer levels. There are two psychoacoustic models presented in 3-Annex D. While they can both be applied to any layer of the MPEG-Audio algorithm, in practice Model 1 has been used for Layers I and II, and Model 2 for Layer III. In both psychoacoustic models, the final output of the model is a signal-to-mask ratio (SMR) for each band (Layers I and II) or group of bands (Layer III).

3) Bit or Noise Allocation:

The allocator looks at both the output samples from the   and the SMR's from the psychoacoustic model, and adjusts the bit allocation (Layers I and II) or noise allocation (Layer III) in order to simultaneously meet both the bitrate requirements and the masking requirements. At low bitrates, these methods attempt to spend bits in a fashion that is psychoacousticly inoffensive when they cannot meet the psychoacoustic demand at the required bitrate.

4) The bitstream formatter:
The bitstream formatter takes the quantized filterbank outputs, the bit allocation (Layers I and II) or noise allocation (Layer III) and other required side information, and encodes and formats that information in an efficient fashion. In the case of Layer III, the Huffman codes are also inserted at this point.

The Filterbank

In Layers I and II, a filterbank with 32 subbands is used. In each subband, 12 or 36 samples are grouped for processing. In Layer III, the filterbank has a signal-dependant resolution, where there are either 6x32 or 18x32 frequency bands. In the case where there are 6x32 frequency samples, the 3 sets of each frequency are quantized separately.

Bit or Noise Allocation Method

There are two different bitrate control methods explained in this Annex. In Layers I and II this method is a bit allocation process, i.e. a number of bits is assigned to each sample (or group of samples) in each subband. The method for Layer III is a noise-allocation loop, where the quantizers are varied in an organized fashion, and the variable to be controlled is the actually injected noise. In either case, the result is a set of quantization parameters and quantized output samples that are given to the bistream formatter.

Bitstream Formatting

The bitstream formatter varies from layer to layer. In Layers I and II, a fixed PCM code is used for each subband sample, with the exception that in Layer II quantized samples may be grouped. In Layer III, Huffman codes are used to represent the quantized frequency samples. These Huffman codes are variable-length codes that allow for more efficient bitstream representation of the quantized samples at the cost of additional complexity.

3-C.1.2 Input High-Pass Filter

The encoding algorithms provide a frequency response down to DC. However, in applications where this is not a requirement, it is recommended that a high-pass filter be included at the input of the encoder. The cut-off frequency should be in the range of 2 to 10Hz.

The application of such a high-pass filter avoids an unneccessarily high bitrate requirement for the lowest subband and increases the overall audio quality.

3-C.1.3 Analysis Subband Filter

An analysis subband filterbank is used to split the broadband signal with sampling frequency fs into 32 equally spaced subbands with sampling frequencies fs/32. The flow chart of this process with the appropriate formulas is given in Figure 3-C.1 "ANALYSIS SUBBAND FILTER FLOW CHART". The analysis subband filtering includes the following steps:

- Input 32 audio samples.
- Build an input sample vector, X, of 512 elements. The 32 audiosamples are shifted in at positions 0 to 31, the most recent on at position 0, and the 32 oldestelements are shifted out.

- Window vector X by vector C. The coefficients are to be found in Table 3-C.1"COEFFICIENTS C_i FOR THE ANALYSIS WINDOW".
- Calculate the 64 values Y_i according to the formula given in theflow chart.
- Calculate the 32 subband samples S_i by matrixing. The coefficients for the matrix can be calculated by the following formula:

  $M_{ik} = \cos [(2i + 1)(k - 16)p/64]$, for $i = 0$ to 31, and $k = 0$ to 63.

Table 3-C.1 Coefficients C_i of the Analysis Window

C[0]= 0.000000000	C[1]=-0.000000477	C[2]=-0.000000477	C[3]=-0.000000477
C[4]=-0.000000477	C[5]=-0.000000477	C[6]=-0.000000477	C[7]=-0.000000954
C[8]=-0.000000954	C[9]=-0.000000954	C[10]=-0.000000954	C[11]=-0.000001431
C[12]=-0.000001431	C[13]=-0.000001907	C[14]=-0.000001907	C[15]=-0.000002384
C[16]=-0.000002384	C[17]=-0.000002861	C[18]=-0.000003338	C[19]=-0.000003338
C[20]=-0.000003815	C[21]=-0.000004292	C[22]=-0.000004768	C[23]=-0.000005245
C[24]=-0.000006199	C[25]=-0.000006676	C[26]=-0.000007629	C[27]=-0.000008106
C[28]=-0.000009060	C[29]=-0.000010014	C[30]=-0.000011444	C[31]=-0.000012398
C[32]=-0.000013828	C[33]=-0.000014782	C[34]=-0.000016689	C[35]=-0.000018120
C[36]=-0.000019550	C[37]=-0.000021458	C[38]=-0.000023365	C[39]=-0.000025272
C[40]=-0.000027657	C[41]=-0.000030041	C[42]=-0.000032425	C[43]=-0.000034809
C[44]=-0.000037670	C[45]=-0.000040531	C[46]=-0.000043392	C[47]=-0.000046253
C[48]=-0.000049591	C[49]=-0.000052929	C[50]=-0.000055790	C[51]=-0.000059605
C[52]=-0.000062943	C[53]=-0.000066280	C[54]=-0.000070095	C[55]=-0.000073433
C[56]=-0.000076771	C[57]=-0.000080585	C[58]=-0.000083923	C[59]=-0.000087261
C[60]=-0.000090599	C[61]=-0.000093460	C[62]=-0.000096321	C[63]=-0.000099182
C[64]= 0.000101566	C[65]= 0.000103951	C[66]= 0.000105858	C[67]= 0.000107288
C[68]= 0.000108242	C[69]= 0.000108719	C[70]= 0.000108719	C[71]= 0.000108242
C[72]= 0.000106812	C[73]= 0.000105381	C[74]= 0.000102520	C[75]= 0.000099182
C[76]= 0.000095367	C[77]= 0.000090122	C[78]= 0.000084400	C[79]= 0.000077724
C[80]= 0.000069618	C[81]= 0.000060558	C[82]= 0.000050545	C[83]= 0.000039577
C[84]= 0.000027180	C[85]= 0.000013828	C[86]=-0.000000954	C[87]=-0.000017166
C[88]=-0.000034332	C[89]=-0.000052929	C[90]=-0.000072956	C[91]=-0.000093937
C[92]=-0.000116348	C[93]=-0.000140190	C[94]=-0.000165462	C[95]=-0.000191212
C[96]=-0.000218868	C[97]=-0.000247478	C[98]=-0.000277042	C[99]=-0.000307560
C[100]=-0.000339031	C[101]=-0.000371456	C[102]=-0.000404358	C[103]=-0.000438213
C[104]=-0.000472546	C[105]=-0.000507355	C[106]=-0.000542164	C[107]=-0.000576973
C[108]=-0.000611782	C[109]=-0.000646591	C[110]=-0.000680923	C[111]=-0.000714302
C[112]=-0.000747204	C[113]=-0.000779152	C[114]=-0.000809669	C[115]=-0.000838757
C[116]=-0.000866413	C[117]=-0.000891685	C[118]=-0.000915051	C[119]=-0.000935555
C[120]=-0.000954151	C[121]=-0.000968933	C[122]=-0.000980854	C[123]=-0.000989437
C[124]=-0.000994205	C[125]=-0.000995159	C[126]=-0.000991821	C[127]=-0.000983715
C[128]= 0.000971317	C[129]= 0.000953674	C[130]= 0.000930786	C[131]= 0.000902653
C[132]= 0.000868797	C[133]= 0.000829220	C[134]= 0.000783920	C[135]= 0.000731945

C[136]= 0.000674248 C[137]= 0.000610352 C[138]= 0.000539303 C[139]=
0.000462532
C[140]= 0.000378609 C[141]= 0.000288486 C[142]= 0.000191689 C[143]=
0.000088215
C[144]=-0.000021458 C[145]=-0.000137329 C[146]=-0.000259876 C[147]=-
0.000388145
C[148]=-0.000522137 C[149]=-0.000661850 C[150]=-0.000806808 C[151]=-
0.000956535
C[152]=-0.001111031 C[153]=-0.001269817 C[154]=-0.001432419 C[155]=-
0.001597881
C[156]=-0.001766682 C[157]=-0.001937389 C[158]=-0.002110004 C[159]=-
0.002283096
C[160]=-0.002457142 C[161]=-0.002630711 C[162]=-0.002803326 C[163]=-
0.002974033
C[164]=-0.003141880 C[165]=-0.003306866 C[166]=-0.003467083 C[167]=-
0.003622532
C[168]=-0.003771782 C[169]=-0.003914356 C[170]=-0.004048824 C[171]=-
0.004174709
C[172]=-0.004290581 C[173]=-0.004395962 C[174]=-0.004489899 C[175]=-
0.004570484
C[176]=-0.004638195 C[177]=-0.004691124 C[178]=-0.004728317 C[179]=-
0.004748821
C[180]=-0.004752159 C[181]=-0.004737377 C[182]=-0.004703045 C[183]=-
0.004649162
C[184]=-0.004573822 C[185]=-0.004477024 C[186]=-0.004357815 C[187]=-
0.004215240
C[188]=-0.004049301 C[189]=-0.003858566 C[190]=-0.003643036 C[191]=-
0.003401756
C[192]= 0.003134727 C[193]= 0.002841473 C[194]= 0.002521515 C[195]=
0.002174854
C[196]= 0.001800537 C[197]= 0.001399517 C[198]= 0.000971317 C[199]=
0.000515938
C[200]= 0.000033379 C[201]=-0.000475883 C[202]=-0.001011848 C[203]=-
0.001573563
C[204]=-0.002161503 C[205]=-0.002774239 C[206]=-0.003411293 C[207]=-
0.004072189
C[208]=-0.004756451 C[209]=-0.005462170 C[210]=-0.006189346 C[211]=-
0.006937027
C[212]=-0.007703304 C[213]=-0.008487225 C[214]=-0.009287834 C[215]=-
0.010103703
C[216]=-0.010933399 C[217]=-0.011775017 C[218]=-0.012627602 C[219]=-
0.013489246
C[220]=-0.014358521 C[221]=-0.015233517 C[222]=-0.016112804 C[223]=-
0.016994476
C[224]=-0.017876148 C[225]=-0.018756866 C[226]=-0.019634247 C[227]=-
0.020506859
C[228]=-0.021372318 C[229]=-0.022228718 C[230]=-0.023074150 C[231]=-
0.023907185
C[232]=-0.024725437 C[233]=-0.025527000 C[234]=-0.026310921 C[235]=-
0.027073860
C[236]=-0.027815342 C[237]=-0.028532982 C[238]=-0.029224873 C[239]=-
0.029890060
C[240]=-0.030526638 C[241]=-0.031132698 C[242]=-0.031706810 C[243]=-
0.032248020
C[244]=-0.032754898 C[245]=-0.033225536 C[246]=-0.033659935 C[247]=-
0.034055710
C[248]=-0.034412861 C[249]=-0.034730434 C[250]=-0.035007000 C[251]=-
0.035242081
C[252]=-0.035435200 C[253]=-0.035586357 C[254]=-0.035694122 C[255]=-
0.035758972
C[256]= 0.035780907 C[257]= 0.035758972 C[258]= 0.035694122 C[259]=
0.035586357
C[260]= 0.035435200 C[261]= 0.035242081 C[262]= 0.035007000 C[263]=
0.034730434
C[264]= 0.034412861 C[265]= 0.034055710 C[266]= 0.033659935 C[267]=
0.033225536
C[268]= 0.032754898 C[269]= 0.032248020 C[270]= 0.031706810 C[271]=
0.031132698
C[272]= 0.030526638 C[273]= 0.029890060 C[274]= 0.029224873 C[275]=
0.028532982
C[276]= 0.027815342 C[277]= 0.027073860 C[278]= 0.026310921 C[279]=
0.025527000
C[280]= 0.024725437 C[281]= 0.023907185 C[282]= 0.023074150 C[283]=
0.022228718
C[284]= 0.021372318 C[285]= 0.020506859 C[286]= 0.019634247 C[287]=
0.018756866
C[288]= 0.017876148 C[289]= 0.016994476 C[290]= 0.016112804 C[291]=
0.015233517
C[292]= 0.014358521 C[293]= 0.013489246 C[294]= 0.012627602 C[295]=
0.011775017

