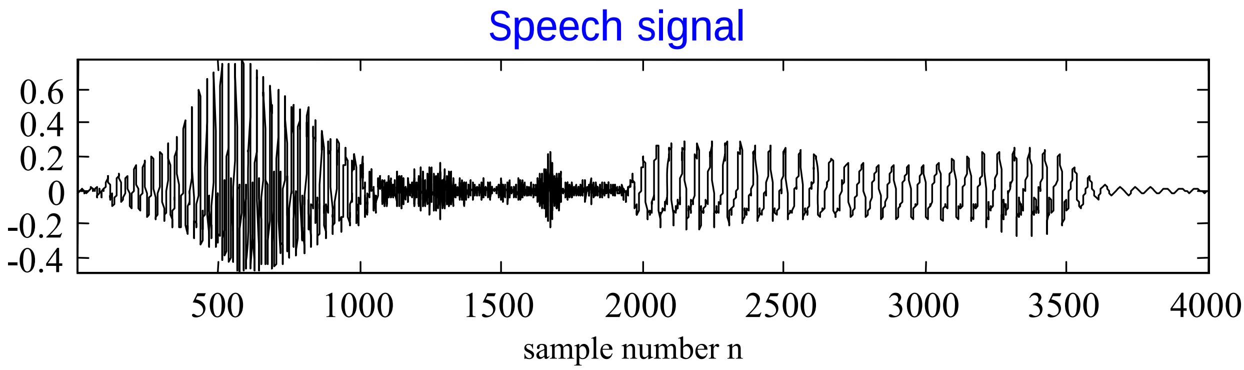
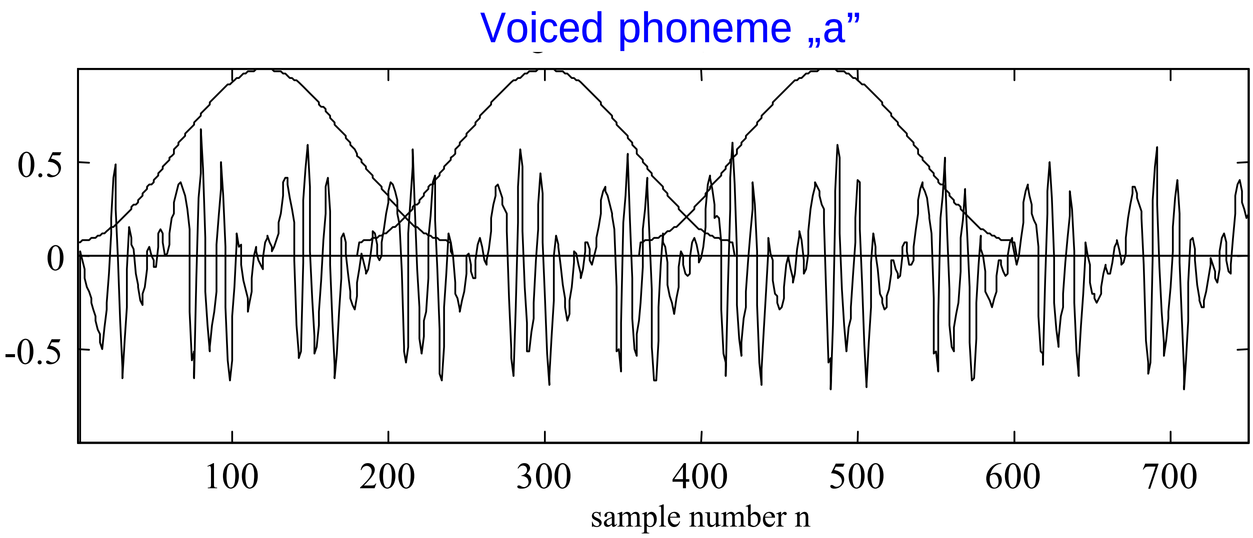
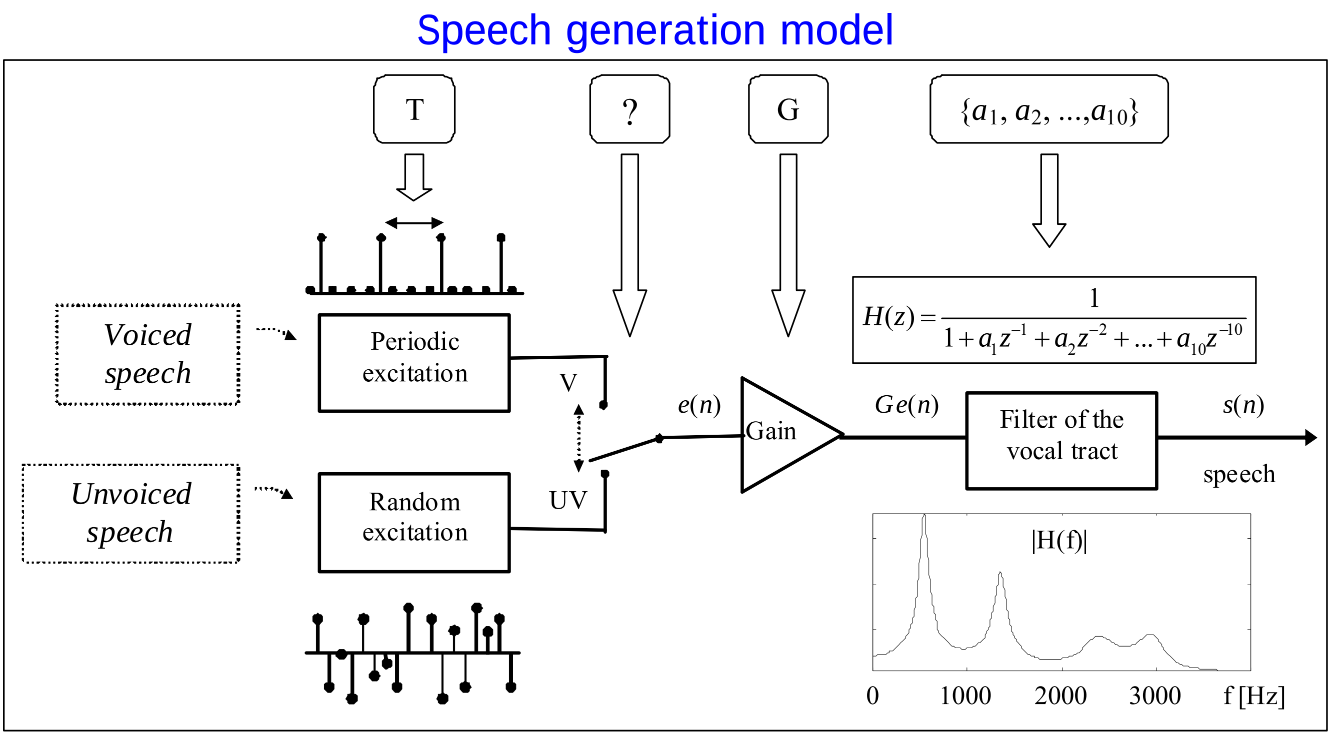
LECTURE 17: DSP applications

17.1. LPC-10 algorithm for speech coding







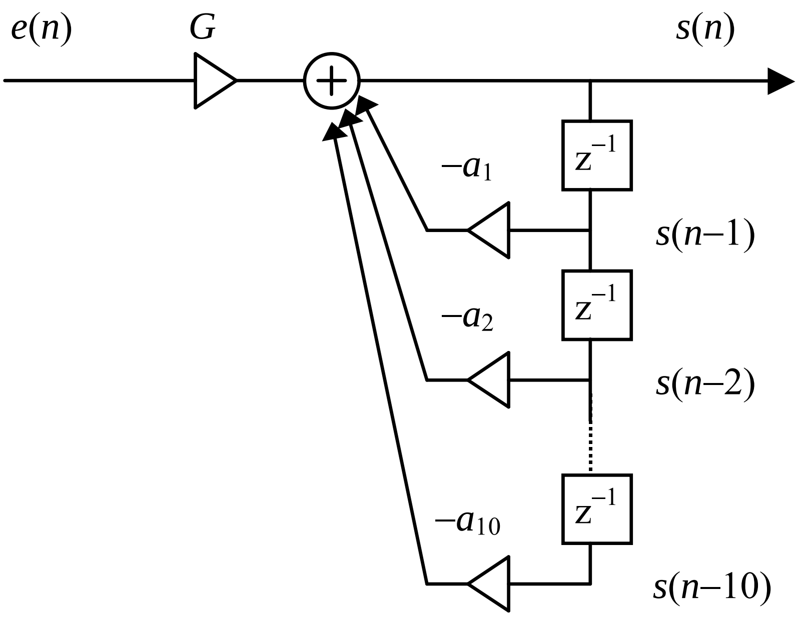
Vocal tract filter

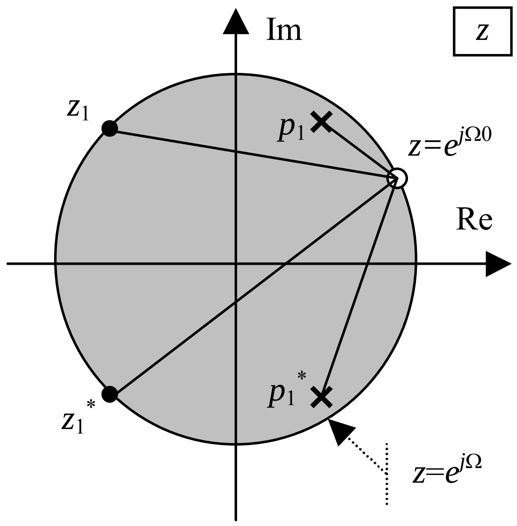
IIR digital filter, 10 poles: 2 for one resonant frequency

$$H(z)=\frac{G}{A(z)}=\frac{G}{1+a_1z^{-1}+a_2z^{-2}+\ldots+a_{10}z^{-10}}$$

$$H(z)=\frac{G}{(1-p_1z^{-1})(1-p_1^*z^{-1})\ldots(1-p_5z^{-1})(1-p_5^*z^{-1})}$$

$$s(n)=G\cdot e(n)-\sum_{k=1}^{10}a_k s(n-k)$$





When synthesized speech $s(n)\cong$ real speech $x(n)$:

$$s(n)=G\cdot e(n)-\sum_{k=1}^{10}a_k s(n-k) \;\rightarrow\; x(n)=G\cdot e(n)-\sum_{k=1}^{10}a_k x(n-k)$$

$$x(n)=Ge(n)+\left[-a_1x(n-1)-a_2x(n-2)-\ldots-a_{10}x(n-10)\right]$$

$$x(n)=\hat{x}(n)+err(n)$$

$$\hat{x}(n)=-a_1x(n-1)-a_2x(n-2)-\ldots-a_{10}x(n-10)=-\sum_{j=1}^{p}a_j x(n-j)$$

$$err(n)=Ge(n)$$

Block-based processing.

Cost function (criteria) of the best signal self-prediction (least-squares):

$$J=\sigma^2=\frac{1}{N-p}\sum_{n=p}^{N-1}err^2(n)=\frac{1}{N-p}\sum_{n=p}^{N-1}\left[x(n)-\hat{x}(n)\right]^2$$

$$J=\frac{1}{N-p}\sum_{n=p}^{N-1}\left[x(n)+\sum_{j=1}^{p}a_j x(n-j)\right]^2$$

$J\rightarrow \min$

For $1\le k\le p$:

$$\frac{\partial J}{\partial a_k}=\frac{\partial}{\partial a_k}\left\{\frac{1}{N-p}\sum_{n=p}^{N-1}\left[x^2(n)+2x(n)\sum_{j=1}^{p}a_j x(n-j)+\left(\sum_{j=1}^{p}a_j x(n-j)\right)^2\right]\right\}=0$$

p equation with *p* unknowns.

Solution:

$$\mathbf{a}=-\mathbf{R}^{-1}\mathbf{r}$$

gdzie:

$$\mathbf{a}=\begin{bmatrix}a_1\\a_2\\\vdots\\a_p\end{bmatrix},\qquad \mathbf{R}=\begin{bmatrix}r(0)&r(1)&\cdots&r(p-1)\\r(1)&r(0)&\cdots&r(p-2)\\\vdots&\vdots&\ddots&\vdots\\r(p-1)&r(p-2)&\cdots&r(0)\end{bmatrix},\qquad \mathbf{r}=\begin{bmatrix}r(1)\\r(2)\\\vdots\\r(p)\end{bmatrix}$$

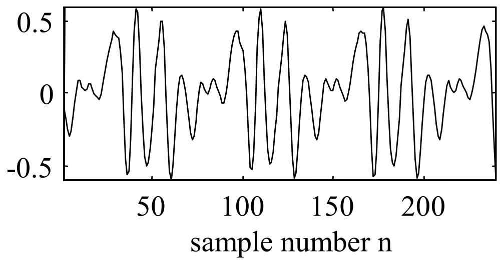
$$J_{\min}=\sigma^2_{\min}=r(0)+\mathbf{a}^T\mathbf{r}=r(0)+\sum_{j=1}^{p}a_j r(j)$$

$$r(k)=\frac{1}{N-p}\sum_{n=p}^{N-1}x(n)x(n-k)$$

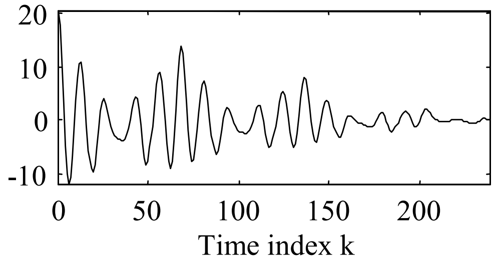
Decision voiced / unvoiced

for phoneme „a”

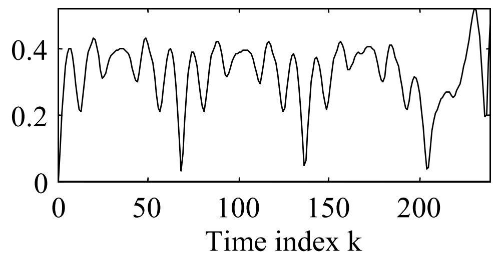
Signal “a”



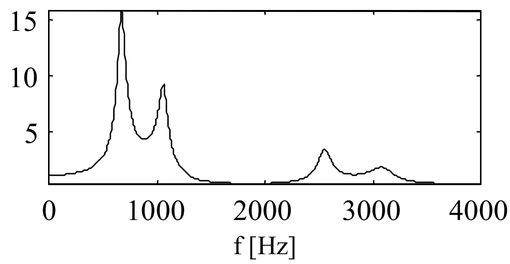
Autocorrelation of “a”



AMDF of “a”

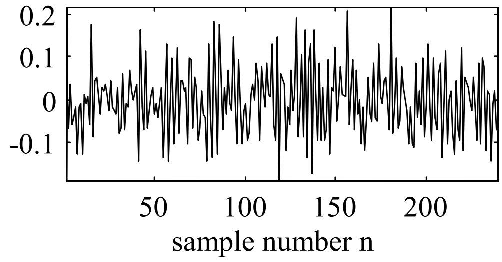


|H(f)| for “a”

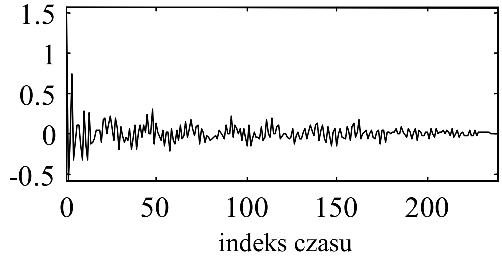


for phoneme „sh”

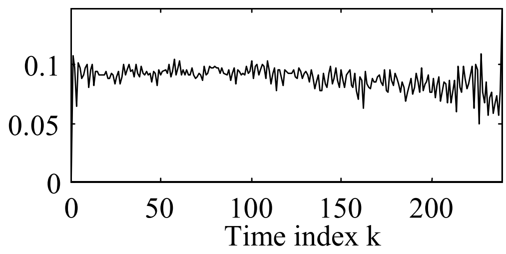
Signal “sh”



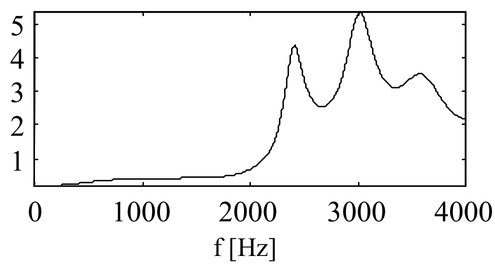
Autocorrelation of “sh”



AMDF of “sh”



|H(f)| for “sh”



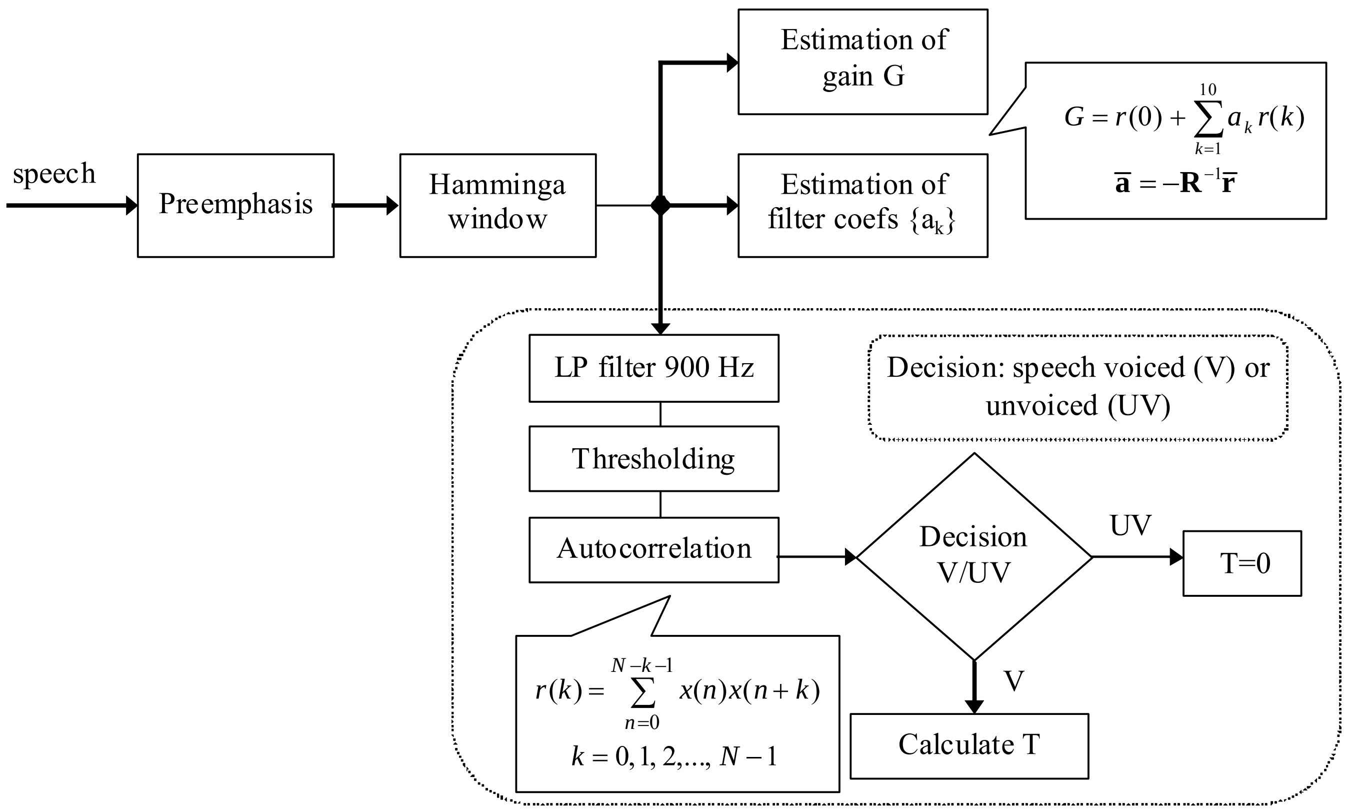
LPC-10 speech coding algorithm

Every 180 samples of speech → 12 parameters $\{T, G, a_1, a_2, ..., a_{10}\}$.