C[296]= 0.010933399 C[297]= 0.010103703 C[298]= 0.009287834 C[299]=
0.008487225
C[300]= 0.007703304 C[301]= 0.006937027 C[302]= 0.006189346 C[303]=
0.005462170
C[304]= 0.004756451 C[305]= 0.004072189 C[306]= 0.003411293 C[307]=
0.002774239
C[308]= 0.002161503 C[309]= 0.001573563 C[310]= 0.001011848 C[311]=
0.000475883
C[312]=-0.000033379 C[313]=-0.000515938 C[314]=-0.000971317 C[315]=-
0.001399517
C[316]=-0.001800537 C[317]=-0.002174854 C[318]=-0.002521515 C[319]=-
0.002841473
C[320]= 0.003134727 C[321]= 0.003401756 C[322]= 0.003643036 C[323]=
0.003858566
C[324]= 0.004049301 C[325]= 0.004215240 C[326]= 0.004357815 C[327]=
0.004477024
C[328]= 0.004573822 C[329]= 0.004649162 C[330]= 0.004703045 C[331]=
0.004737377
C[332]= 0.004752159 C[333]= 0.004748821 C[334]= 0.004728317 C[335]=
0.004691124
C[336]= 0.004638195 C[337]= 0.004570484 C[338]= 0.004489899 C[339]=
0.004395962
C[340]= 0.004290581 C[341]= 0.004174709 C[342]= 0.004048824 C[343]=
0.003914356
C[344]= 0.003771782 C[345]= 0.003622532 C[346]= 0.003467083 C[347]=
0.003306866
C[348]= 0.003141880 C[349]= 0.002974033 C[350]= 0.002803326 C[351]=
0.002630711
C[352]= 0.002457142 C[353]= 0.002283096 C[354]= 0.002110004 C[355]=
0.001937389
C[356]= 0.001766682 C[357]= 0.001597881 C[358]= 0.001432419 C[359]=
0.001269817
C[360]= 0.001111031 C[361]= 0.000956535 C[362]= 0.000806808 C[363]=
0.000661850
C[364]= 0.000522137 C[365]= 0.000388145 C[366]= 0.000259876 C[367]=
0.000137329
C[368]= 0.000021458 C[369]=-0.000088215 C[370]=-0.000191689 C[371]=-
0.000288486
C[372]=-0.000378609 C[373]=-0.000462532 C[374]=-0.000539303 C[375]=-
0.000610352
C[376]=-0.000674248 C[377]=-0.000731945 C[378]=-0.000783920 C[379]=-
0.000829220
C[380]=-0.000868797 C[381]=-0.000902653 C[382]=-0.000930786 C[383]=-
0.000953674
C[384]= 0.000971317 C[385]= 0.000983715 C[386]= 0.000991821 C[387]=
0.000995159
C[388]= 0.000994205 C[389]= 0.000989437 C[390]= 0.000980854 C[391]=
0.000968933
C[392]= 0.000954151 C[393]= 0.000935555 C[394]= 0.000915051 C[395]=
0.000891685
C[396]= 0.000866413 C[397]= 0.000838757 C[398]= 0.000809669 C[399]=
0.000779152
C[400]= 0.000747204 C[401]= 0.000714302 C[402]= 0.000680923 C[403]=
0.000646591
C[404]= 0.000611782 C[405]= 0.000576973 C[406]= 0.000542164 C[407]=
0.000507355
C[408]= 0.000472546 C[409]= 0.000438213 C[410]= 0.000404358 C[411]=
0.000371456
C[412]= 0.000339031 C[413]= 0.000307560 C[414]= 0.000277042 C[415]=
0.000247478
C[416]= 0.000218868 C[417]= 0.000191212 C[418]= 0.000165462 C[419]=
0.000140190
C[420]= 0.000116348 C[421]= 0.000093937 C[422]= 0.000072956 C[423]=
0.000052929
C[424]= 0.000034332 C[425]= 0.000017166 C[426]= 0.000000954 C[427]=-
0.000013828
C[428]=-0.000027180 C[429]=-0.000039577 C[430]=-0.000050545 C[431]=-
0.000060558
C[432]=-0.000069618 C[433]=-0.000077724 C[434]=-0.000084400 C[435]=-
0.000090122
C[436]=-0.000095367 C[437]=-0.000099182 C[438]=-0.000102520 C[439]=-
0.000105381
C[440]=-0.000106812 C[441]=-0.000108242 C[442]=-0.000108719 C[443]=-
0.000108719
C[444]=-0.000108242 C[445]=-0.000107288 C[446]=-0.000105858 C[447]=-
0.000103951
C[448]= 0.000101566 C[449]= 0.000099182 C[450]= 0.000096321 C[451]=
0.000093460
C[452]= 0.000090599 C[453]= 0.000087261 C[454]= 0.000083923 C[455]=
0.000080585

```
   C[456]= 0.000076771    C[457]= 0.000073433    C[458]= 0.000070095    C[459]=
0.000066280
   C[460]= 0.000062943    C[461]= 0.000059605    C[462]= 0.000055790    C[463]=
0.000052929
   C[464]= 0.000049591    C[465]= 0.000046253    C[466]= 0.000043392    C[467]=
0.000040531
   C[468]= 0.000037670    C[469]= 0.000034809    C[470]= 0.000032425    C[471]=
0.000030041
   C[472]= 0.000027657    C[473]= 0.000025272    C[474]= 0.000023365    C[475]=
0.000021458
   C[476]= 0.000019550    C[477]= 0.000018120    C[478]= 0.000016689    C[479]=
0.000014782
   C[480]= 0.000013828    C[481]= 0.000012398    C[482]= 0.000011444    C[483]=
0.000010014
   C[484]= 0.000009060    C[485]= 0.000008106    C[486]= 0.000007629    C[487]=
0.000006676
   C[488]= 0.000006199    C[489]= 0.000005245    C[490]= 0.000004768    C[491]=
0.000004292
   C[492]= 0.000003815    C[493]= 0.000003338    C[494]= 0.000003338    C[495]=
0.000002861
   C[496]= 0.000002384    C[497]= 0.000002384    C[498]= 0.000001907    C[499]=
0.000001907
   C[500]= 0.000001431    C[501]= 0.000001431    C[502]= 0.000000954    C[503]=
0.000000954
   C[504]= 0.000000954    C[505]= 0.000000954    C[506]= 0.000000477    C[507]=
0.000000477
   C[508]= 0.000000477    C[509]= 0.000000477    C[510]= 0.000000477    C[511]=
0.000000477
```

3-C.1.4 Psychoacoustic Models

Two examples of psychoacoustic models are presented in Annex D, "PSYCHOACOUSTIC MODELS".

3-C.1.5 Encoding

3-C.1.5.1 Layer I Encoding

1. Introduction

This clause describes a possible Layer I encoding method. The description is made according to Figure 3-C.2, "LAYERI, II ENCODER FLOW CHART".

2. Psychoacoustic Model

The calculation of the psychoacoustic parameters can be done either with Psychoacoustic Model I described in Annex D, clause 3-D.1. or with Psychoacoustic Model II as described in Annex D, clause 3-D.2. The FFT shiftlength equals 384 samples. Either model provides the signal-to-mask ratio for every subband.

3. Analysis Subband Filtering

The subband analysis is described in the clause 3-C.1.3, "ANALYSIS SUBBAND FILTER".

4. Scalefactor Calculation

The calculation of the scalefactor for each subband is performed every 12 subband samples. The maximum of the absolute value of these 12 samples is determined. The next largest value in 3-Annex B, Table 3-B.1., "LAYER I, II SCALEFACTORS" is used as the scalefactor.

5. Coding of Scalefactors

The index in the 3-Annex B, Table 3-B.1., "LAYER I ,II SCALEFACTORS" is represented by 6 bits, MSB first. The scalefactor is transmitted only if a non-zero number of bits has been allocated to the subband.

6. Bit Allocation

Before adjustment to a fixed bitrate, the number of bits that are available for coding the samples and the scalefactors must be determined. This number can be obtained by subtracting from the total number of bits available "cb", the numbers of bits needed for bit allocation "bbal", and the number of bits required for ancillary data "banc":

adb = cb - (bbal + banc)

The resulting number of bits can be used to code the subband samples and the scalefactors. The principle used in the allocation procedure is minimization of the total noise-to-mask ratio over the frame with the constraint that the number of bits used does not exceed the number of bits available for that frame. The possible number of bits allocated to one sample can be found in the table in clause 2.4.2.5 of the main part of the audio standard (Audio data, LayerI); the range is 0...15 bits, excluding an allocation of 1 bit.

The allocation procedure is an iterative procedure, where in each iteration step the number of levels of the subband samples of greatest benefit is increased.

First the mask-to-noise ratio "MNR" for each subband is calculated by subtracting from the signal-to-noise-ratio "SNR" the signal-to-mask-ratio "SMR":

MNR = SNR - SMR

The signal-to-noise-ratio can be found in the 3-Annex C, Table 3-C.2., "LAYER I SIGNAL-TO-NOISE-RATIOS". The signal-to-mask-ratio is the output of the psychoacoustic model.

Then zero bits are allocated to the samples and the scalefactors. The number of bits for the samples "bspl" and the number of bits for the scalefactors "bscf" are set to zero. Next an iterative procedure is started. Each iteration loop contains the following steps :

- Determination of the minimal MNR of all subbands.
- The accuracy of the quantization of the subband with the minimal MNR is increased by using the next higher number of bits.
- The new MNR of this subband is calculated.
- bspl is updated according to the additional number of bits required. If a non-zero number of bits is assigned to a subband for the first time, bscf has to be incremented by 6 bits. Then adb is calculated again using the formula: adb=cb-(bbal+bscf+bspl+banc)

The iterative procedure is repeated as long as adb is not less than any possible increase of bspl and bscf within one loop.

7. Quantization and Encoding of Subband Samples

A linear quantizer with a symmetric zero representation is used to quantize the subband samples. This representation prevents small value changes around zero from quantizing to different levels. Each of the subband samples is normalized by dividing its value by the scalefactor to obtain X, and quantized using the following formula :

- Calculate AX+B
- Take the N most significant bits.
- Invert the MSB.

A and B can be found in 3-Annex C, Table 3-C.3, "LAYER I QUANTIZATION COEFFICIENTS". N represents the necessary number of bits to encode the number of steps. The inversion of the most significant bit (MSB) is done in order to avoid the all ´1´ representation of the code, because the all ´1´ code is used for the synchronization word.

8. Coding of Bit Allocation

The 4-bit code for the allocation is given in clause 2.4.2.5, "Audio data LayerI", of the main part of the audio standard.

9. Formatting

The encoded subband information is transferred in frames (See also clauses 2.4.1.2, 2.4.1.3, 2.4.1.5 and 2.4.1.8 of the clause 2.4.1 "Specification of the Coded Audio Bitstream Syntax " of the main part of the audio standard. The number of slots in a frame varies with the sample frequency (Fs) and bitrate. Each frame contains information on 384 samples of the original input signal, so the frame rate is Fs/384.

Fs (kHz)	Frame size (ms)
48	8
44.1	8.7074...
32	12

A frame may carry audio information from one or two channels.

The length of a slot in LayerI is 32 bits. The number of slots in a frame can be computed by this formula :

Number of slots/frame (N) = * 12

If this does not give an integer number the result is truncated and 'padding' is required. This means that the number of slots may vary between N and N + 1.

An overview of the Layer I format is given below:

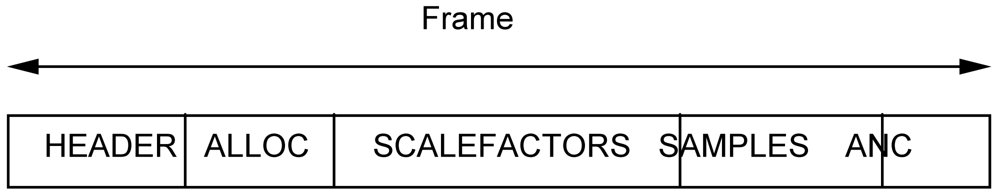


TABLE 3-C.2 LAYER I SIGNAL-TO-NOISE RATIOS

No. of steps	**SNR (dB)**
0	0.00
3	7.00
7	16.00
15	25.28
31	31.59
63	37.75
127	43.84
255	49.89
511	55.93
1023	61.96
2047	67.98
4095	74.01
8191	80.03
16383	86.05

32767 92.01

TABLE 3-C.3 LAYER I QUANTIZATION COEFFICIENTS

No. of steps	A	B
3	0.750000000	-0.250000000
7	0.875000000	-0.125000000
15	0.937500000	-0.062500000
31	0.968750000	-0.031250000
63	0.984375000	-0.015625000
127	0.992187500	-0.007812500
255	0.996093750	-0.003906250
511	0.998046875	-0.001953125
1023	0.999023438	-0.000976563
2047	0.999511719	-0.000488281
4095	0.999755859	-0.000244141
8191	0.999877930	-0.000122070
16383	0.999938965	-0.000061035
32767	0.999969482	-0.000030518

3-C.1.5.2 Layer II Encoding

1. Introduction

This clause describes a possible Layer II encoding method. The description is made according to Figure 3-C.2, "LAYERI, II ENCODER FLOWCHART".

2. Psychoacoustic Model

The calculation of the psychoacoustic parameters can be done either with Psychoacoustic Model I described in Annex D, clause 3-D.1. or with Psychoacoustic Model II described in Annex D, clause 3-D.2. If Psychoacoustic Model I is used to calculate the psychoacoutic parameters, the FFT shiftlength is 1152 samples. If Psychoacoustic Model II is used, the calculation is performed twice with a shiftlength of 576 samples and the largest of each pair of signal to mask ratios is used. Either model provides the signal-to-mask ratio for every subband.

3. Analysis Subband Filter

The analysis subband filter is described in clause 3-C.1.3, "ANALYSIS SUBBAND FILTER".

4. Scalefactor Calculation

The calculation of the scalefactor for each subband is performed every 12 subband samples. The maximum of the absolute value of these 12 samples is determined. The next largest value in 3-Annex B, Table 3-B.1., "LAYER I, II TABLE OF SCALEFACTORS" is the scalefactor.

5. Coding of Scalefactors

A frame corresponds to 36 subband samples and therefore contains three scalefactors per subband. Define 'scf' as the index in Annex B, Table 3-B.1., "LAYER I, II SCALEFACTORS". First, the two differences dscf1 and dscf2 of the successive scalefactor indices scf1, scf2 and scf3 are calculated:

dscf1 = scf1 - scf2
dscf2 = scf2 - scf3

The class of each of the differences is determined as follows:

class.	dscf
1	dscf <= -3

2 -3 < dscf < 0
3 dscf = 0
4 0 < dscf < 3
5 dscf >= 3

The pair of classes of differences indicate the entry point in Table 3-C.4., "LAYER II SCALEFACTOR TRANSMISSION PATTERNS". The "adjusted scalefactor pattern" gives the three scalefactors which are actually used. "1", "2" and "3" mean respectively the first, second and third scalefactor within a frame, "4" means the maximum of the three scalefactors. If, after this adjusting of scalefactors two or three are the same, not all scalefactors must be transmitted for a certain subband within one frame. Only the scalefactors indicated in the "transmission pattern" are transmitted. The information describing the number and the position of the scalefactors in each subband is called "scalefactor select information".

6. Coding of Scalefactor Select Information

The "scalefactor select information" (scfsi) is coded by a two bit word, which is also to be found in 3-ANNEX C, Table 3-C.4., "LAYER II SCALEFACTOR TRANSMISSION PATTERNS". Only the scfsi for the subbands which will get a nonzero bit allocation are transmitted.

7. Bit Allocation

Before adjustment to a fixed bitrate, the number of bits ,"adb", that are available for coding the samples and the scalefactors must be determined. This number can be obtained by subtracting from the total number of available bits "cb", the number of bits needed for bit allocation "bbal", and the number of bits "banc" required for ancillary data:

adb = cb - (bbal + banc)

The resulting number can be used to code the subband samples and the scalefactors. The principle used in the allocation procedure is minimization of the total noise-to-mask ratio over the frame with the constraint that the number of bits used does not exceed the number of bits available for that frame. Use is made of 3-Annex B, Table 3-B.2., "LAYER II POSSIBLE QUANTIZATIONS PER SUBBAND" that indicates for every subband the number of steps that may be used to quantize the samples. The number of bits required to represent these quantized samples can be derived from 3-Annex B, Table 3-B.4., "LAYER II CLASSES OF QUANTIZATION".

The allocation procedure is an iterative procedure where, in each iteration step the number of levels of the subband that has the greatest benefit is increased.

First the mask-to-noise ratio "MNR" for each subband is calculated by subtracting from the signal-to-noise-ratio "SNR" the signal-to-mask-ratio "SMR":

MNR = SNR - SMR

The signal-to-noise-ratio can be found in table 3-C.5. "LAYER II SIGNAL-TO-NOISE-RATIOS". The signal-to-mask-ratio is the output of the psychoacoustic model.

Then zero bits are allocated to the samples and the scalefactors. The number of bits for the samples "bspl" and the number of bits for the scalefactors "bscf" are set to zero. Next an iterative procedure is started. Each iteration loop contains the following steps :

- Determination of the minimal MNR of all subbands.

- The accuracy of the quantization of the subband with the minimal MNR is increased by using the next higher entry in the relevant Annex B, Table 3-B.2., "LAYER II POSSIBLE QUANTIZATIONS PER SUBBAND".

- The new MNR of this subband is calculated.

- bspl is updated according to the additional number of bits required. If a non-zero number of bits is assigned to a subband for the first time, bsel has to be updated, and bscf has to be updated according to the number of scalefactors required for this subband. Then adb is calculated again using the formula :

adb=cb-(bbal + bsel + bscf + bspl +banc)

The iterative procedure is repeated as long as adb is not less than any possible increase of bspl, bsel and bscf within one loop.

8. Quantization and Encoding of Subband Samples

Each of the 12 subband samples is normalized by dividing its value by the scale factor to obtain X and quantized using the following formula:

- Calculate A * X + B
- Take the N most significant bits.
- Invert the MSB

A and B can be found in the 3-ANNEX C, TABLE 3-C.6., "LAYER II QUANTIZATION COEFFICIENTS". Nrepresents the necessary number of bits to encode the number of steps. The inversion of the MSB is done in order to avoid the all '1' code that is used for the synchronization word.

Given the number of steps that the samples will be quantized to, 3-Annex B, Table 3-B.4., "LAYER II CLASSES OF QUANTIZATION" shows whether grouping will be used. If grouping is not required, the three samples are coded with individual codewords.

If grouping is required, three consecutive samples are coded as one codeword. Only one value vm, MSB first, is transmitted for this triplet. The relationships between the coded value vm (m=3,5,9) and the three consecutive subband samples x, y, z are:

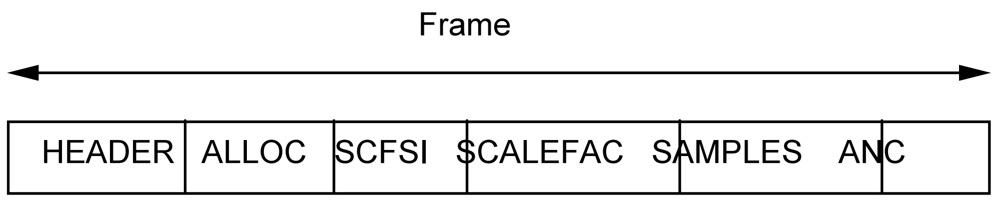
v3 = 9z + 3y + x (v3 in 0... 26)
v5 = 25z + 5y + x(v5 in 0...124)
v9 = 81z + 9y + x(v9 in 0...728)

9. Coding of Bit Allocation

For the purpose of a more efficient coding, only a limited number of possible quantizations, which may be different for each subband, are allowed. Only the index with wordlength "nbal" in the relevant Annex B, Table 3-B.2., "LAYER II POSSIBLE QUANTZATIONS PER SUBBAND" is transmitted, MSB first.

10. Formatting

An overview of the Layer II format can be seen as follows:



The differences compared to the Layer I format are:

- The length of a slot equals 8 bits.
- A new block scfsi containing the scalefactor select information has been introduced.
- The bit allocation information, scalefactors and samples have been subject to further coding (see the related clauses).

The details can be found in the clause 2.4.1 of the main part of this audio standard, "SPECIFICATION OF THE CODED AUDIO BITSTREAM SYNTAX " .

TABLE 3-C.4: LAYER II Scalefactor transmission patterns

Class1	Class2	Transmission pattern	Select Information
1	1	1 2 3 0	
1	2	1 2 2 3	
1	3	1 2 2 3	
1	4	1 3 3 3	
1	5	1 2 3 0	
2	1	1 1 3 1	
2	2	1 1 1 2	
2	3	1 1 1 2	
2	4	4 4 4 2	
2	5	1 1 3 1	
3	1	1 1 1 2	
3	2	1 1 1 2	
3	3	1 1 1 2	
3	4	3 3 3 2	
3	5	1 1 3 1	
4	1	2 2 2 2	
4	2	2 2 2 2	
4	3	2 2 2 2	
4	4	3 3 3 2	
4	5	1 2 3 0	
5	1	1 2 3 0	
5	2	1 2 2 3	
5	3	1 2 2 3	
5	4	1 3 3 3	
5	5	1 2 3 0	

TABLE 3-C.5: LAYER II SIGNAL-TO-NOISE RATIOS

No. of steps	SNR (dB)
0	0.00
3	7.00
5	11.00
7	16.00
9	20.84
15	25.28
31	31.59
63	37.75
127	43.84
255	49.89
511	55.93
1023	61.96
2047	67.98

4095	74.01
8191	80.03
16383	86.05
32767	92.01
65535	98.01

TABLE 3-C.6: LAYER II QUANTIZATION COEFFICIENTS

No. of steps	A	B
3	0.750000000	-0.250000000
5	0.625000000	-0.375000000
7	0.875000000	-0.125000000
9	0.562500000	-0.437500000
15	0.937500000	-0.062500000
31	0.968750000	-0.031250000
63	0.984375000	-0.015625000
127	0.992187500	-0.007812500
255	0.996093750	-0.003906250
511	0.998046875	-0.001953125
1023	0.999023438	-0.000976563
2047	0.999511719	-0.000488281
4095	0.999755859	-0.000244141
8191	0.999877930	-0.000122070
16383	0.999938965	-0.000061035
32767	0.999969482	-0.000030518
65535	0.999984741	-0.000015259

FIGURE 3-C.1 Analysis subband filter flow chart

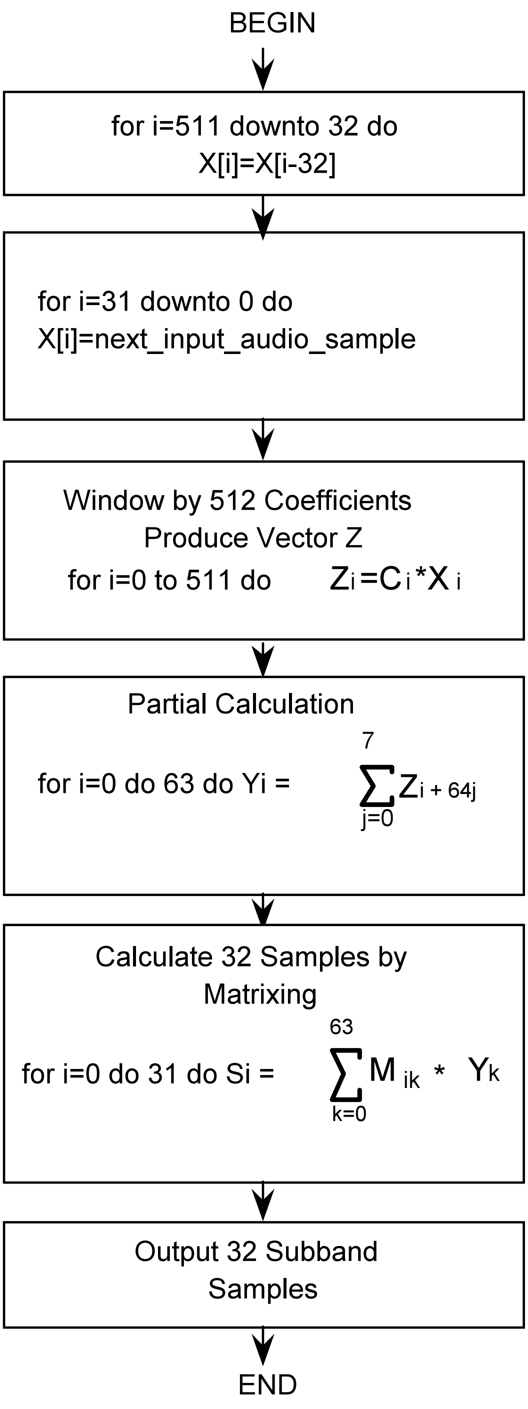
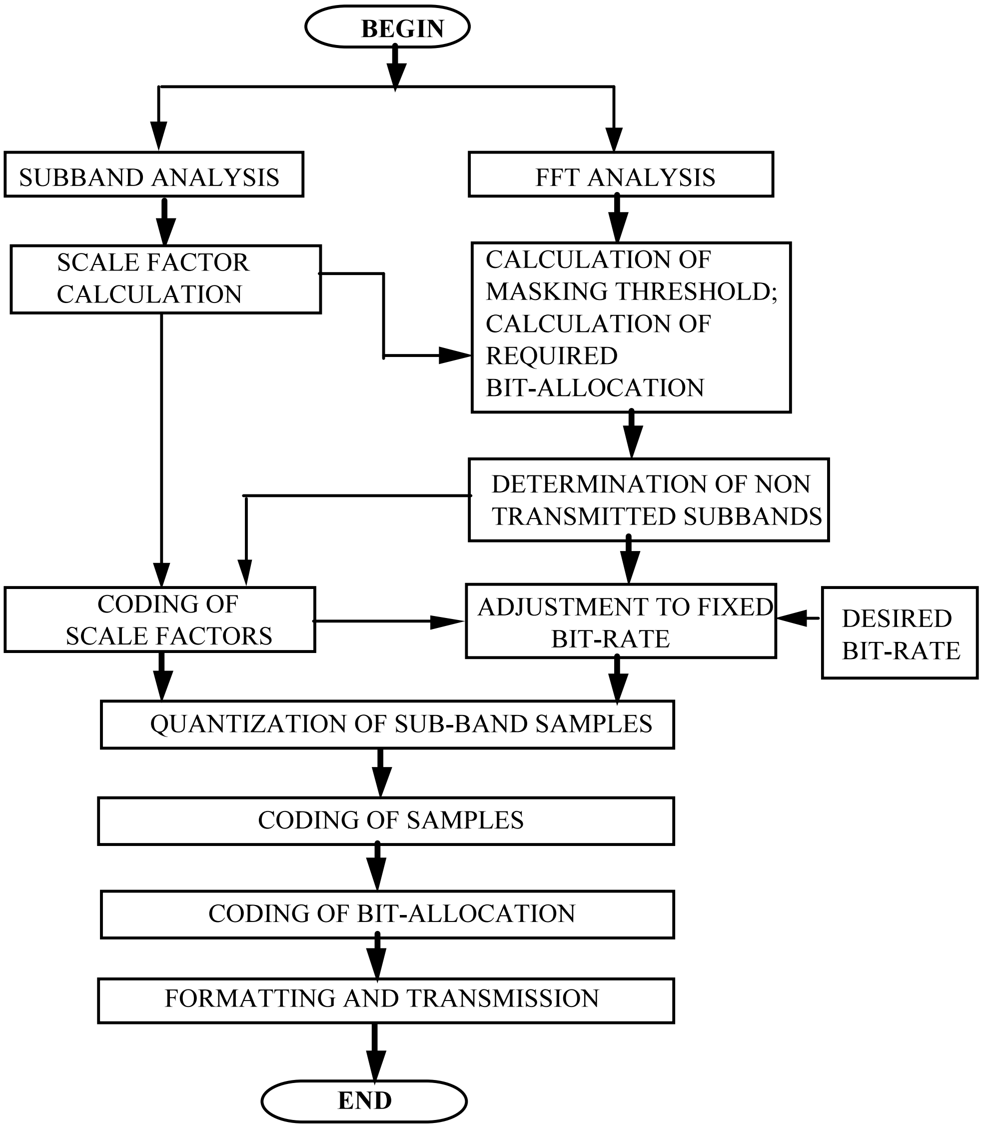


FIGURE 3-C.2 Layer I, II encoder flow chart



3-C.1.5.3 Layer III Encoding

1. Introduction

This clause describes a possible Layer III encoding method. The basic data flow is described by the general psychoacoustic coder block diagram. The basic blocks are described in more detail and below.

2. Psychoacoustic Model

The calculation of the psychoacoustic parameters can be done either with Psychoacoustic Model I described in Annex D, clause 3-D.1. or with Psychoacoustic Model II described in Annex D, clause 3-D.2. A description of modifications to Psychoacoustic Model II for use with Layer III can be found below. The model is run twice per block, using a shiftlength of 576 samples. A signal-to-mask-ratio is provided for every scale factor band

2.1. Adaptation of Psychoacoustic Model II for Layer III

For the use with Layer III encoding the psychoacoustic model 2 (Annex D, clause 3-D.2.) is modified as described below.

General Considerations:

The model is calculated twice in parallel. One computation is done with a shift length **iblen** of 192 samples (to be used with short blocks), the other is done with a shift length of 576 samples. For the shift length of 192 samples the block length of the FFT is changed to 256, and the parameters changed accordingly.

Change to Unpredictability Calculation:

The calculation of the unpredictability metric in Psychoacoustic Model II is changed.

- Calculation of the unpredictability
  The unpredictability cw is calculated for the first 206 spectral lines. For the other spectral lines, the unpredicatblility is set to 0.4.
  The unpredictability for the first 6 lines is calculated from the long FFT (window length = 1024, shiftlen = 576). For the spectral lines 6 upto 205 the unpredictability is calculated from the short FFT (window length 256, shiftlen = 192):

$$cw(w) = \begin{cases} cw\_l(w) & \text{for } 0 = w < 6 \\ cw\_s(w/4) & \text{for } 6 = w < 206,\ w=6,10,14,... \\ 0.4 & \text{for } w = 206 \end{cases}$$

  cw_l is the unpredictability calculated from the long FFT, cw_s is the unpredictability calculated from the second short block out of three short blocks within one granule.