180*8b = **1440 bitów** **54 bits** (after quantization)

Compression ratio = 27

Algorthm	Compression	Bitstream [kbits/s]
PCM (G711)	1 : 1	64
ADPCM (G721)	2 : 1	32
LD-CE**LP** (G728)	4 : 1	16
RPE-**LP**T (GSM)	5 : 1	13
VSE**LP**	8 : 1	8
CE**LP** (standard USA)	13 : 1	4,8
LPC-10e (standard USA)	27 : 1	2,4



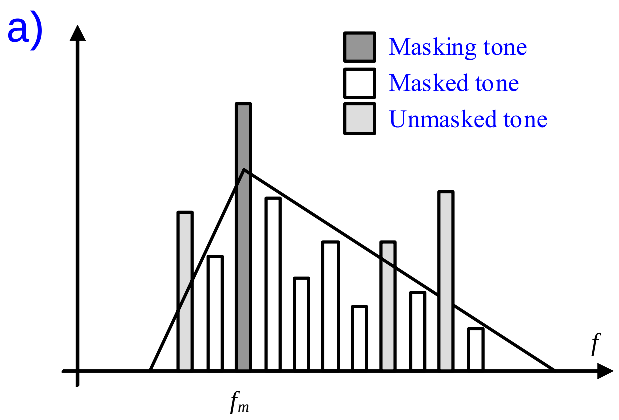
Matlab program

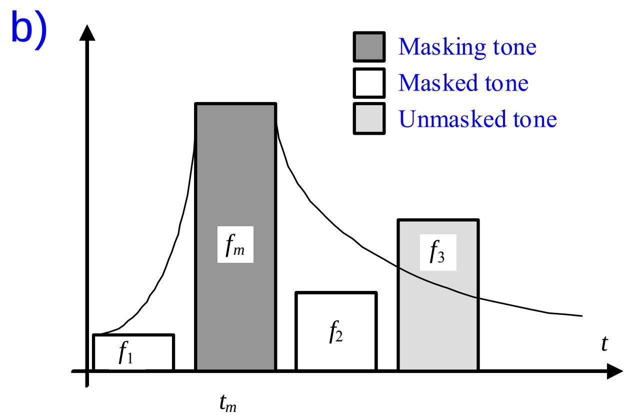
```
[x,fs,Nbits]=wavread('speech.wav');    % read speech signal
plot(x); title('speech'); pause        % display it
soundsc(x,fs);                         % play it on loudspeakers (headphones)
N=length(x);                           % signal length
Mlen=240;                              % analyzed fragment length (number of samples)
Mstep=180;                             % analyzed fragment shift (number of samples)
Np=10;                                 % prediction order (IIR-AR filter order)
where=181;                             % initial position of voiced excitation

lpc=[];                                % table for calculated speech model coeffs
s=[];                                  % whole synthesized speech
ss=[];                                 % one fragment (block) of synthesized speech
bs=zeros(1,Np);                        % buffer for speech fragment
Nramek=floor((N-240)/180+1);           % number of speech fragments (blocks) to be analyzed

% x=filter([1 -0.9735], 1, x);         % optional filtering (pre-emphasis) – optional

for  nr = 1 : Nramek
     % take new data block (fragment of speech samples)
     n = 1+(nr-1)*Mstep : Mlen + (nr-1)*Mstep;
     bx = x(n);

     % ANALYSIS – calculate speech model parameters
     bx = bx - mean(bx);                                      % remove mean value
     for k = 0 : Mlen-1
          r(k+1) = sum( bx(1 : Mlen-k) .* bx(1+k : Mlen) );   % autocorrelation
     end
  %  subplot(411); plot(n,bx); title('speech fragment');
  %  subplot(412); plot(r); title('its autocorrelation');

     offset=20; rmax=max( r(offset : Mlen) );           % find max of autocorrelation
     imax=find(r==rmax);                                % find its position
     if ( rmax > 0.35*r(1) )  T=imax; else T=0; end     % decision V/UV?
     if (T>80) T=round(T/2); end                        % second sub-harmonics found
     T                                                  % display value of T
     rr(1:Np,1)=(r(2:Np+1))';
     for m=1:Np
          R(m,1:Np)=[r(m:-1:2) r(1:Np-(m-1))];          % build autocorrelation matrix
     end
     a=-inv(R)*rr;                                      % find coeffs of prediction filter
     gain=r(1)+r(2:Np+1)*a;                             % find gain
     H=freqz(1,[1;a]);                                  % filter freq response
 %   subplot(413); plot(abs(H)); title('Freq response of vocal tract');

  %  lpc=[lpc; T; gain; a; ];                           % store parameter values
```

```
    % SYNTHESIS – generate speech using calculated parameters
    % T = 0; % remove „%” and set: T = 80, 50, 30, 0
    if (T~=0) where=where-Mstep; end                 % where next voiced excitation?
    for n=1:180
         if( T==0)
             exc=2*(rand(1,1)-0.5); where=271;       % random excitation
         else
             if (n==where) exc=1; where=where+T;     % voiced excitation
             else exc=0; end
         end
         ss(n)=gain*exc-bs*a;              % filtration of artificial excitation
         bs=[ss(n) bs(1:Np-1) ];           % shifting output buffer
    end
  %  subplot(414); plot(ss); title('synthesized speech fragment'); pause
    s = [s ss];                            % store synthesized speech fragment
   end
% s=filter(1,[1 -0.9735],s);               % filtratiom (de-emphasis) – inverse filter – optional

plot(s); title('synthesized speech'); pause
soundsc(s,fs)
```

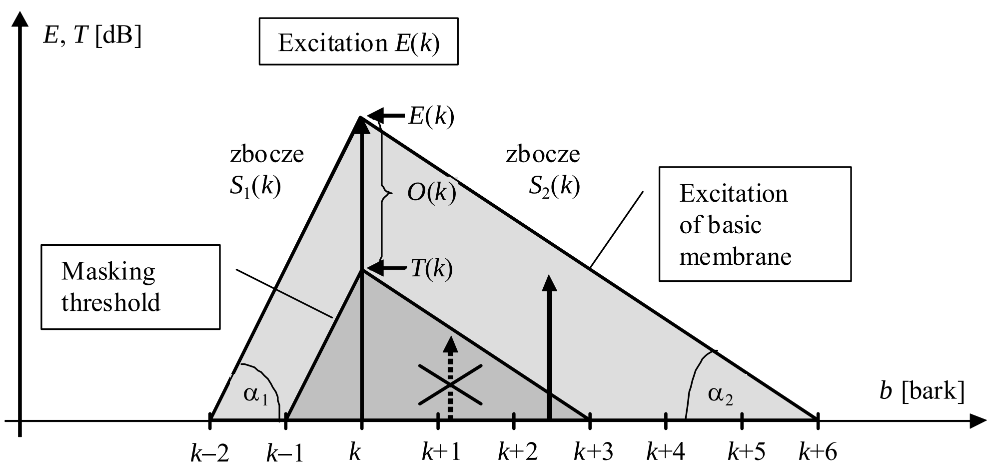
17.2. MPEG-2 audio coding layer 2

Illustration of masking effects in: frequency (a) and time (b)

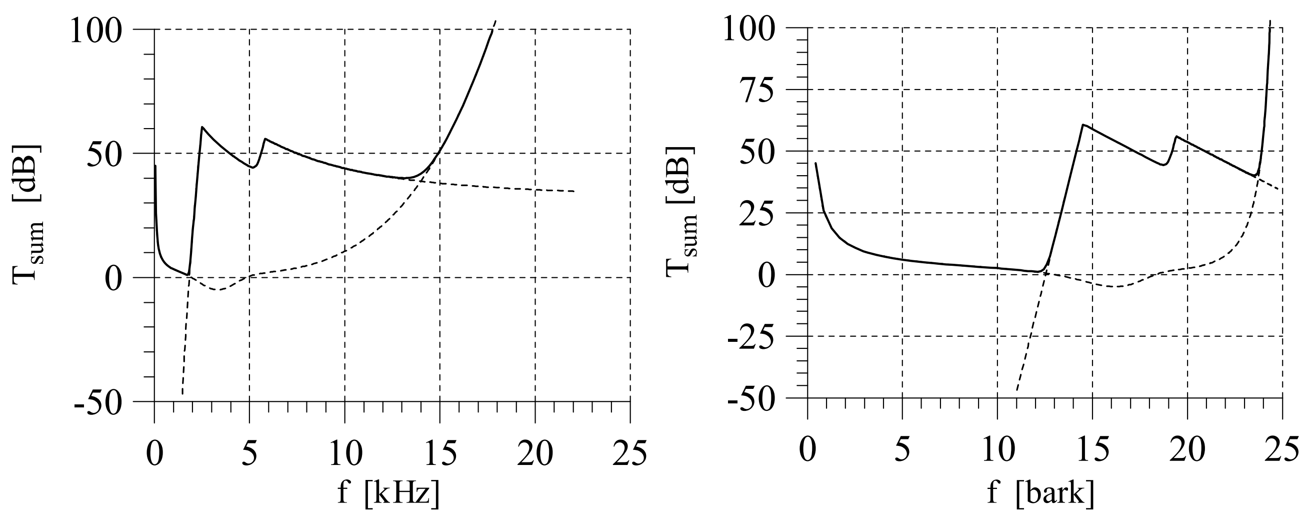




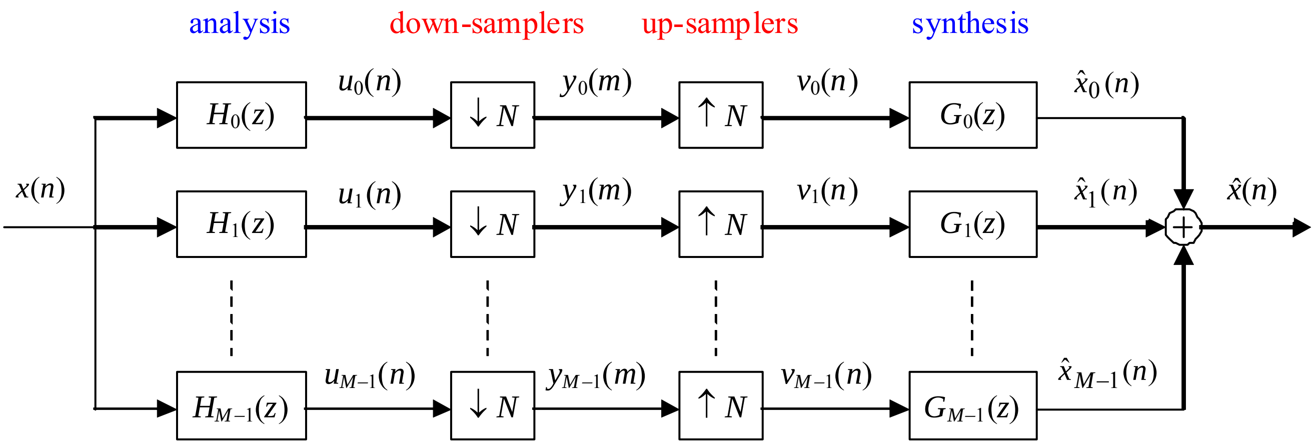
Graphical illustration of frequency masking