- The spreading function has been replaced:

  If j = i tmpy= 3.0 (j -i)
  else tmpy= 1.5(j-i) is used.

  Only values of the spreading function greater than 10-6 are used. All other values are set to zero.

- For converting the unpredictability the parameters

  conv1 = -0.299
  conv2 = - 0.43

  are used.

- The parameter NMT (noise masking tone) is set to 6.0 db for all threshold calculation partions. The parameter TMN (tone masking noise) is set to 29.0 db for all partitions.
  For minval see table "threshold calculation partitions"

- The psychoacoustic entropy is estimated from the ratio thr/eb, where thr is the threshold and eb is the energy:

  pe = - ? (cbwidthk • log(thrk/(ebk+1.)))

  where k indexes the threshold calculation partitions and cbwidth is the width of the threshold calculation partition (see tables).

- pre-echo control

  The following constants are used for the control of pre-echo's (see block diagram):

  rpelev = 2.
  rpelev2 = 16.

- The threshold is not spread over the FFT lines. The threshold calculation partitions are converted directly to scalefactor bands. The first partition which is added to the scalefactor band is weighted with w1, the last with w2 (see table 3-Annex 3-C.8 "converting threshold calculation partitions to scalefactor bands"). The table contains also the number of partitions (cbw) converted to one scalefactor band (excluding the first and the last partition).
  The parameters bo and bu are shown in the table 3-Annex 3-C.8 used for converting threshold calculation partitions to scalefactor bands.

- For short blocks a simplified version of the threshold calculation (constant signal to noise ratio) is used. The constants can be found in the columns "SNR (dB)" in table 3-Annex 3-C.7. below.

Tables:

Table 3-C.7: Threshold calculation partitions with following parameters: width, minval, threshold in quiet, norm and bval:

Table 3-C.7a: Sampling_frequency = 48 kHz
long blocks

no.	FFT-lines		minval	qthr	norm	bval
0	1	24.5	4.532	0.970	0.000	
1	1	24.5	4.532	0.755	0.469	
2	1	24.5	4.532	0.738	0.937	
3	1	24.5	0.904	0.730	1.406	
4	1	24.5	0.904	0.724	1.875	
5	1	20	0.090	0.723	2.344	
6	1	20	0.090	0.723	2.812	
7	1	20	0.029	0.723	3.281	
8	1	20	0.029	0.718	3.750	
9	1	20	0.009	0.690	4.199	
10	1	20	0.009	0.660	4.625	
11	1	18	0.009	0.641	5.047	
12	1	18	0.009	0.600	5.437	
13	1	18	0.009	0.584	5.828	
14	1	12	0.009	0.531	6.187	
15	1	12	0.009	0.537	6.522	
16	2	6	0.018	0.857	7.174	
17	2	6	0.018	0.858	7.800	
18	2	3	0.018	0.853	8.402	
19	2	3	0.018	0.824	8.966	
20	2	3	0.018	0.778	9.483	
21	2	3	0.018	0.740	9.966	
22	2	0	0.018	0.709	10.426	
23	2	0	0.018	0.676	10.866	
24	2	0	0.018	0.632	11.279	
25	2	0	0.018	0.592	11.669	
26	2	0	0.018	0.553	12.042	
27	2	0	0.018	0.510	12.386	
28	2	0	0.018	0.513	12.721	
29	3	0	0.027	0.608	13.115	
30	3	0	0.027	0.673	13.561	
31	3	0	0.027	0.636	13.983	
32	3	0	0.027	0.586	14.371	
33	3	0	0.027	0.571	14.741	
34	4	0	0.036	0.616	15.140	
35	4	0	0.036	0.640	15.562	
36	4	0	0.036	0.597	15.962	

37	4	0	0.036	0.538	16.324
38	4	0	0.036	0.512	16.665
39	5	0	0.045	0.528	17.020
40	5	0	0.045	0.516	17.373
41	5	0	0.045	0.493	17.708
42	6	0	0.054	0.499	18.045
43	7	0	0.063	0.525	18.398
44	7	0	0.063	0.541	18.762
45	8	0	0.072	0.528	19.120
46	8	0	0.072	0.510	19.466
47	8	0	0.072	0.506	19.807
48	10	0	0.180	0.525	20.159
49	10	0	0.180	0.536	20.522
50	10	0	0.180	0.518	20.873
51	13	0	0.372	0.501	21.214
52	13	0	0.372	0.496	21.553
53	14	0	0.400	0.497	21.892
54	18	0	1.628	0.495	22.231
55	18	0	1.628	0.494	22.569
56	20	0	1.808	0.497	22.909
57	25	0	22.607	0.494	23.248
58	25	0	22.607	0.487	23.583
59	35	0	31.650	0.483	23.915
60	67	0	605.867	0.482	24.246
61	67	0	605.867	0.524	24.576

Table 3.-C.7b: Sampling_frequency = 44.1 kHz
long blocks

no.	FFT-lines		minval	qthr	norm	bval
0	1	24.5	4.532	0.951	0.000	
1	1	24.5	4.532	0.700	0.431	
2	1	24.5	4.532	0.681	0.861	
3	1	24.5	0.904	0.675	1.292	
4	1	24.5	0.904	0.667	1.723	
5	1	20	0.090	0.665	2.153	
6	1	20	0.090	0.664	2.584	
7	1	20	0.029	0.664	3.015	
8	1	20	0.029	0.664	3.445	
9	1	20	0.029	0.655	3.876	
10	1	20	0.009	0.616	4.279	
11	1	20	0.009	0.597	4.670	
12	1	18	0.009	0.578	5.057	
13	1	18	0.009	0.541	5.415	
14	1	18	0.009	0.575	5.774	
15	2	12	0.018	0.856	6.422	
16	2	6	0.018	0.846	7.026	
17	2	6	0.018	0.840	7.609	
18	2	3	0.018	0.822	8.168	
19	2	3	0.018	0.800	8.710	
20	2	3	0.018	0.753	9.207	
21	2	3	0.018	0.704	9.662	
22	2	0	0.018	0.674	10.099	
23	2	0	0.018	0.640	10.515	
24	2	0	0.018	0.609	10.917	
25	2	0	0.018	0.566	11.293	
26	2	0	0.018	0.535	11.652	
27	2	0	0.018	0.531	11.997	
28	3	0	0.027	0.615	12.394	

29	3	0	0.027	0.686	12.850
30	3	0	0.027	0.650	13.277
31	3	0	0.027	0.611	13.681
32	3	0	0.027	0.567	14.062
33	3	0	0.027	0.520	14.411
34	3	0	0.027	0.513	14.751
35	4	0	0.036	0.557	15.119
36	4	0	0.036	0.584	15.508
37	4	0	0.036	0.570	15.883
38	5	0	0.045	0.579	16.263
39	5	0	0.045	0.585	16.654
40	5	0	0.045	0.548	17.020
41	6	0	0.054	0.536	17.374
42	6	0	0.054	0.550	17.744
43	7	0	0.063	0.532	18.104
44	7	0	0.063	0.504	18.447
45	7	0	0.063	0.496	18.781
46	9	0	0.081	0.516	19.130
47	9	0	0.081	0.527	19.487
48	9	0	0.081	0.516	19.838
49	10	0	0.180	0.497	20.179
50	10	0	0.180	0.489	20.510
51	11	0	0.198	0.502	20.852
52	14	0	0.400	0.502	21.196
53	14	0	0.400	0.491	21.531
54	15	0	0.429	0.497	21.870
55	20	0	1.808	0.504	22.214
56	20	0	1.808	0.504	22.558
57	21	0	1.899	0.495	22.898
58	27	0	24.415	0.486	23.232
59	27	0	24.415	0.484	23.564
60	36	0	32.554	0.483	23.897
61	73	0	660.124	0.475	24.229
62	18	0	162.770	0.515	24.542

Table 3-C.7c: Sampling_frequency = 32 kHz
long blocks

no.	FFT-lines		minval	qthr	norm	bval
0	2	24.5	9.064	0.997	0.312	
1	2	24.5	9.064	0.893	0.937	
2	2	24.5	1.808	0.881	1.562	
3	2	20	0.181	0.873	2.187	
4	2	20	0.181	0.872	2.812	
5	2	20	0.057	0.871	3.437	
6	2	20	0.018	0.860	4.045	
7	2	20	0.018	0.839	4.625	
8	2	18	0.018	0.812	5.173	
9	2	18	0.018	0.784	5.698	
10	2	12	0.018	0.741	6.184	
11	2	12	0.018	0.697	6.634	
12	2	6	0.018	0.674	7.070	
13	2	6	0.018	0.651	7.492	
14	2	6	0.018	0.633	7.905	
15	2	3	0.018	0.611	8.305	
16	2	3	0.018	0.589	8.695	
17	2	3	0.018	0.575	9.064	

18	3	3	0.027	0.654	9.483
19	3	3	0.027	0.724	9.966
20	3	0	0.027	0.701	10.425
21	3	0	0.027	0.673	10.866
22	3	0	0.027	0.631	11.279
23	3	0	0.027	0.592	11.669
24	3	0	0.027	0.553	12.042
25	3	0	0.027	0.510	12.386
26	3	0	0.027	0.505	12.721
27	4	0	0.036	0.562	13.091
28	4	0	0.036	0.598	13.488
29	4	0	0.036	0.589	13.873
30	5	0	0.045	0.607	14.268
31	5	0	0.045	0.620	14.679
32	5	0	0.045	0.580	15.067
33	5	0	0.045	0.532	15.424
34	5	0	0.045	0.517	15.771
35	6	0	0.054	0.517	16.120
36	6	0	0.054	0.509	16.466
37	6	0	0.054	0.506	16.807
38	8	0	0.072	0.522	17.158
39	8	0	0.072	0.531	17.518
40	8	0	0.072	0.519	17.869
41	10	0	0.090	0.512	18.215
42	10	0	0.090	0.509	18.562
43	10	0	0.090	0.497	18.902
44	12	0	0.108	0.494	19.239
45	12	0	0.108	0.501	19.579
46	13	0	0.117	0.507	19.925
47	14	0	0.252	0.502	20.269
48	14	0	0.252	0.493	20.606
49	16	0	0.289	0.497	20.944
50	20	0	0.572	0.506	21.288
51	20	0	0.572	0.510	21.635
52	23	0	0.658	0.504	21.979
53	27	0	2.441	0.496	22.319
54	27	0	2.441	0.493	22.656
55	32	0	2.894	0.490	22.993
56	37	0	33.458	0.483	23.326
57	37	0	33.458	0.458	23.656
58	12	0	10.851	0.500	23.937

Table 3-C.7d: Sampling_frequency = 48 kHz
short blocks

no.	FFT-lines		qthr	norm	SNR (db)	bval
0	1	4.532	0.970	-8.240	0.000	
1	1	0.904	0.755	-8.240	1.875	
2	1	0.029	0.738	-8.240	3.750	
3	1	0.009	0.730	-8.240	5.437	
4	1	0.009	0.724	-8.240	6.857	
5	1	0.009	0.723	-8.240	8.109	
6	1	0.009	0.723	-8.240	9.237	
7	1	0.009	0.723	-8.240	10.202	
8	1	0.009	0.718	-8.240	11.083	
9	1	0.009	0.690	-8.240	11.864	

10	1	0.009	0.660	-7.447	12.553
11	1	0.009	0.641	-7.447	13.195
12	1	0.009	0.600	-7.447	13.781
13	1	0.009	0.584	-7.447	14.309
14	1	0.009	0.532	-7.447	14.803
15	1	0.009	0.537	-7.447	15.250
16	1	0.009	0.857	-7.447	15.667
17	1	0.009	0.858	-7.447	16.068
18	1	0.009	0.853	-7.447	16.409
19	2	0.018	0.824	-7.447	17.044
20	2	0.018	0.778	-6.990	17.607
21	2	0.018	0.740	-6.990	18.097
22	2	0.018	0.709	-6.990	18.528
23	2	0.018	0.676	-6.990	18.930
24	2	0.018	0.632	-6.990	19.295
25	2	0.018	0.592	-6.990	19.636
26	3	0.054	0.553	-6.990	20.038
27	3	0.054	0.510	-6.990	20.486
28	3	0.054	0.513	-6.990	20.900
29	4	0.114	0.608	-6.990	21.305
30	4	0.114	0.673	-6.020	21.722
31	5	0.452	0.637	-6.020	22.128
32	5	0.452	0.586	-6.020	22.512
33	5	0.452	0.571	-6.020	22.877
34	7	6.330	0.616	-5.229	23.241
35	7	6.330	0.640	-5.229	23.616
36	11	9.947	0.597	-5.229	23.974
37	17	153.727	0.538	-5.229	24.312

**Table 3-C.7e: Sampling_frequency = 44.1 kHz
short blocks**

no.	FFT-lines	qthr	norm	SNR (db)	bval
0	1	4.532	0.952	-8.240	0.000
1	1	0.904	0.700	-8.240	1.723
2	1	0.029	0.681	-8.240	3.445
3	1	0.009	0.675	-8.240	5.057
4	1	0.009	0.667	-8.240	6.422
5	1	0.009	0.665	-8.240	7.609
6	1	0.009	0.664	-8.240	8.710
7	1	0.009	0.664	-8.240	9.662
8	1	0.009	0.664	-8.240	10.515
9	1	0.009	0.655	-8.240	11.293
10	1	0.009	0.616	-7.447	12.009
11	1	0.009	0.597	-7.447	12.625
12	1	0.009	0.578	-7.447	13.210
13	1	0.009	0.541	-7.447	13.748
14	1	0.009	0.575	-7.447	14.241
15	1	0.009	0.856	-7.447	14.695
16	1	0.009	0.846	-7.447	15.125
17	1	0.009	0.840	-7.447	15.508
18	1	0.009	0.822	-7.447	15.891
19	2	0.018	0.800	-7.447	16.537
20	2	0.018	0.753	-6.990	17.112
21	2	0.018	0.704	-6.990	17.620
22	2	0.018	0.674	-6.990	18.073
23	2	0.018	0.640	-6.990	18.470

24	2	0.018	0.609	-6.990	18.849
25	3	0.027	0.566	-6.990	19.271
26	3	0.027	0.535	-6.990	19.741
27	3	0.054	0.531	-6.990	20.177
28	3	0.054	0.615	-6.990	20.576
29	3	0.054	0.686	-6.990	20.950
30	4	0.114	0.650	-6.020	21.316
31	4	0.114	0.612	-6.020	21.699
32	5	0.452	0.567	-6.020	22.078
33	5	0.452	0.520	-6.020	22.438
34	5	0.452	0.513	-5.229	22.782
35	7	6.330	0.557	-5.229	23.133
36	7	6.330	0.584	-5.229	23.484
37	7	6.330	0.570	-5.229	23.828
38	19	171.813	0.578	-4.559	24.173

Table 3-C.7f: Sampling_frequency = 32 kHz
short blocks

no.	FFT-lines		qthr	norm	SNR (db)	bval
0	1	4.532	0.997	-8.240	0.000	
1	1	0.904	0.893	-8.240	1.250	
2	1	0.090	0.881	-8.240	2.500	
3	1	0.029	0.873	-8.240	3.750	
4	1	0.009	0.872	-8.240	4.909	
5	1	0.009	0.871	-8.240	5.958	
6	1	0.009	0.860	-8.240	6.857	
7	1	0.009	0.839	-8.240	7.700	
8	1	0.009	0.812	-8.240	8.500	
9	1	0.009	0.784	-8.240	9.237	
10	1	0.009	0.741	-7.447	9.895	
11	1	0.009	0.697	-7.447	10.500	
12	1	0.009	0.674	-7.447	11.083	
13	1	0.009	0.651	-7.447	11.604	
14	1	0.009	0.633	-7.447	12.107	
15	1	0.009	0.611	-7.447	12.554	
16	1	0.009	0.589	-7.447	13.000	
17	1	0.009	0.575	-7.447	13.391	
18	1	0.009	0.654	-7.447	13.781	
19	2	0.018	0.724	-7.447	14.474	
20	2	0.018	0.701	-6.990	15.096	
21	2	0.018	0.673	-6.990	15.667	
22	2	0.018	0.631	-6.990	16.177	
23	2	0.018	0.592	-6.990	16.636	
24	2	0.018	0.553	-6.990	17.057	
25	2	0.018	0.510	-6.990	17.429	
26	2	0.018	0.506	-6.990	17.786	
27	3	0.027	0.562	-6.990	18.177	
28	3	0.027	0.598	-6.990	18.597	
29	3	0.027	0.589	-6.990	18.994	
30	3	0.027	0.607	-6.020	19.352	
31	3	0.027	0.620	-6.020	19.693	
32	4	0.072	0.580	-6.020	20.066	
33	4	0.072	0.532	-6.020	20.461	
34	4	0.072	0.517	-5.229	20.841	
35	5	0.143	0.517	-5.229	21.201	
36	5	0.143	0.509	-5.229	21.549	

37	6	0.172	0.506	-5.229	21.911
38	7	0.633	0.522	-4.559	22.275
39	7	0.633	0.531	-4.559	22.625
40	8	0.723	0.519	-3.980	22.971
41	10	9.043	0.512	-3.980	23.321

Table 3-C.8: Tables for converting threshold calculation partitions to scalefactor bands

Table 3-C.8a: Sampling_frequency = 48 kHz
long blocks

no. sb	**cbw**	**bu**	**bo**	**w1**	**w2**
0	3	0	4	1.000	0.056
1	3	4	7	0.944	0.611
2	4	7	11	0.389	0.167
3	3	11	14	0.833	0.722
4	3	14	17	0.278	0.639
5	2	17	19	0.361	0.417
6	3	19	22	0.583	0.083
7	2	22	24	0.917	0.750
8	3	24	27	0.250	0.417
9	3	27	30	0.583	0.648
10	3	30	33	0.352	0.611
11	3	33	36	0.389	0.625
12	4	36	40	0.375	0.144
13	3	40	43	0.856	0.389
14	3	43	46	0.611	0.160
15	3	46	49	0.840	0.217
16	3	49	52	0.783	0.184
17	2	52	54	0.816	0.886
18	3	54	57	0.114	0.313
19	2	57	59	0.687	0.452
20	1	59	60	0.548	0.908

Table 3-C.8b: Sampling_frequency = 44.1 kHz
long blocks

no. sb	**cbw**	**bu**	**bo**	**w1**	**w2**
0	3	0	4	1.000	0.056
1	3	4	7	0.944	0.611
2	4	7	11	0.389	0.167
3	3	11	14	0.833	0.722
4	3	14	17	0.278	0.139
5	1	17	18	0.861	0.917
6	3	18	21	0.083	0.583
7	3	21	24	0.417	0.250
8	3	24	27	0.750	0.805
9	3	27	30	0.194	0.574
10	3	30	33	0.426	0.537
11	3	33	36	0.463	0.819
12	4	36	40	0.180	0.100
13	3	40	43	0.900	0.468
14	3	43	46	0.532	0.623
15	3	46	49	0.376	0.450
16	3	49	52	0.550	0.552
17	3	52	55	0.448	0.403
18	2	55	57	0.597	0.643

19	2	57	59	0.357	0.722
20	2	59	61	0.278	0.960

Table 3-C.8c: Sampling_frequency = 32 kHz
long blocks

no. sb	**cbw**	**bu**	**bo**	**w1**	**w2**
0	1	0	2	1.000	0.528
1	2	2	4	0.472	0.305
2	2	4	6	0.694	0.083
3	1	6	7	0.917	0.861
4	2	7	9	0.139	0.639
5	2	9	11	0.361	0.417
6	3	11	14	0.583	0.083
7	2	14	16	0.917	0.750
8	3	16	19	0.250	0.870
9	3	19	22	0.130	0.833
10	4	22	26	0.167	0.389
11	4	26	30	0.611	0.478
12	4	30	34	0.522	0.033
13	3	34	37	0.967	0.917
14	4	37	41	0.083	0.617
15	3	41	44	0.383	0.995
16	4	44	48	0.005	0.274
17	3	48	51	0.726	0.480
18	3	51	54	0.519	0.261
19	2	54	56	0.739	0.884
20	2	56	58	0.116	1.000

Table 3-C.8d: Sampling_frequency = 48 kHz
short blocks

no. sb	**cbw**	**bu**	**bo**	**w1**	**w2**
0	2	0	3	1.000	0.167
1	2	3	5	0.833	0.833
2	3	5	8	0.167	0.500
3	3	8	11	0.500	0.167
4	4	11	15	0.833	0.167
5	4	15	19	0.833	0.583
6	3	19	22	0.417	0.917
7	4	22	26	0.083	0.944
8	4	26	30	0.055	0.042
9	2	30	32	0.958	0.567
10	3	32	35	0.433	0.167
11	2	35	37	0.833	0.618

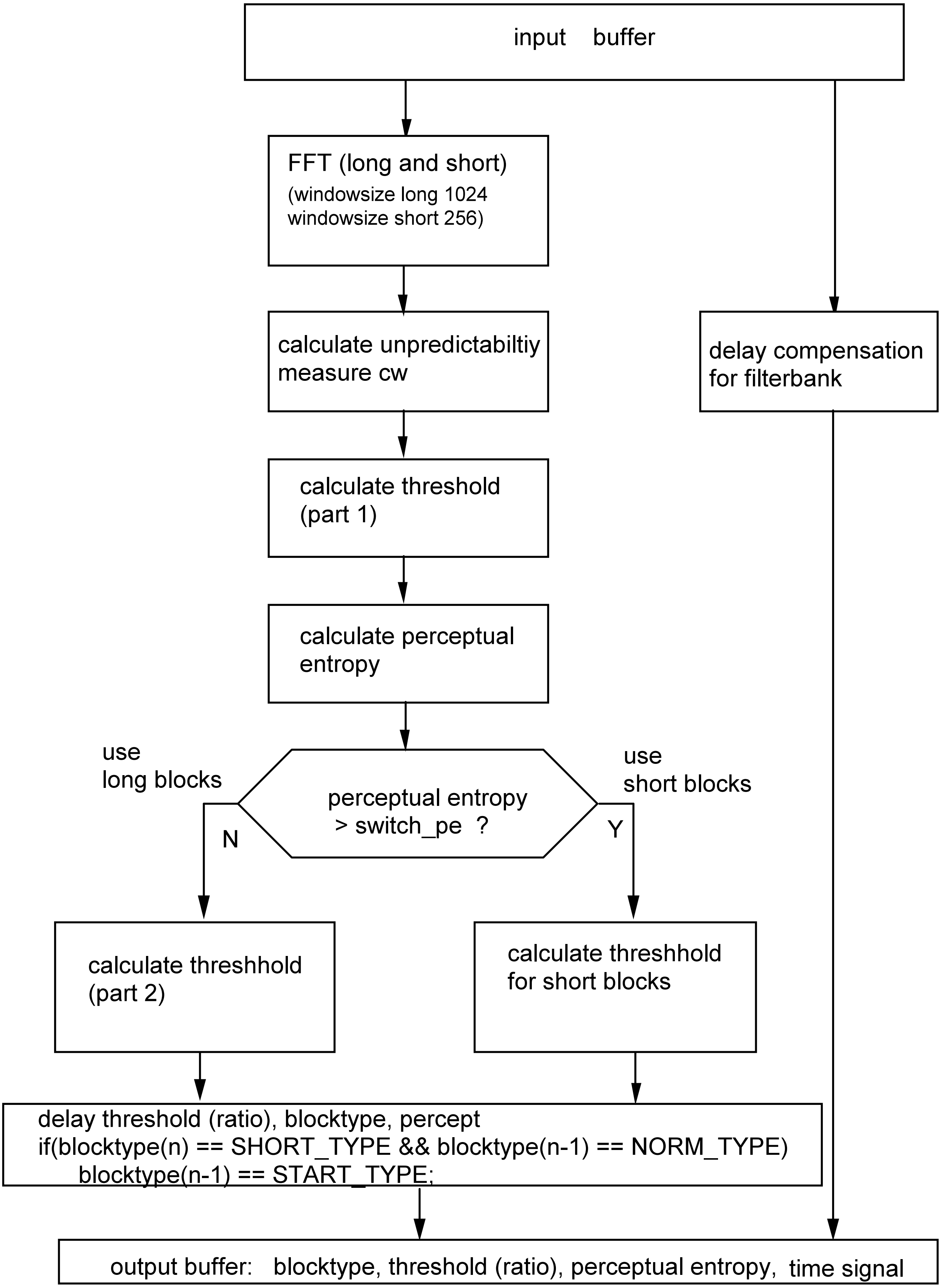
Table 3-C.8e: Sampling_frequency = 44.1 kHz
short blocks

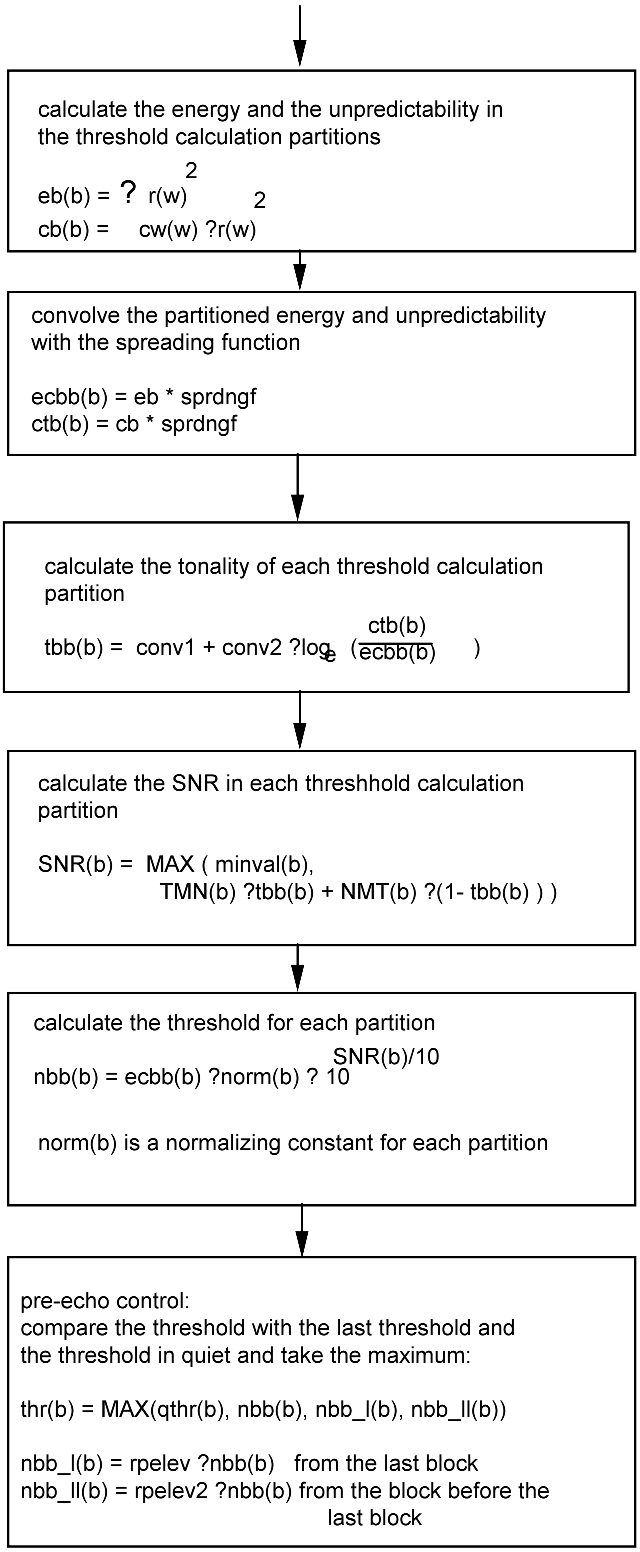
no. sb	**cbw**	**bu**	**bo**	**w1**	**w2**
0	2	0	3	1.000	0.167
1	2	3	5	0.833	0.833
2	3	5	8	0.167	0.500
3	3	8	11	0.500	0.167
4	4	11	15	0.833	0.167
5	5	15	20	0.833	0.250
6	3	20	23	0.750	0.583
7	4	23	27	0.417	0.055

8	3	27	30	0.944	0.375
9	3	30	33	0.625	0.300
10	3	33	36	0.700	0.167
11	2	36	38	0.833	1.000