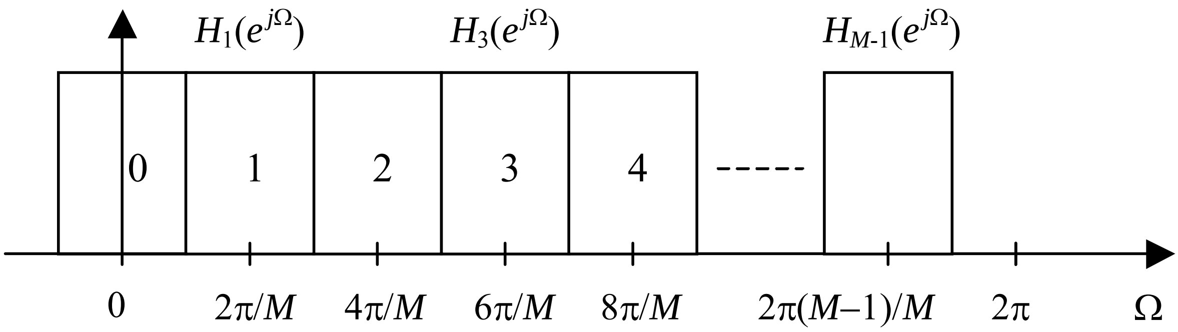


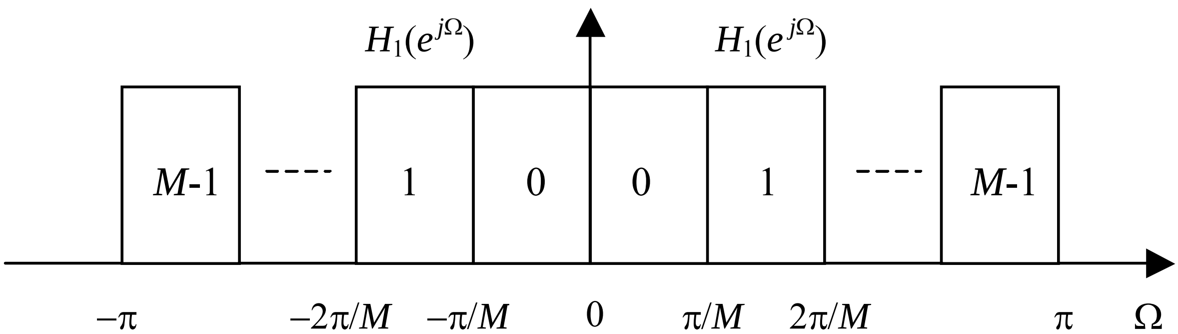
Masking and absolute threshold of hearing

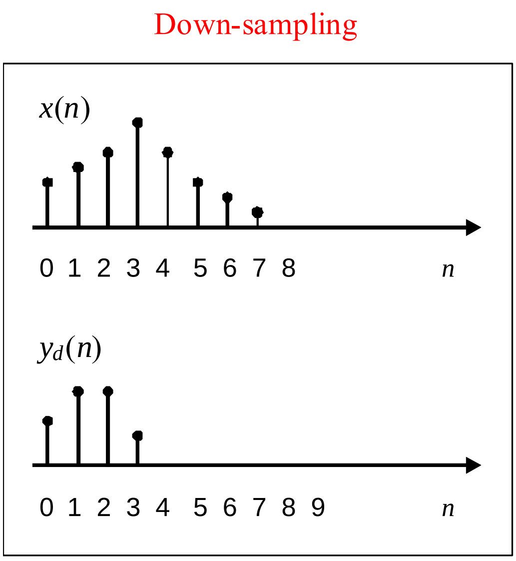


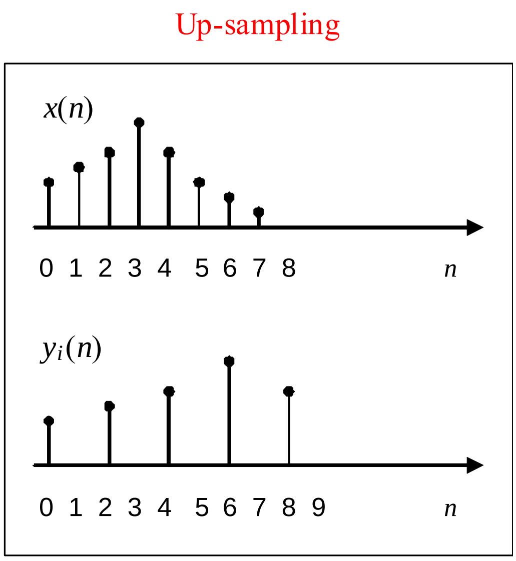
FILTER BANKS











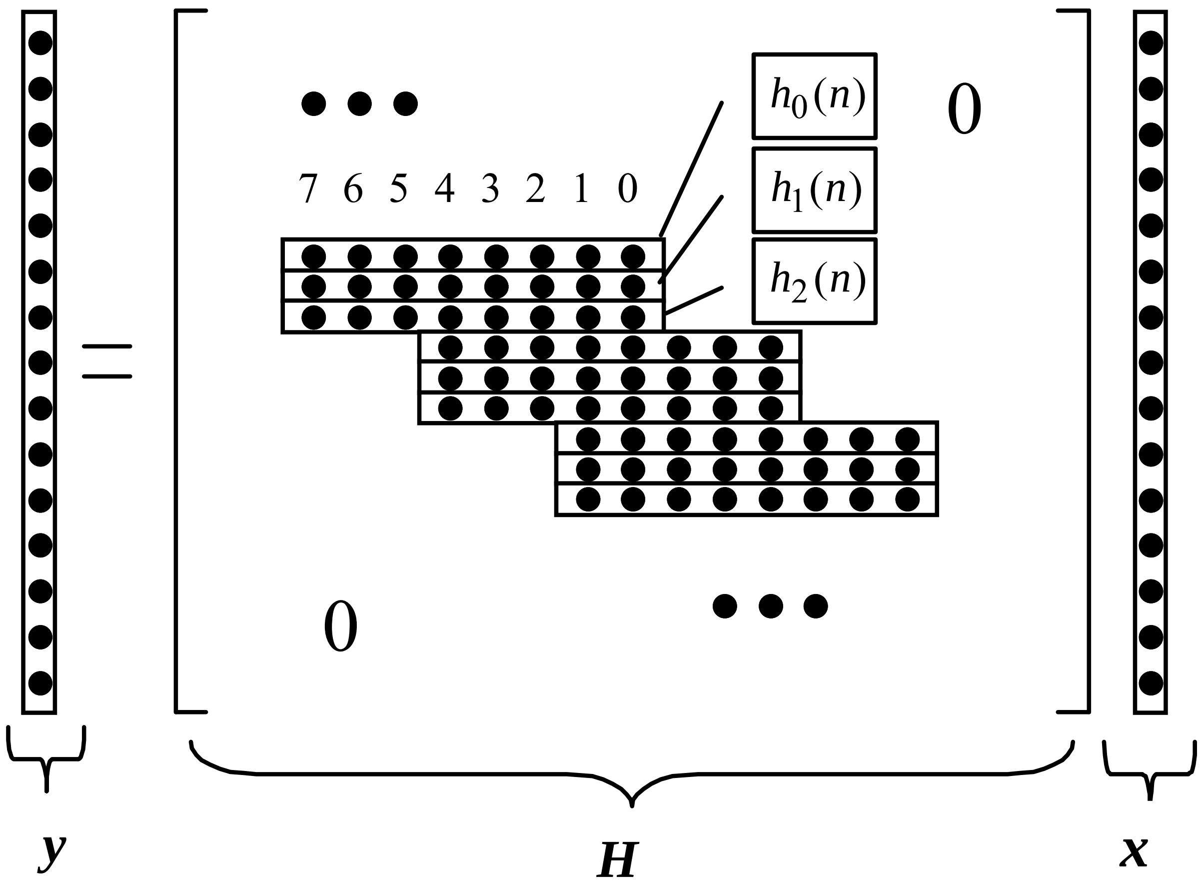
FILTER BANKS (cont.)

Simple explanation =

1) **Inner product of audio signal with many impulse responses** of the FB filters (having *N* samples each).
2) **Shifting impulse responses** in time by *M* samples.
3) Again: inner product of signals with impulse responses of the FB.
4) Shift in time.

...etc....

In MPEG-audio: N=512, M=32.



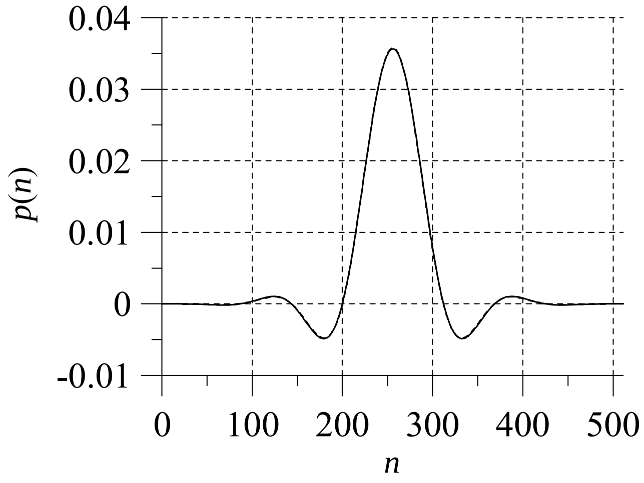
FILTER BANKS (cont.)

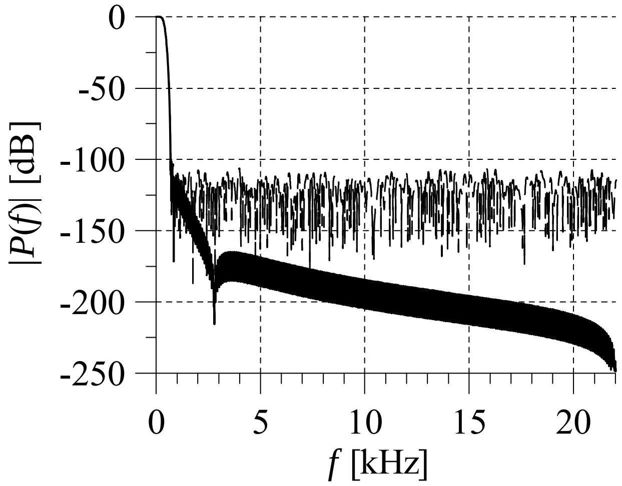
Impulse responses of 32 filters in MPEG-audio standard:

$$h_k(n) = 2\,p(n)\cos\left(\frac{\pi}{M}(k+0,5)(n-\frac{L-1}{2}) + (-1)^k\frac{\pi}{4}\right) \qquad \text{(analysis)}$$

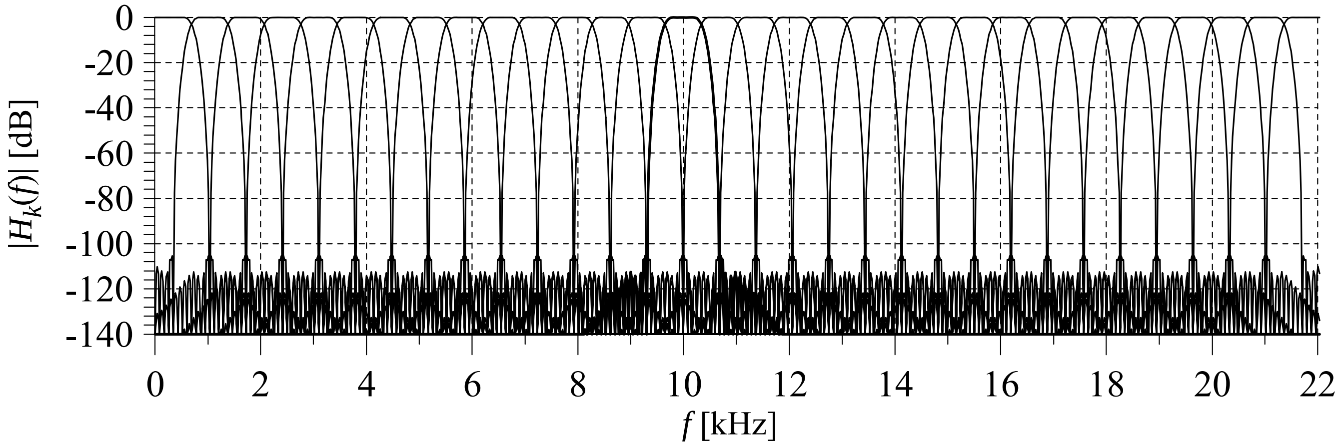
$$g_k(n) = 2\,p(n)\cos\left(\frac{\pi}{M}(k+0,5)(n-\frac{L-1}{2}) - (-1)^k\frac{\pi}{4}\right) \qquad \text{(synthesis)}$$

$p(n)$ – prototype filter that is modulated using cos(.) functions

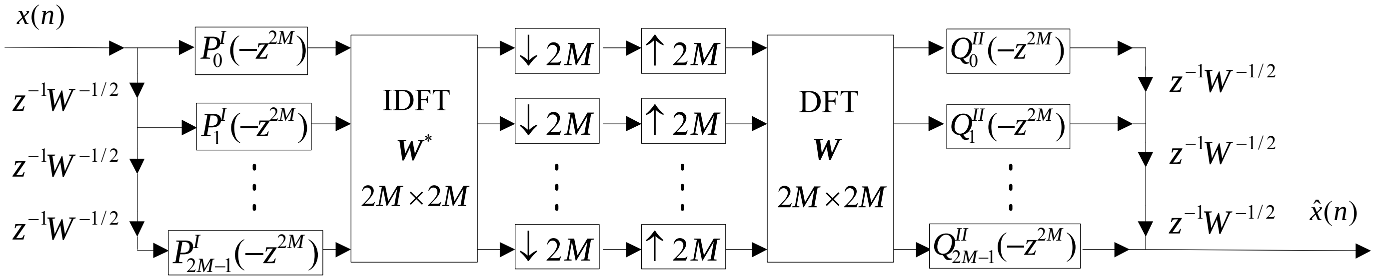


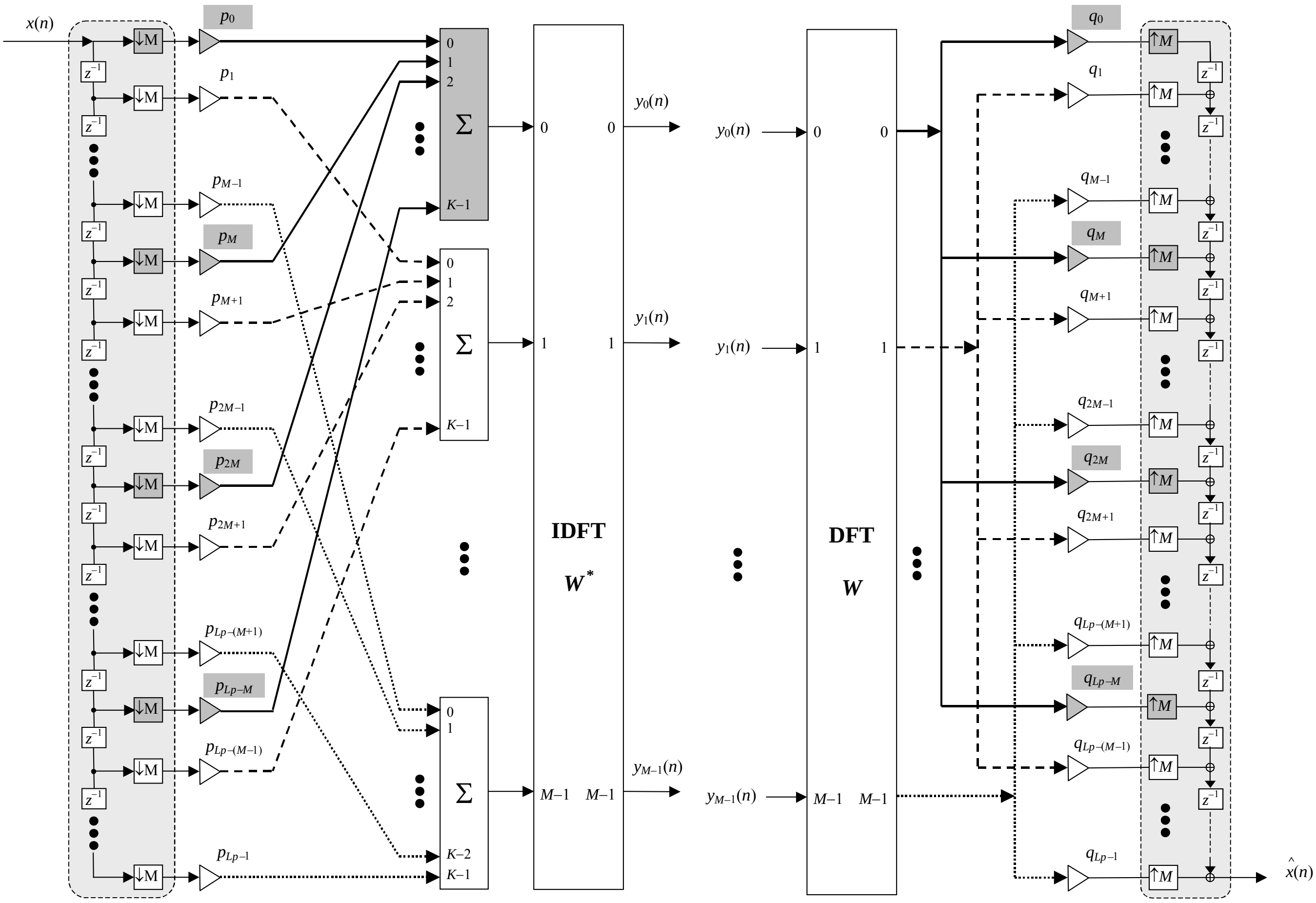


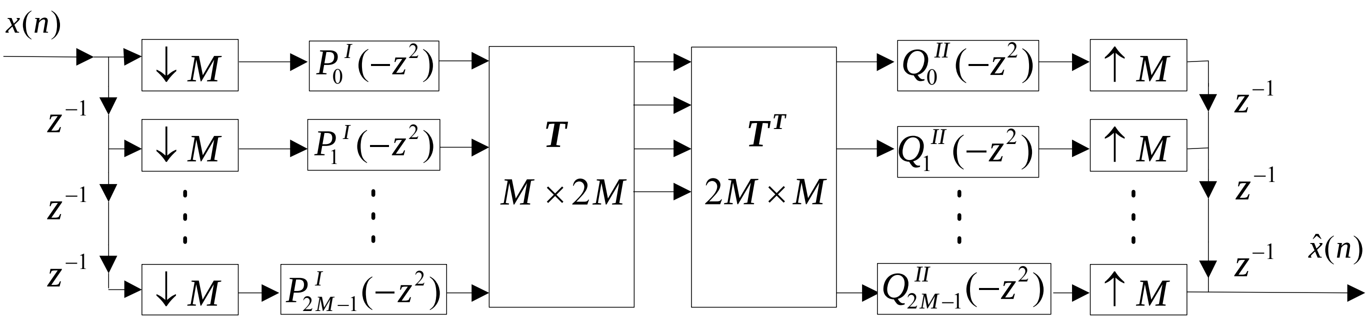
Frequency responses of all filters in the bank of filters



Since $\cos(\omega t) = \dfrac{e^{j\omega t} + e^{j\omega t}}{2}$, **then** DCT → DFT.





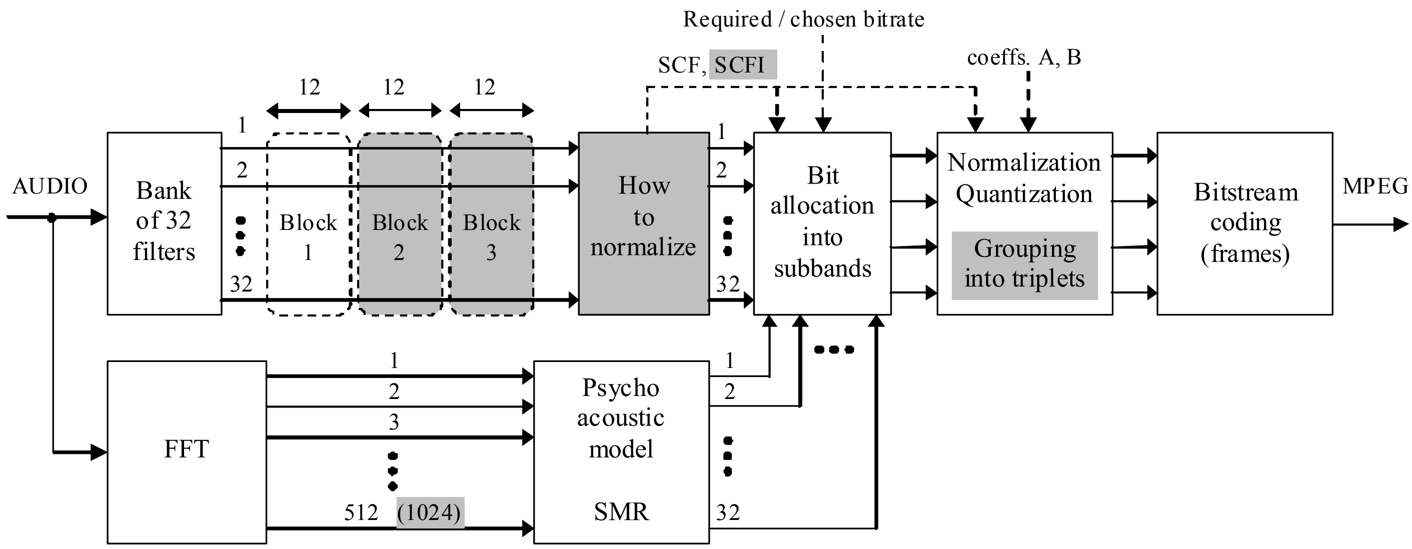


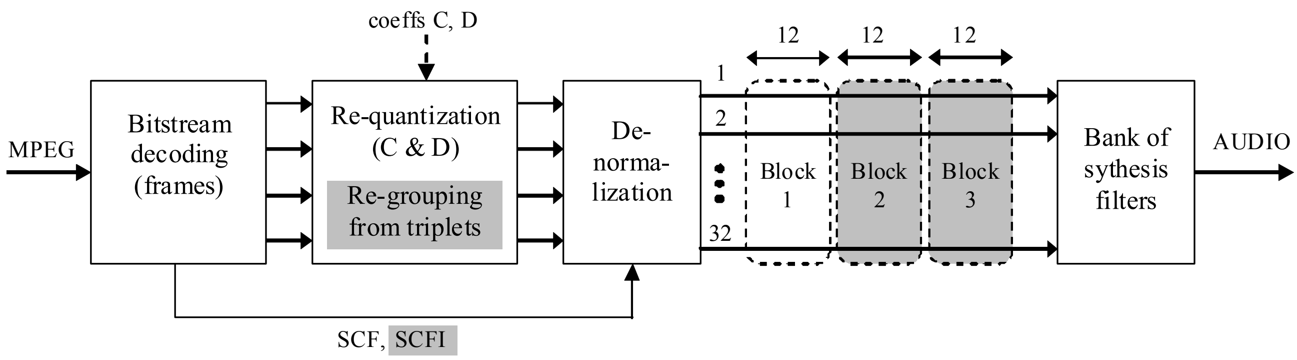
$$t(k,m) = 2\cos\left(\frac{\pi}{M}(k+1/2)(m-\frac{L-1}{2}) + (-1)^k \pi/4\right)$$