Table 3-C.8f: Sampling_frequency = 32 kHz
short blocks

no. sb	cbw	bu	bo	w1	w2
0	2	0	3	1.000	0.167
1	2	3	5	0.833	0.833
2	3	5	8	0.167	0.500
3	3	8	11	0.500	0.167
4	4	11	15	0.833	0.167
5	5	15	20	0.833	0.250
6	4	20	24	0.750	0.250
7	5	24	29	0.750	0.055
8	4	29	33	0.944	0.375
9	4	33	37	0.625	0.472
10	3	37	40	0.528	0.937
11	1	40	41	0.062	1.000


input buffer
FFT (long and short)
(windowsize long 1024
windowsize short 256)
calculate unpredictabiltiy
measure cw
delay compensation
for filterbank
calculate threshold
(part 1)
calculate perceptual
entropy
use
long blocks
perceptual entropy
> switch_pe ?
use
short blocks
N
Y
calculate threshhold
(part 2)
calculate threshhold
for short blocks
delay threshold (ratio), blocktype, percept
if(blocktype(n) == SHORT_TYPE && blocktype(n-1) == NORM_TYPE)
blocktype(n-1) == START_TYPE;
output buffer: blocktype, threshold (ratio), perceptual entropy, time signal



Block diagram psychoacoustic model II, layer III: calculate threshold (part 1) convert from threshold calculation partitions to codec partitions (scalefactor bands):

energy in each scalefactor band:

$$en(sb) = w1\ ?eb(bu) + \sum_{b = bu+1}^{b = bo-1} eb(b) + w2\ ?eb(bo)$$

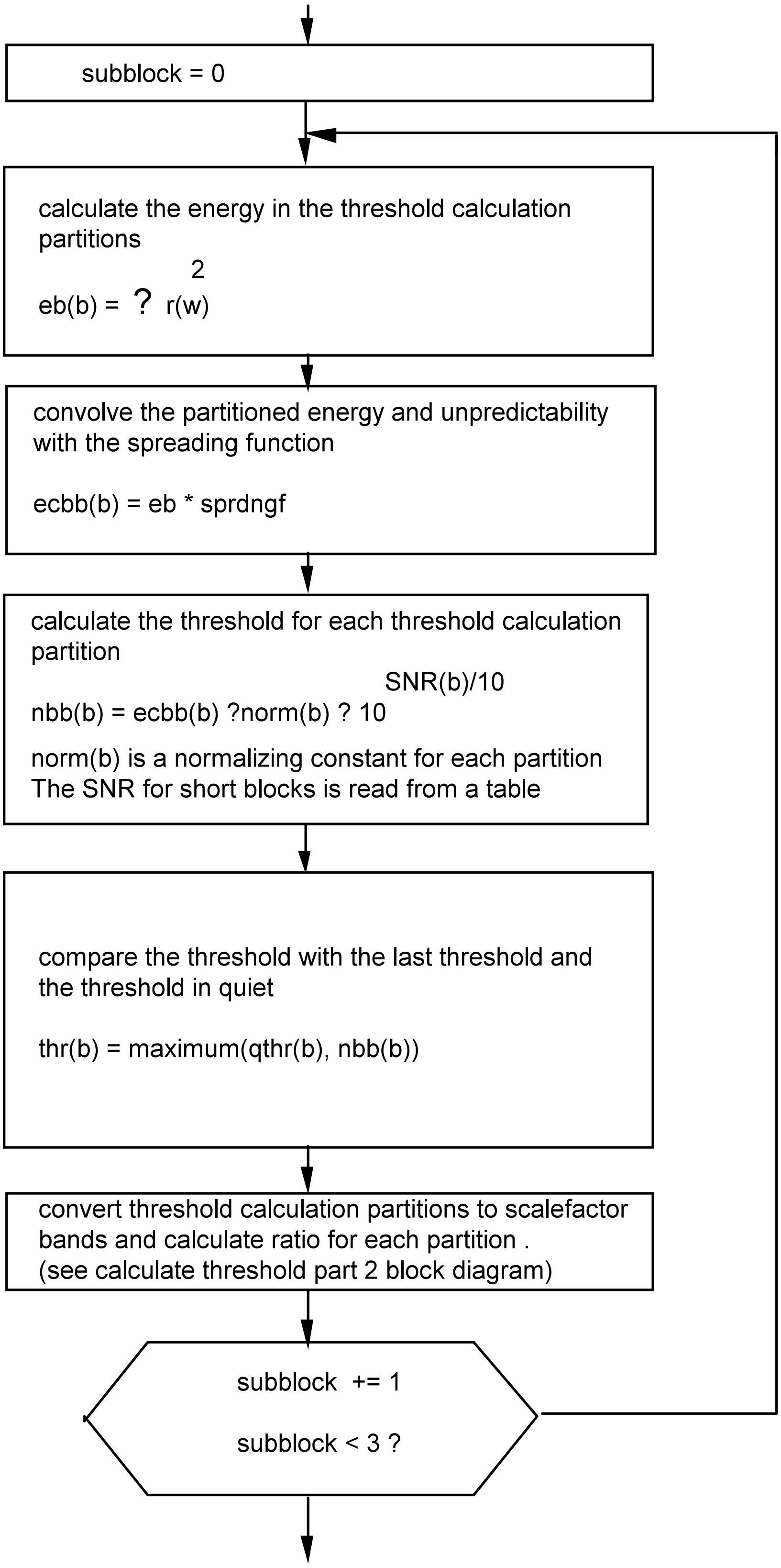
threshold in each scalefactor band:

$$thrn(sb) = w1\ ?thr(bu) + \sum_{b = bu+1}^{b = bo-1} thr(b) + w2\ ?thr(bo)$$

Calculate ratios for each scalefactor band:

$$ratio(sb) = \frac{thrn(sb)}{en(sb)}$$

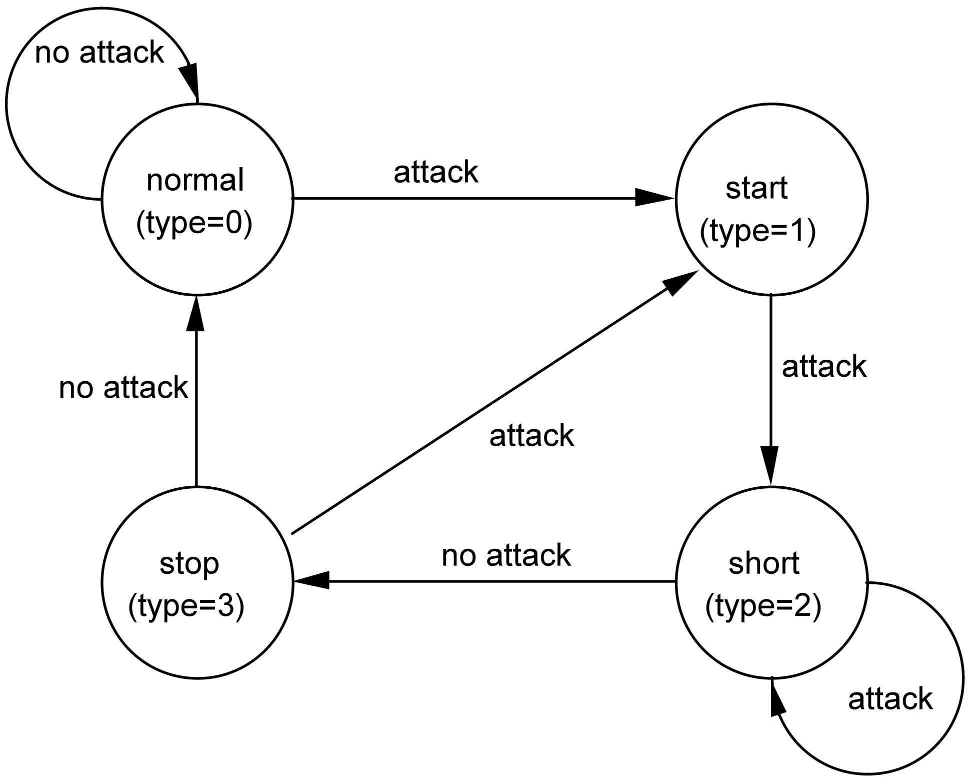
Block diagram psychoacoustic model II, layer III: calculate threshold (part 2)



Block diagram psychoacoustic model II, layer III: calculate threshold for short blocks

Window switching decision:

The decision whether the filterbank should be switched to short windows is derived from the calculation of the masking threshold by calculating the estimate of the psychoacoustic entropy (PE) and switching when the PE exceeds the value 1800. If this condition is met, the sequence start (block_type=1), short (block_type=2), short, stop (block_type=3) is started. The figure below shows the possible state changes for the window switching logic.



3. Analysis Part of the Hybrid Filterbank

The subband analysis of the polyphase filterbank is described in clause 3-C.1.3, "SUBBAND ANALYSIS FILTER". The output of the polyphase filterbank is the input to the subdivision using the MDCT. According to the output of the psychoacoustic model (variables **blocksplit_flag** and **block_type**) the window and transform types **normal**, **start**, **short** or **stop** are used.

18 consecutive output values of one granule and 18 output values of the granule before are assembled to one block of 36 samples.

Block type "**normal**"

$$z_i = x'_i \sin\left(\frac{\text{š}}{36}\left(i + \frac{1}{2}\right)\right) \qquad \text{for i=0 to 35}$$

Block type "**start**"

$$z_i = \begin{cases} x'_i \sin\left(\frac{\pi}{36}\left(i + \frac{1}{2}\right)\right) & \text{for i=0 to 17} \\ x'_i & \text{for i=18 to23} \\ x'_i \sin\left(\frac{\pi}{12}\left(i - 18 + \frac{1}{2}\right)\right) & \text{for i=24 to 29} \\ 0 & \text{for i=30 to 35} \end{cases}$$

Block type "**stop**"

$$z_i = \begin{cases} 0 & \text{for i=0 to 5} \\ x'_i \sin\left(\frac{\pi}{12}\left(i - 6 + \frac{1}{2}\right)\right) & \text{for i=6 to 11} \\ x'_i & \text{for i=12 to 17} \\ x'_i \sin\left(\frac{\pi}{36}\left(i + \frac{1}{2}\right)\right) & \text{for i=18 to 35} \end{cases}$$

Block type "**short"**

The block of 36 samples is divided into three overlapping blocks:

$$y_i^{(0)} = x'_{i+6} \quad \text{for i=0 to 11}$$

$$y_i^{(1)} = x'_{i+12} \quad \text{for i=0 to 11}$$

$$y_i^{(2)} = x'_{i+18} \quad \text{for i=0 to 11}$$

Each of the three small blocks is windowed separately:

$$z_i^{(k)} = y_i^{(k)} \sin\left(\frac{\pi}{12}\left(i + \frac{1}{2}\right)\right) \quad \text{for i=0 to 11, for k=0 to 2}$$

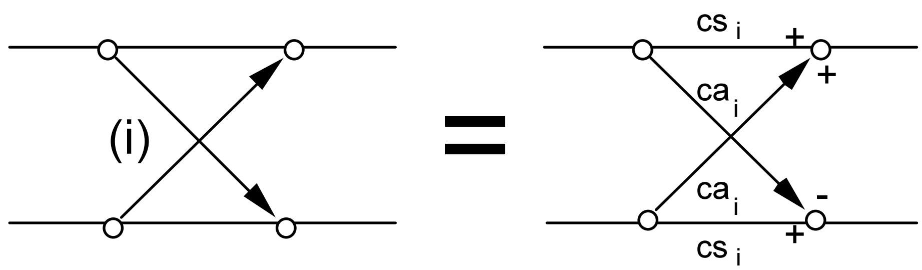
MDCT:

In the following n is the number of windowed samples. For short blocks n is 12, for long blocks n is 36. The analytical expression of the MDCT is:

$$x_i = \sum_{k=0}^{n-1} z_k \cos\left(\frac{\pi}{2n}\left(2k+1+\frac{n}{2}\right)(2i+1)\right) \qquad \text{for i=0 to } \frac{n}{2} - 1$$

Aliasing-Butterfly, Encoder:

The calculation of aliasing reduction in the encoder is performed as in the decoder. The general procedure is shown in Fig. 3-Annex 3-A.5. The butterfly definition to be used in the encoder is shown below. The coefficients cai and csi can be found in 3-Annex Table 3-B.9



4. Calculation of average available bits

The average number of bits per granule is calculated from the frame size. The bitrate 64 kb/second is used as an example. At bitrate 64 kb/second at 48000 samples per second,

(64000 * 0.24 bits per frame) / (2 granules per frame) = 768 bits per granule

As the header takes 32 bits and the side information takes 17 bytes (136 bits) in single_channel mode, the average amount of available bits for the main_data for a granule is given by

mean_bits = 768 bits per granule - (32+136 bits per frame)/(2 granules per frame) = 684 bits per granule

Bit reservoir:

The bit reservoir can provide additional bits which may be used for the granule. The number of bits which are provided is determined within the iteration loops.

5. Quantization and Encoding of Frequency Domain Samples

The frequency domain data are quantized and coded within two nested iteration loops. Chapter 3-C1.5.4 contains a detailed description of these iteration loops.

6. Formatting

The details about the Layer III bitstream format can be found in the clause 2.4.4 of the main part of this audio standard, "SPECIFICATION OF THE CODED AUDIO BITSTREAM SYNTAX " . The formatting of the Huffman code words is described below:

The Huffman codewords are in sequence from low to high frequencies. In the iteration loops the following variables have been calculated and are used in encoding the Huffman codewords:

is(i), i=0...575	quantized frequency domain values
table_select[region]	Huffman code table used for regions (region = 0, 1, 2)
region_adress1	defines the border between region 0 and 1
region_adress2	defines the border between region 1 and 2
max_value[region]	maximum absolute value of quantized data in regions (region = 0, 1, 2)

The data are written to the bitstream according to the Huffman code syntax described in clause 2.4.2.7

The actual assembly of the Huffman code for the big_values part is described in a pseudo high level language:

```
for region number from 0 to 2
  if table_select for this region is 0
    nothing to do, all values in region are zero
```

```
 else
  if table_select for this region is > 15
   an ESC-table is used: look up linbits value connected to the table used
   for i = begin of region to end of region, count in pairs
    x = is(i), y = is(i+1)
    if x > 14
     linbitsx = x - 15, x = 15
    end if
    signx = sign(x), x = abs(x)
    if y > 14
     linbitsy = y - 15, y = 15
    end if
    signy = sign(y), y = abs(y)
    look for codeword = hcod([x][y]]) in table table_seletct
    write hcod([x][y]), beginning with the leftmost bit, number of bits is hlen([x][y])
    if x > 14
     write linbitsx to the bitstream, number of bits is linbits
    end if
    if x != 0
     write signx to bitstream
    end if
    if y > 14
     write linbitsy to the bitstream, number of bits is linbits
    end if
    if y != 0
     write signy to bitstream
    end if
   end do
  else
   no ESC-words are used in this region:
   for i = beginning of region to end of region, count in pairs
    x = is(i), y = is(i+1)
   signx = sign(x), x = abs(x)
   signy = sign(y), y = abs(y)
    look for codeword = hcod([x][y]) in table table_seletct
    write hcod([x][y]), beginning with the leftmost bit, number of bits is hlen([x][y])
   if x != 0
     write signx to bitstream
    end if
    if y != 0
     write signy to bitstream
    end if
   end do
  end if
 end if
end for
```

A possible application for the private_bits is to use them as frame counter.