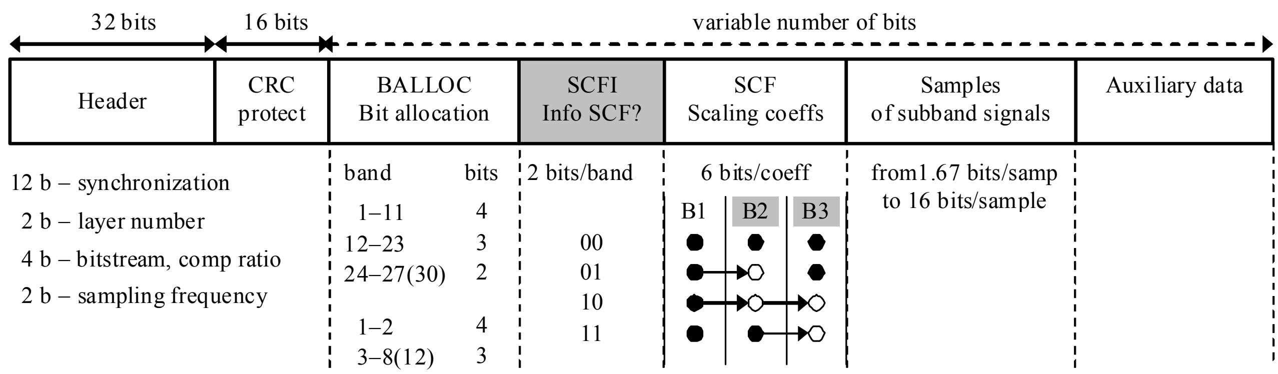
Block diagram of MPEG-1 audio MP1 and MP2 encoder and decoder

Input AUDIO bit-stream: 2*768 kbits/s (2 channels * 16 bits * 48 kHz)
Output MPEG bit-stream: from 2*32 to 2*192 kbits/s

Gray color denotes MP2 extensions in respect to MP1.







Tab. A. Coefficients used for scaling of subband samples

Number	Coefficient
0	2,00000000000000
1	1,58740105196820
2	1,25992104989487
⋮	⋮
60	0,00000190734863
61	0,00000151386361
62	0,00000120155435

Tab. B. Classes of scaling coefficients in each frequency sub-band for blocks B_1 and B_2 (by analogy for B_2 and B_3)

$\Delta nr(B_{12}) = nr(B_1) - nr(B_2)$	Class B_{12}
$\Delta nr(B_{12}) \leq -3$	1
$-3 < \Delta nr(B_{12}) < 0$	2
$\Delta nr(B_{12}) = 0$	3
$0 < \Delta nr(B_{12}) < 3$	4
$3 \leq \Delta nr(B_{12})$	5

Tab. C. Choosing scaling coeffs for each frequency subbands from coeffs for blocks B_1, B_2, B_3

KlasaB_{12}	KlasaB_{23}	Wybór	Kod
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
⋮	⋮	⋮	⋮
5	5	1 2 3	0

Tab. D. Exaple of bit allocation (**quantization levels**) for frequency subbands

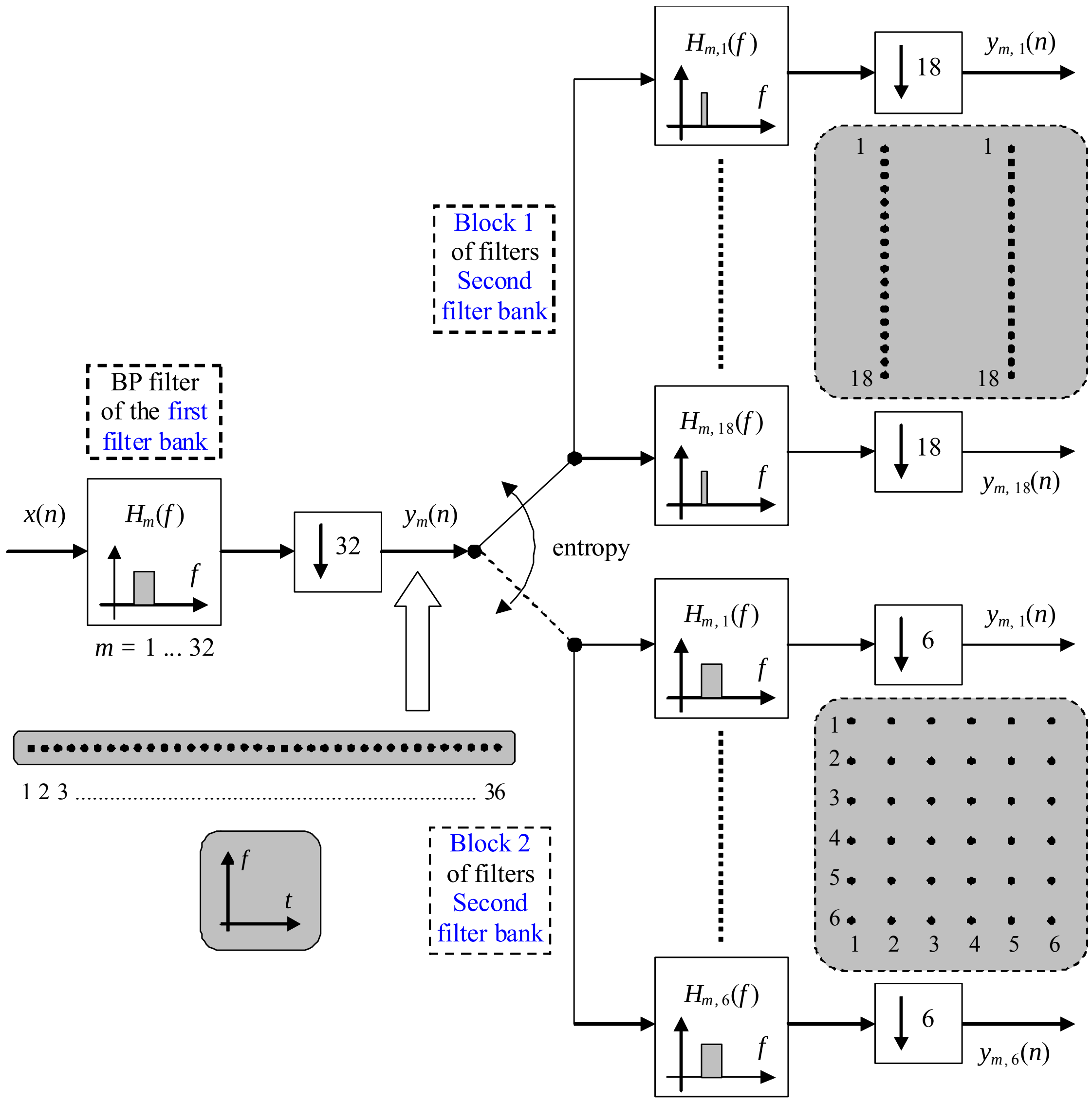
Subband number	Bits per index	Sent index (column number)														
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
1 – 3	4	3	7	15	31	63	127	255	512	1023	2047	4095	8191	16383	32767	65535
4 – 11	4	3	5	7	9	15	31	63	127	255	512	1023	2047	4095	8191	65535
12 – 23	3	3	5	7	9	15	31	65535								
24 – 30	2	3	7	65535												

Tab. E. Classes of quatization (A, B) and requatization (C, D) of subband samples for MP2 layer (CW=coding word)

Levels of quantization	Coeff. A	Coeff. B	Coeff. C	Coeff. D	SNR [dB]	Samples per CW	Bits per CW
3	0,750000000	−0,250000000	1,33333333333	0,50000000000	0,00	3	5
5	0,625000000	−0,375000000	1,60000000000	0,50000000000	7,00	3	7
7	0,875000000	−0,125000000	1,14285714286	0,25000000000	11,00	1	3
9	0,562500000	−0,437500000	1,77777777777	0,50000000000	16,00	3	10
15	0,937500000	−0,062500000	1,06666666666	0,12500000000	20,84	1	4
31	0,968750000	−0,031250000	1,03225806452	0,06250000000	25,28	1	5
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
65535	0,999984741	−0,000015259	1,00001525902	0,00003051758	98,01	1	16

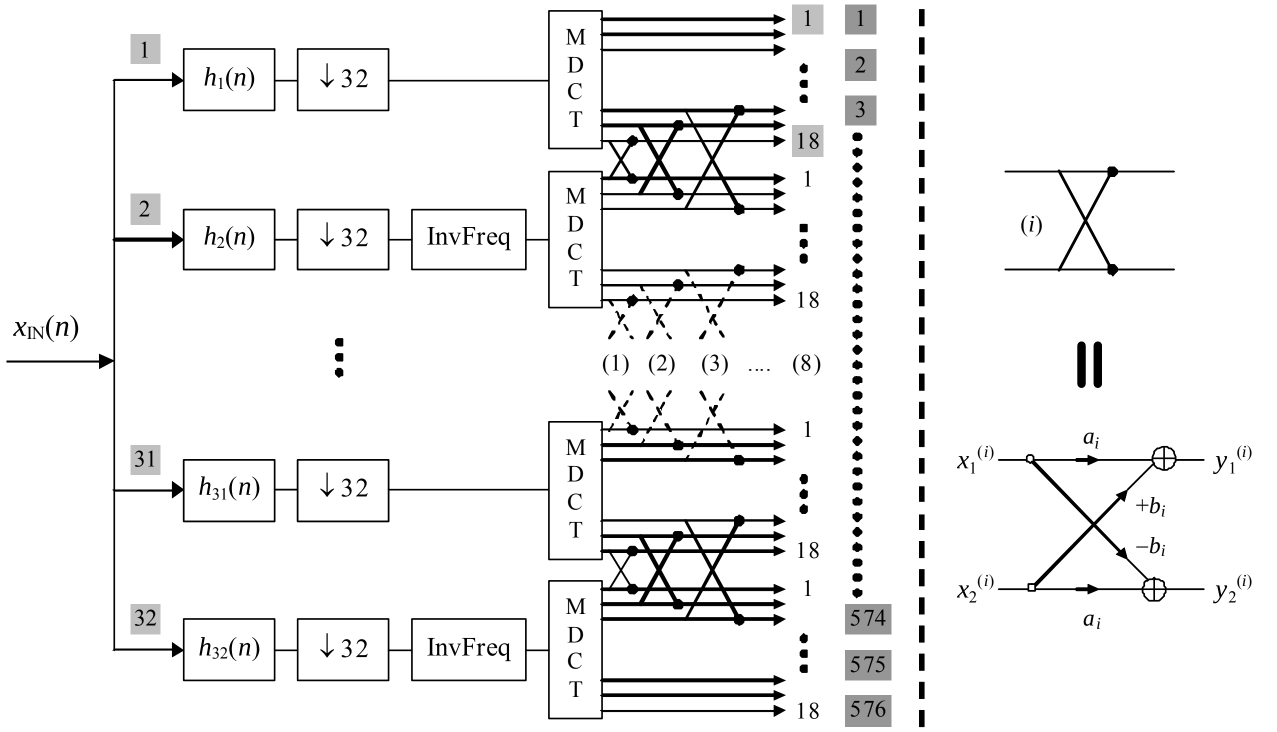
Signal filtering in MPEG-1 MP3 layer

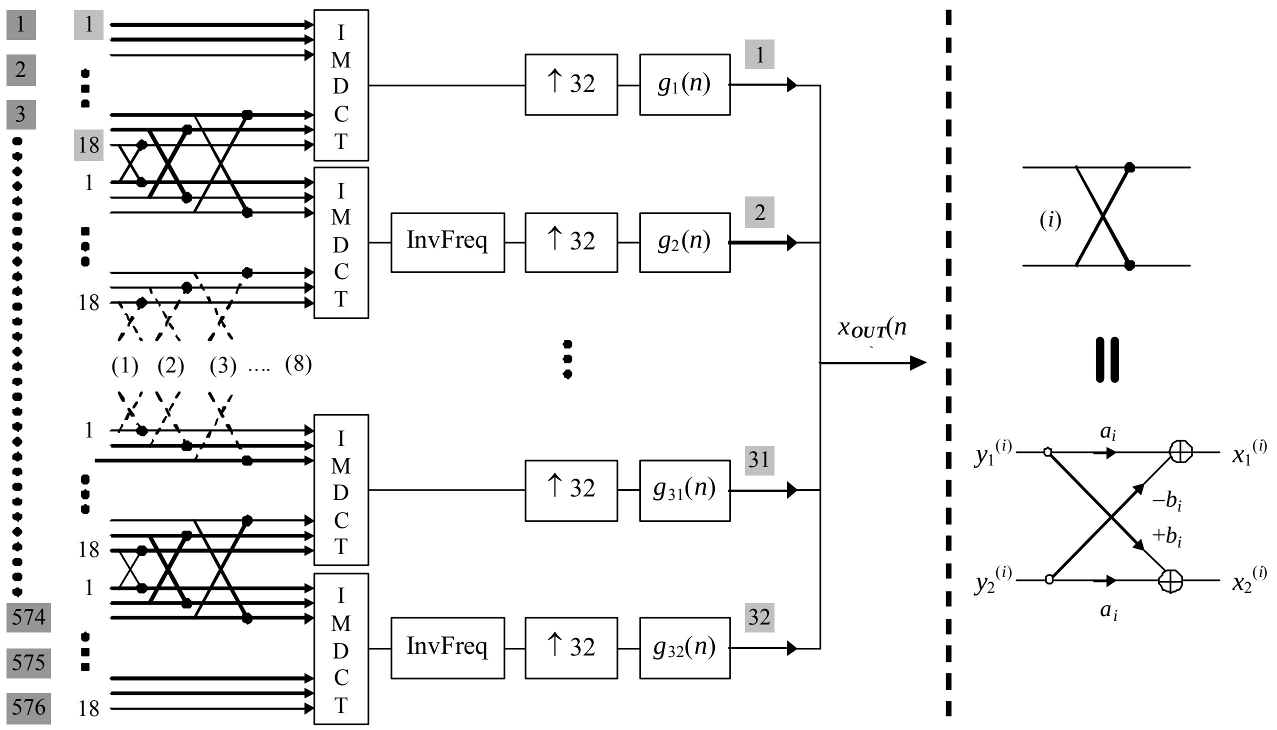
$\downarrow M$ – M-times down-sampling



Signal filtering in MPEG-1 MP3 layer (cont.)

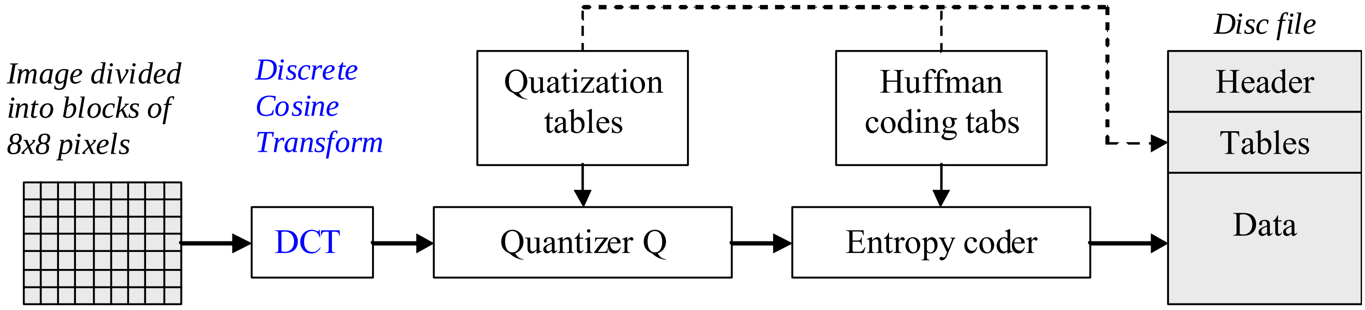
MDCT – modified discrete cosine transform
IMDCT – inverse modified discrete cosine transform
$h(n)$ – filter impulse response
$\downarrow M$ – M-times down-sampling
$\uparrow M$ – M-times up-sampling