3-C.1.5.4 Layer III Iteration Loops

1. Introduction

The description of the Layer III loop module is subdivided into three levels. The top level is called "loops frame program". The loops frame program calls a subroutine named "outer iteraton loop" which calls the subroutine "inner iteration loop". For each level a corresponding flow diagram is shown.

The loops module quantizes an input vector of spectral data in an iterative process according to several demands. The inner loop quantizes the input vector and increases the quantizer step size until the output vector can be coded with the available amount of bit. After completion of the inner loop an outer loop checks the distortion of each scalefactor band and, if the allowed distortion is exceeded, amplifies the scalefactor band and calls the inner loop again.

Layer III loops module input:
(1) vector of the magnitudes of the spectral values xr(0..575)
(2) xmin(cb), the allowed distortion of the scalefactor bands
(3) blocksplit_flag which in conjunction with switch_point determines the number of scalefactor bands
(4) mean_bits (bit available for the Huffman coding and the coding of the scalefactors)
(5) more_bits, the number of bits in addition to the average number of bits, as demanded by the value of the psychoacoustic entropy for the granule:
more_bits = 3.1 * PE - (average number of bits)

Layer III loops module output:
(1) vector of quantized values ix(0..575)
(2) ifq(cb), the scalefactors
(3) qquant (quantizer step size information)
(4) number of unused bit available for later use
(5) preflag (loops preemphasis on/off)
(6) Huffman code related side information
- big_values (number of pairs of Huffman coded values, excluding "count1")
- count1table_select (Huffman code table of absolut values <= 1 at the upper end of the spectrum
- table_select[0..2](Huffman code table of regions)
- region_address1,2 (used to calculate boundaries between regions)
- part2_3_length

2. Preparatory Steps

2.1 Reset of all iteration variables

The scalefactors of the coder partitions scalefac[cb] are set to zero.

The counter qquant for the quantizer step size is reset to zero.

Preflag is reset to zero.

Scalefac_scale is reset to zero.

The inital value of quantanf is set as follows:

quantanf = system_const * loge(sfm),

where sfm is the spectral flatness measure and quantanf depends on the computational implementation of the encoder.

The spectral flatness measure sfm is given by

$$\mathrm{sfm} = \frac{e^{\frac{1}{n}\left(\sum_{i=0}^{n-1} \log xr(i)^2\right)}}{\frac{1}{n}\sum_{i=0}^{n-1} xr(i)^2}$$

The value of system_const is chosen so that for all signals the first iteration of the inner loop for all signals comes out with a bit sum higher than the desired bitsum. By that it is ensured that the first call of the inner loop results in the solution which uses as many of the available bits as possible. In order to spare computing time it is desirable to minimize the number of iterations by adapting the value of quantanf to the bitrate and the signal statistics.

2.2 Bit reservoir control

Bits are saved to the reservoir when fewer than the mean_bits are used to code one granule. If bits are saved for a frame, the value of main_data_end is increased accordingly. See diagram 3-Annex 3-A.7.1.

The number of bits which are made available for the main_data (called "max_bits") is derived from the actual estimated threshold (the PE as calculated by the psychoacoustic model), the average number of bits (mean_bits) and the actual content of the bit reservoir. The number of bytes in the bit reservoir is given by main_data_end.

The actual rules for the control of the bit reservoir are given below:

- If a number of bytes available to the inner iteration loop is not used for the Huffman encoding or other main_data, the number is added to the bit reservoir.
- If the bit reservoir contains more than 0.8 times the maximum allowed content of the bit reservoir, all bytes exceeding this number are made available for main_data (in addition to mean_bits)
- If more_bits is greater than 100 bits, then max(more_bits/8, 0.6*main_data_end) bytes are taken from the bit reservoir and made available for main_data (in addition to mean_bits).
- After the actual loops computations have been completed, the number of bytes not used for main_data is added to the bit reservoir.
- If after the step above the number of bytes in the bit reservoir exceeds the maximum allowed content, stuffing bits are written to the bitstream and the content of the bit reservoir is adjusted accordingly.

2.3 Calculation of the scalefactor select information (scfsi)

The scfsi contains the information, which scalefactors (grouped in the scfsi_bands) of the first granule can also be used for the second granule. These scalefactors are therefore not transmitted, the gained bits can be used for the huffman coding.
To determine the usage of the scfsi, the following information of each granule must be stored:

1. The block type
2. The total energy of the granule:

$$en\_tot = int\left\{ \log2\left(\sum_{i=1}^{n} |xr(i)|^2 \right) \right\}$$

where n is the total number of spectral values

3. The energy of each scalefactor band:

$$en(cb) = int\left\{ \log2\left(\sum_{i=lbl(cb)}^{lbl(cb)+bw(cb)-1} |xr(i)|^2 \right) \right\}$$

where lbl(cb) is the number of the first coefficient belonging to scalefactor band cb and bw(cb) is the number of coefficients within scalefactor band cb

4. The allowed distortion of each scalefactor band:

$$xm(cr.bd) = int\{\log2(xmin(i))\}$$

xmin(cb) is calculated by the psychoacoustic model.

The scalefactors of the first granule are always transmitted. When coding the second granule, the information of the two granules is compared. There are four criteria to determine if the scfsi can be used in general. If one of the four is not fulfilled, the scfsi is disabled (that means it is set to 0 in all scfsi_bands). The criteria are (index *0* means first, index *1* second granule):

1. The spectral values are not all zero
2. None of the granules contains short blocks

3.

$$\left| en\_tot_0 - en\_tot_1 \right| < en\_tot_{krit}$$

4.

$$\sum_{\text{all scalefactor bands}} \left| en(cb)_0 - en(cb)_1 \right| < en\_dif_{krit}$$

If the scfsi is not disabled after the tests above, there are two criterias for each scfsi_band, which have both to be fulfilled to enable scfsi (that means to set it to 1 in this scfsi_band):

1.

$$\sum_{\text{all cr. bds in scfsi\_band}} \left| en(cr.bd)_0 - en(cr.bd)_1 \right| < en(scfsi\_band)_{krit}$$

2.

$$\sum_{\text{all cr. bds in scfsi\_band}} \left| xm(cr.bd)_0 - xm(cr.bd)_1 \right| < xm(scfsi\_band)_{krit}$$

The constants (with the index *krit*) have to be chosen so, that the scfsi is only enabled in case of similar energy/distortion.
Suggested values are:

en_tot	=	10	
en_dif	=	100	
en(scfsi_band)	=	10	for each scfsi_band
xm(scfsi_band)	=	10	for each scfsi_band

3. Outer Iteration Loop (distortion control loop)

The outer iteration loop controls the quantization noise which is produced by the quantization of the frequency domain lines within the inner iteration loop. The colouration of the noise is done by multiplikation of the lines within scalefactor bands with the actual scalefactors before doing the quantization. The following pseudo-code illustrates the multiplication.

```
do for each scalefactor band:
  do from lower index to upper index of scale factor band
    xr(i) = xr(i) * sqrt(2) ^ ((1 + scalefac_scale) * ifq(scalefactor band))
  end do
end do
```

In the actual system the multiplication is done incrementally with just the increase of the scalefactors applied in each distortion control loop. This is described in clause 3.5 below.

The distortion loop is always starting with scalefac_scale = 0. If after some iterations the maximum length of the scalefactors would be exceeded (see scalefac_compress table in 2.4.2.7 and 3.5 below), then scalefac_scale is increased to the value 1 thus increasing the possible dynamic range of the scalefactors. In this case the actual scalefactors and frequency lines have to be corrected accordingly.

3.1 Saving of the scalefactors

The scalefactors of all scalefactor bands ifq(cb) as well as the quantizer step size qquant are saved. If the computation of the outer loop is cancelled without having reached a proper result this values together with the quantized spectrum give an approximation and can be transmitted.

3.2 Call of inner iteration loop

For each outer iteration loop (distortion control loop) the inner iteration loop (rate control loop) is called. The parameters are the frequency domain values (hybrid filterbank output) with the scalefactors applied to the values within the scalefactor bands and the number of bits which are available to the rate control loop. The result is the number of bits actually used and the quantized frequency lines ix(i).

3.3 Calculation of the distortion of the scalefactor bands

For each scalefactor band the actual distortion is calculated according to:

$$xfsf(cr.bd.) = \sum_{lbl(cr.bd.)}^{i=lbl(cr.bd.)+bw(cr.bd.)-1} \frac{\left(|xr(i)| - ix(i)^{4/3} * \sqrt[4]{2}^{qquant+quantanf}\right)^2}{bandwidth(cr.bd.)}$$

where lbl(cb) is the number of the coefficient representing the lowest frequency in a scalefactor band and bw(cb) is the number of coefficients within this band.

3.4 Preemphasis

The preemphasis option (switched on by setting preflag to a value of 1) provides the possibility to amplifiy the upper part of the spectrum according to the preemphasis tables, B.6 in the annex.

```
if preflag==1
{
        xmin(j) = xmin(j) *ifqstep2*prefact(j)
        for (i=lower limit of scalefactor band j; i <=upper limit of scalefactor band j; i++)  {
                xr(i) = xr(i) * ifqstepprefact(j)
        }
}
```

The condition to switch on the preemphasis is up to the implementation. For example preemphasis could be switched on if in all of the upper 4 scalefactor bands the actual distortion exceeds the threshold after the first call of the inner loop.
If the second granule is being coded and scfsi is active in at least one scfsi_band, the preemphasis in the second granule is set equal to the setting in first granule.

3.5 Amplification of scalefactor bands which violate the masking threshold

All spectral values of the scalefactor bands which have a distortion that exceeds the allowed distortion are amplified by a factor of ifqstep. The value of ifqstep is transmitted by scalefac_scale.

```
if (xmin - xfsf) of scalefactor band j  <  0
{
        xmin(j) = xmin(j) * ifqstep2
        ifq(j) = ifq(j) + 1
        for (i=lower limit of scalefactor band; i <=upper limit of scalefactor band; i++)  {
                xr(i) = xr(i) * ifqstep
        }
}
```

If the second granule is being coded and scfsi is active in at least one scfsi_band, the following steps have to be done:

1. ifqstep has to be set similar to the first granule
2. If it is the first iteration, the scalefactors of scalefactor bands in which scfsi is enabled have to be taken over from the first granule. The corresponding spectral values have to be amplified:
   ```
   if ( scfsi according to scalefactor band j  =  1)
   {
   ifq(j) = ifq(j)first granule
   for (i=lower limit of scalefactor band; i <=upper limit of scalefactor band; i++)
           { xr(i) = xr(i) * ifqstepifq(j) }
   }
   ```
3. If it is not the first iteration, the amplification must be prevented for scalefactor bands in which scfsi is enabled.

3.5 Conditions for the termination of the loops processing

Normally the loops processing terminates if there is no scalefactor band with more than the allowed distortion. However this is not always possible to obtain. In this case there are other conditions to terminate the outer loop. If

a) all scalefactor bands are already amplified
b) the amplification of at least one band exceeds the upper limit which is determined by the transmission format of the scalefactors. The upper limit is a scalefactor of 15 for scalefactor bands 0 through 10 and 7 for scalefactors 11 through 20.

the loops processing stops and by restoring the saved ifq(cb.) a useful output is available. For realtime implementation there might be a third condition added which terminates the loops in case of a lack of computing time.

4. Inner Iteration Loop (rate control loop)

The inner iteration loop does the actual quantization of the frequency domain data and prepares the formatting. The table selection, subdivision of the big_values range into regions and the selection of the quantizer step size takes place here.

4.1 Quantization

The quantization of the complete vector of spectral values is done according to

$$ix(i) = \mathrm{nint}\left(\left(\frac{|xr(i)|}{\sqrt[4]{2}^{\,qquant+quantanf}}\right)^{0.75} - 0.0946\right)$$

4.2 Test of the maximum of the quantized values

The maximum allowed quantized value is limited. This limit is set to constraint the table size if a table-lookup is used to requantize the quantized frequency lines. The limit is given by the possible values of the length identifier, "linbits", of values flagged with an ESC-code. Therefore before any bit counting is done the quantizer stepsize is increased by

$$qquant = qquant+1$$

until the maximum of the quantized values is within the range of the largest Huffman code table.

4.3 Calculation of the run length of zeros

The run length rzero of pairs of spectral coefficients quantized to zero on the upper end of the spectrum is counted and called "rzero".

4.4 Calculation of the run length of values less or equal one

The run length of quadrupels of spectral coefficients quantized to one or zero, following the rzero pairs of zeros, is calculated and called "count1"

4.5 Counting the bit necessary to code the values less or equal one

One Huffman code word is used to code one of the "count1" quadrupels. There are two different Huffman code books with corresponding code length tables (table A and table B in 3-Annex 3-B.7). The number of bits to code all the count1 quadrupels is given by:

$$bitsum\_count1 = \min(bitsum\_table0 , bitsum\_table1)$$

where
count1table_0 is used to point to table A

$$bitsum\_table0 = \sum_{k=firstcount1}^{k=firstcount1+count1-1} count1table\_0\left(ix(4k)+2*ix(4k+1)+4*ix(4k+2)+8*ix(4k+3)\right)$$

and
count1table_1 is used to point to table B

$$bitsum\_table1 = \sum_{k=firstcount1}^{k=firstcount1+count1-1} count1table\_1\left(ix(4k)+2*ix(4k+1)+4*ix(4k+2)+8*ix(4k+3)\right)$$

The information which table is used is transmitted by count1table_select, which is "0" for table A or "1" for table B, respectively.

4.6 Call of subroutine SUBDIVIDE

The number of pairs of quantized values not counted in "count1" or "rzero" is called bigvalues. SUBDIVIDE splits the scalefactor bands corresponding to this values into three groups. The last one, incomplete generally, counts as a complete one. Region_adress1/2 contains the number of scalefactor bands in the first and the last region, respectively. The number of scalefactor bands in the second region can be calculated using bigvalues. If bigvalues comprises only two scalefactor bands region_adress2 is set to zero. If there are less than two also region_adress1 is zero. The split strategy is up to the implementation. A very simple one for instance is to assign 1/3 of the scalefactor bands to the first and 1/4 to the last region.

Subdivide in case of blocksplit is done analoguous but only two subregions. Region_address 1 is set to a default in this case. This default is 8 in the case of split_point=0 and 9 in the case of split_point=1. Both this values point to the same absolute frequency.

4.7 Calculation of the code book for each subregion

There are 32 different Huffman code tables for the coding of pairs of quantized values available. They differ from each other in the maximum value that can be coded and in the signal statistic they are optimized for. Only codes for values < 16 are in the table. For values >=16 there are two tables provided where the largest value 15 is an escape character. In this case the value 15 is coded in an additional word using a linear PCM code with a word length called linbits. linbits can be calculated by taking the base 2 logarithm of the PCM code, which is x - max_table_entry (see clause 2.4.2.7).

A simple way to choose a table is to use the maximum of the quantized values in a subregion. Tables which have the same size are optimized for different signal statistics. Therefore additional coding gain can be achieved for example by trying all of this tables.

4.8 Counting of the bit necessary to code the values in the subregions

The number of bits necessary to code the quantized values of one subregion is given by:

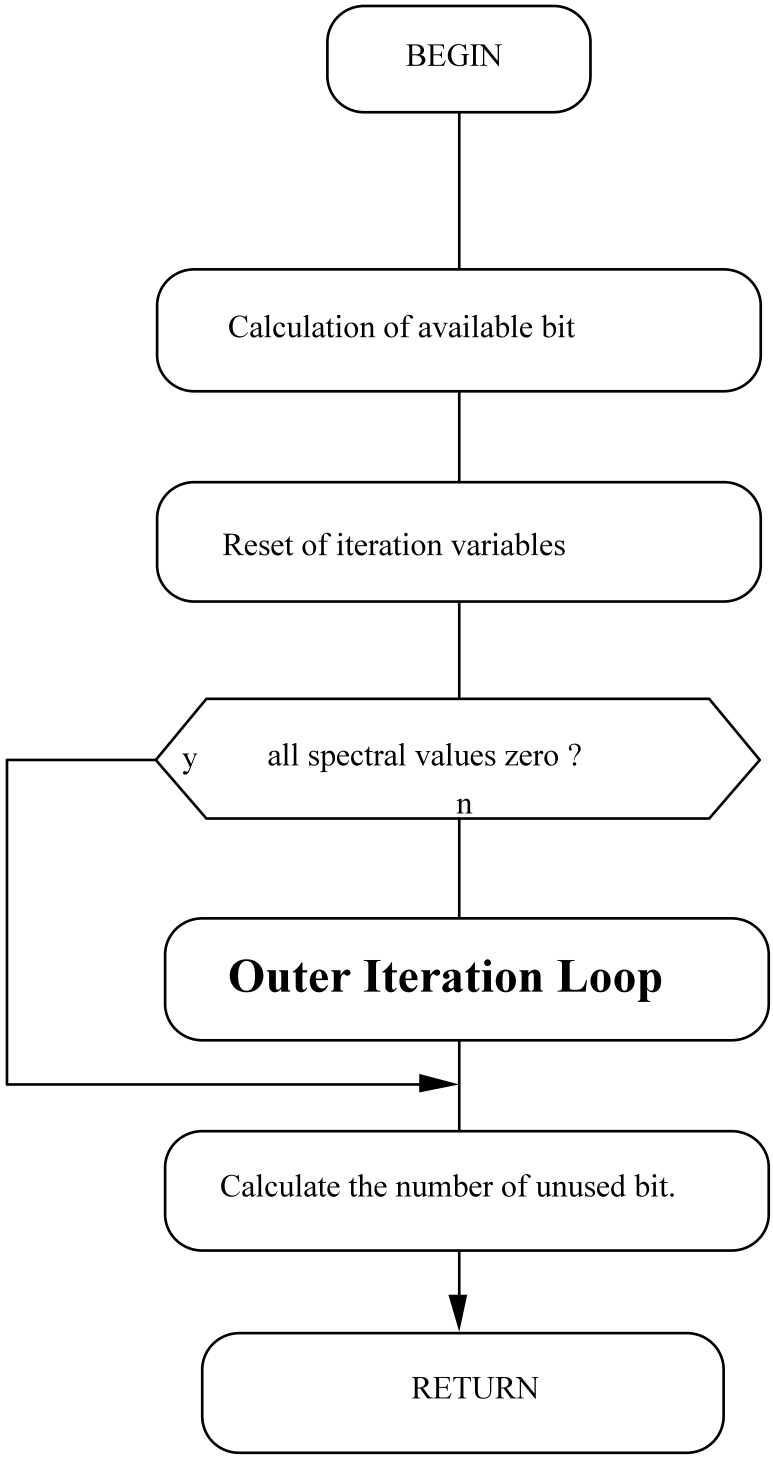
$$\text{bitsum}(j) = \sum_{k=0}^{k=np(j)-1} \text{bitz}\Big(\text{tableselect}(j),\ \min\big(15,\ ix(2k+fe(j))\big),\ \min\big(15,\ ix(2k+fe(j)+1)\big)\Big)$$

$$+ \sum_{k=0}^{k=np(j)-1} \Big(s\big(\ ix(2k+fe(j)) - 15\big) + s\big(\ ix(2k+fe(j)+1) - 15\big)\Big) * \text{linbits}(j)$$

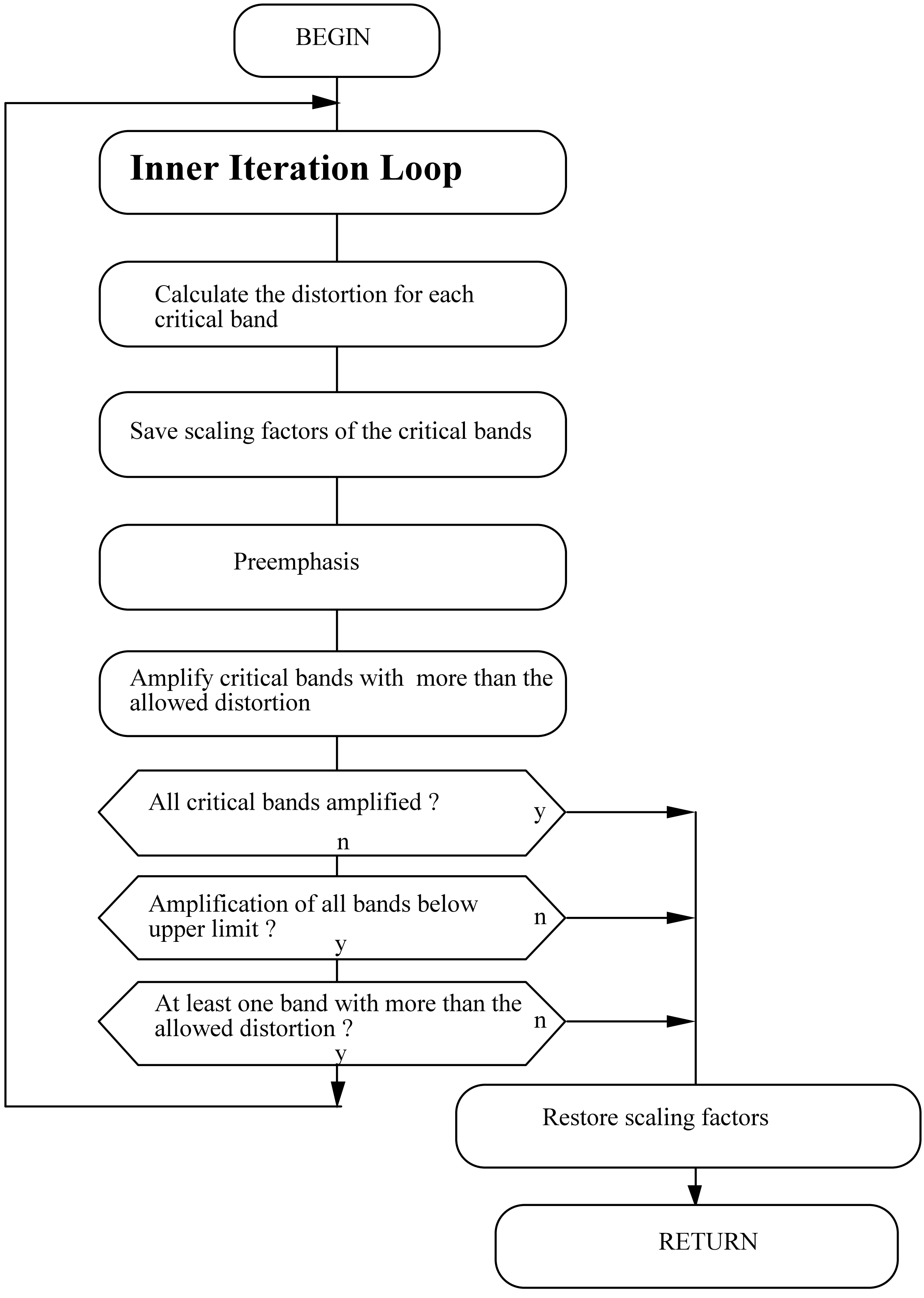
np(j): number of pairs in a sub region
fe(j): number of the first quantized value in a sub-region
bitz: table with Huffman code length

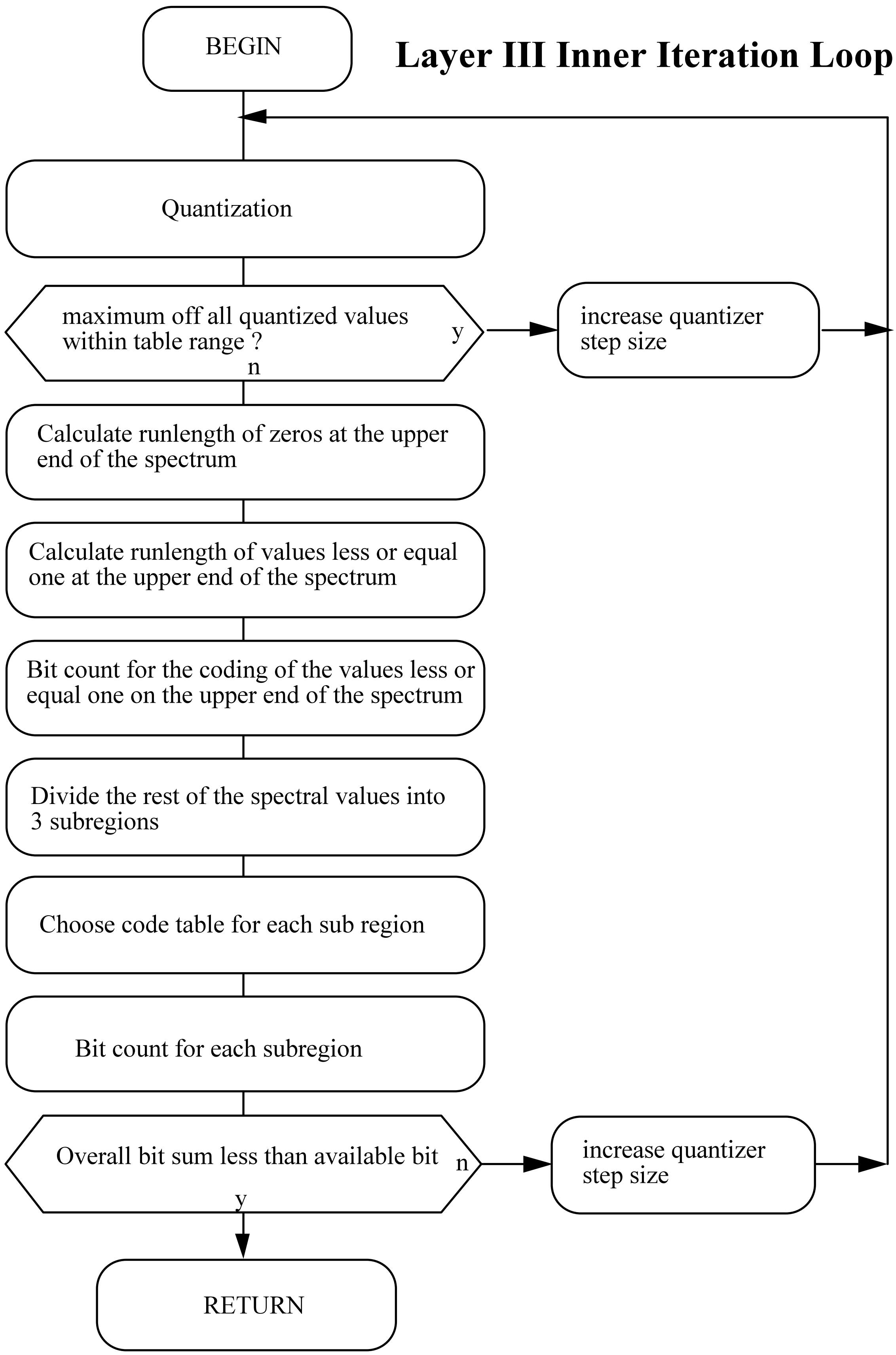
s(...) step function: if $x >= 0$ $s(x) = 1$
if $x < 0$ $s(x) = 0$

Layer III Iteration loops



Layer III Outer Iteration Loop




Layer III Inner Iteration Loop
BEGIN
Quantization
maximum off all quantized values within table range ?
y
n
increase quantizer step size
Calculate runlength of zeros at the upper end of the spectrum
Calculate runlength of values less or equal one at the upper end of the spectrum
Bit count for the coding of the values less or equal one on the upper end of the spectrum
Divide the rest of the spectral values into 3 subregions
Choose code table for each sub region
Bit count for each subregion
Overall bit sum less than available bit
n
y
increase quantizer step size
RETURN