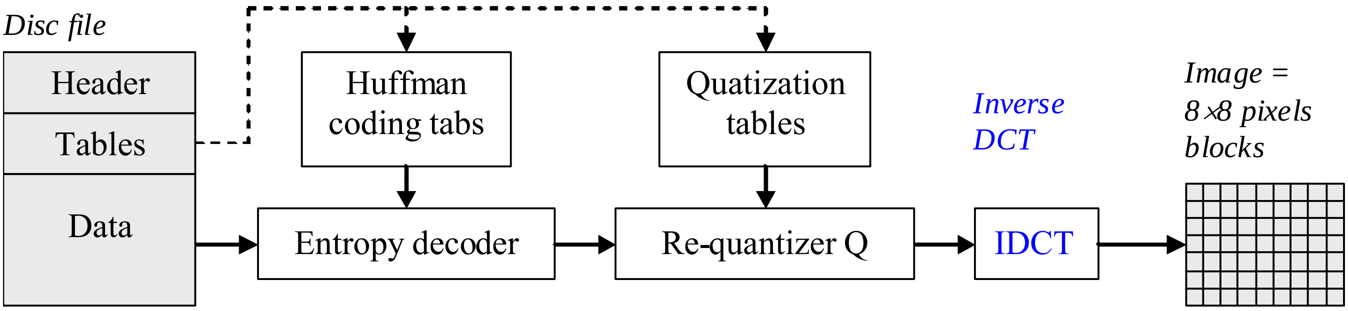


17.3. JPEG still image compression

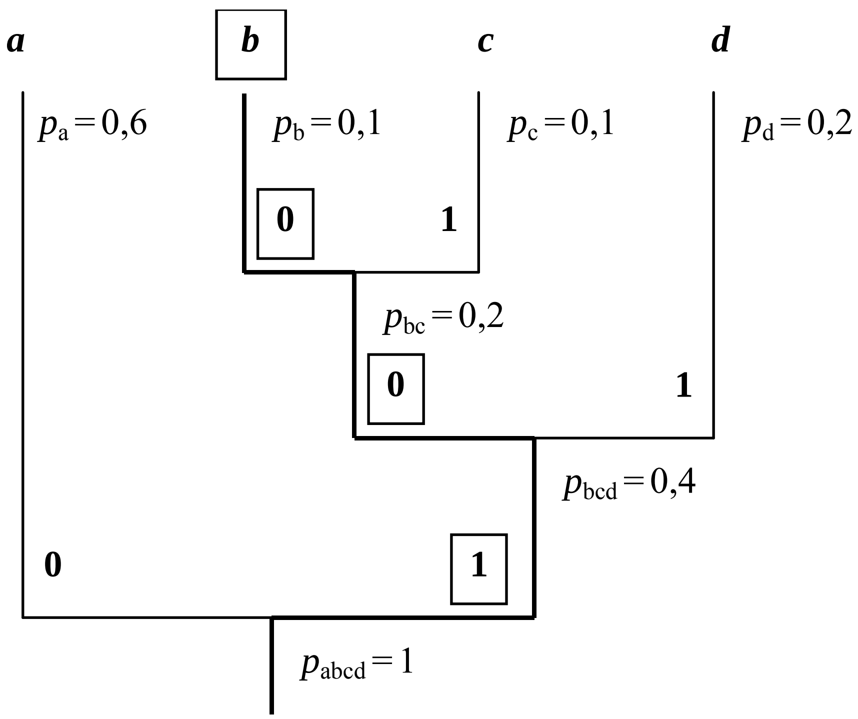
JPEG encoder



JPEG decoder

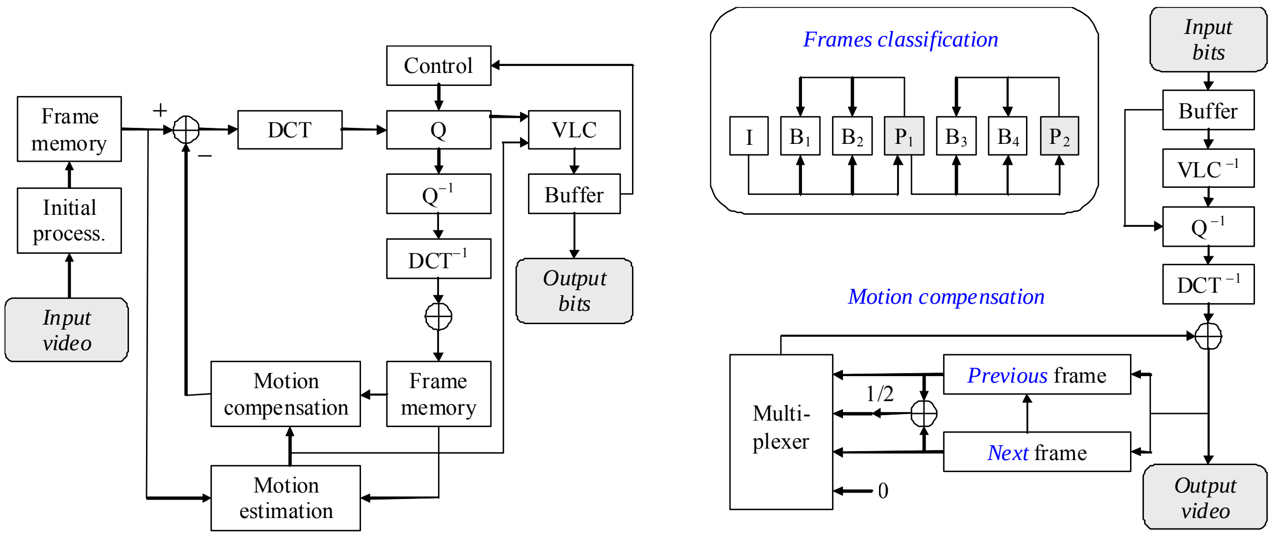


Huffman coding

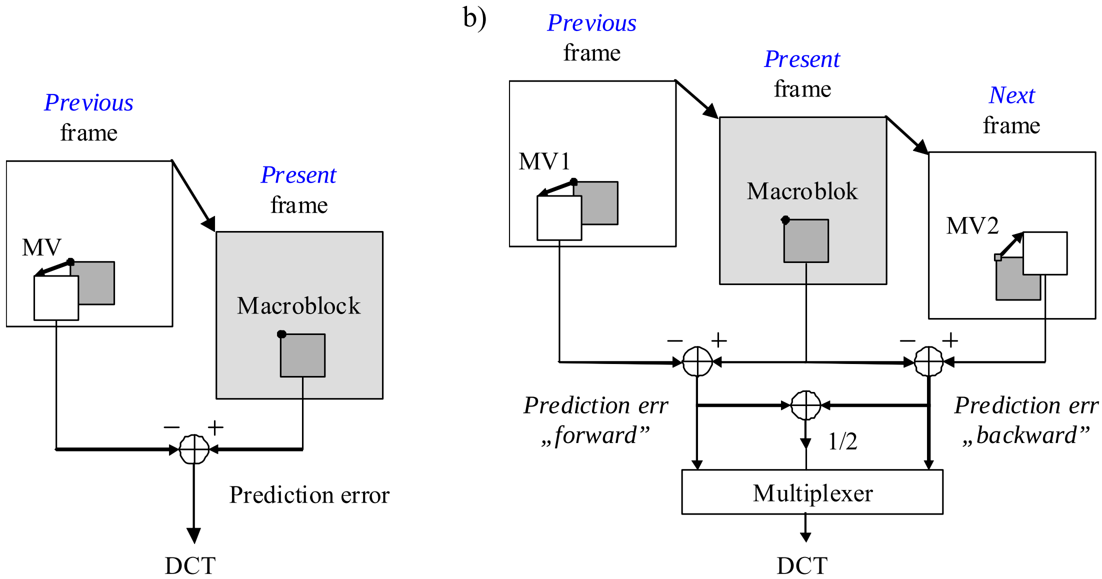


17.4. MPEG-2 video coding

DCT – 2D discrete cosine transform
Q – quantizer
VLI – lossless coding (*Variable Length Integers*)
MV – motion vector
I – frame coded without motion compensation
P_1 – frame coded differentially (predictively) in respect to frame I, with *forward* motion compensation
P_2 – frame coded differentially (predictively) in respect to frame P_1, with *forward* motion compensation
B – frames coded differentially with *forward* (from I) and *backward* (from P) motion compensation



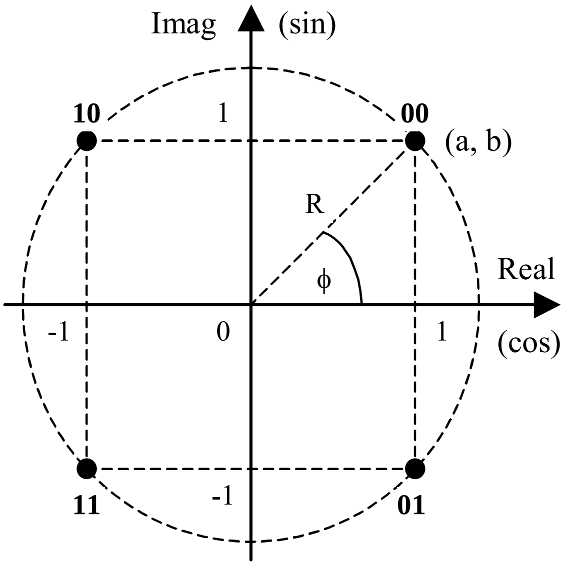
a) b)

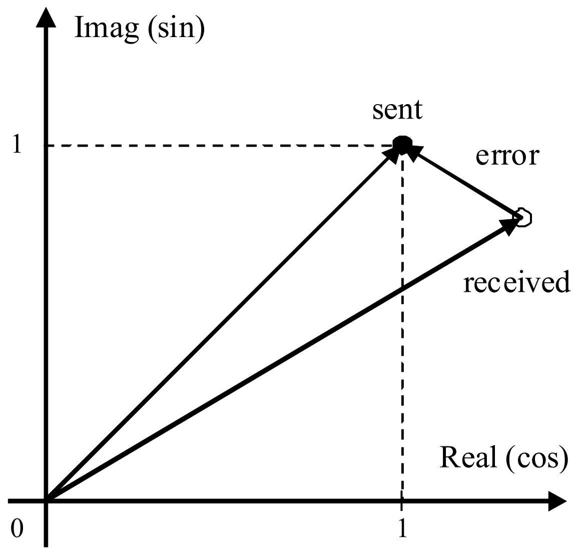


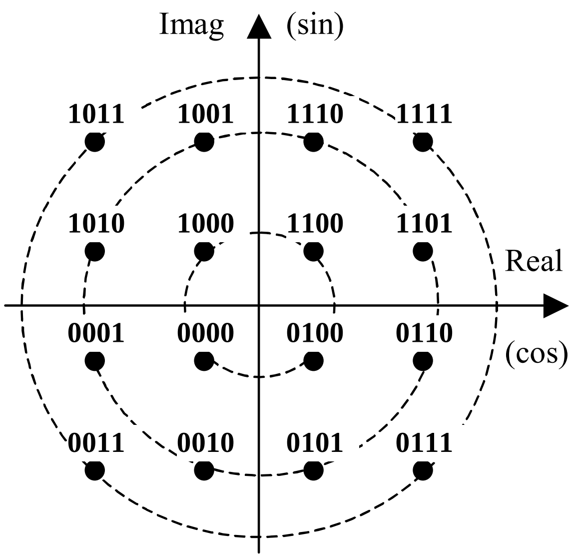
17.3. ADSL modem

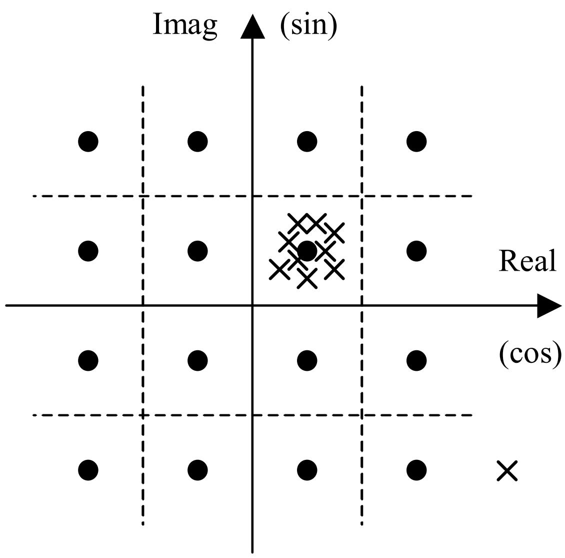
Quadrature amplitude modulation of each frequency carrier:

$$x_m(t) = A\cos\left(2\pi f t + \phi_m\right) = a_m \cos\left(2\pi f t\right) + b_m \sin\left(2\pi f t\right)$$

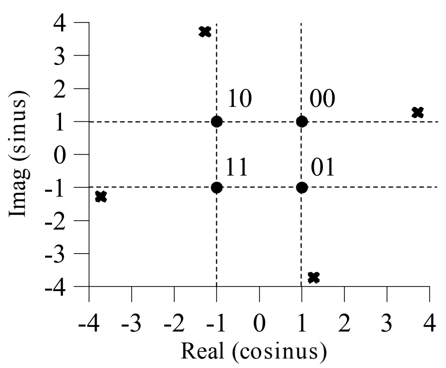


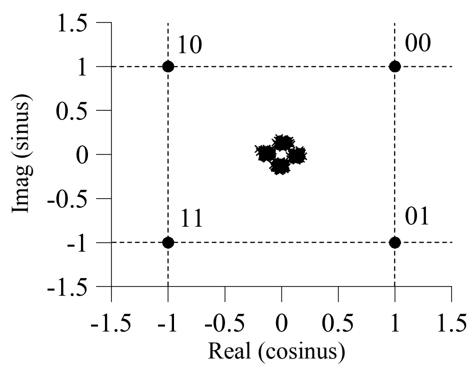






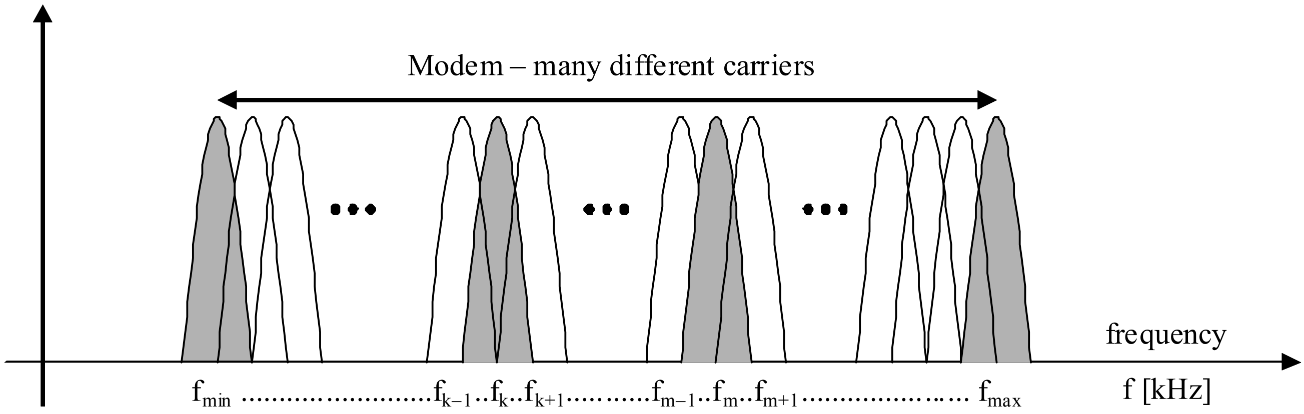
Channel distortions: scaling (attenuation) and rotation (phase shift due to time delay) of the signal constellation state plus noise



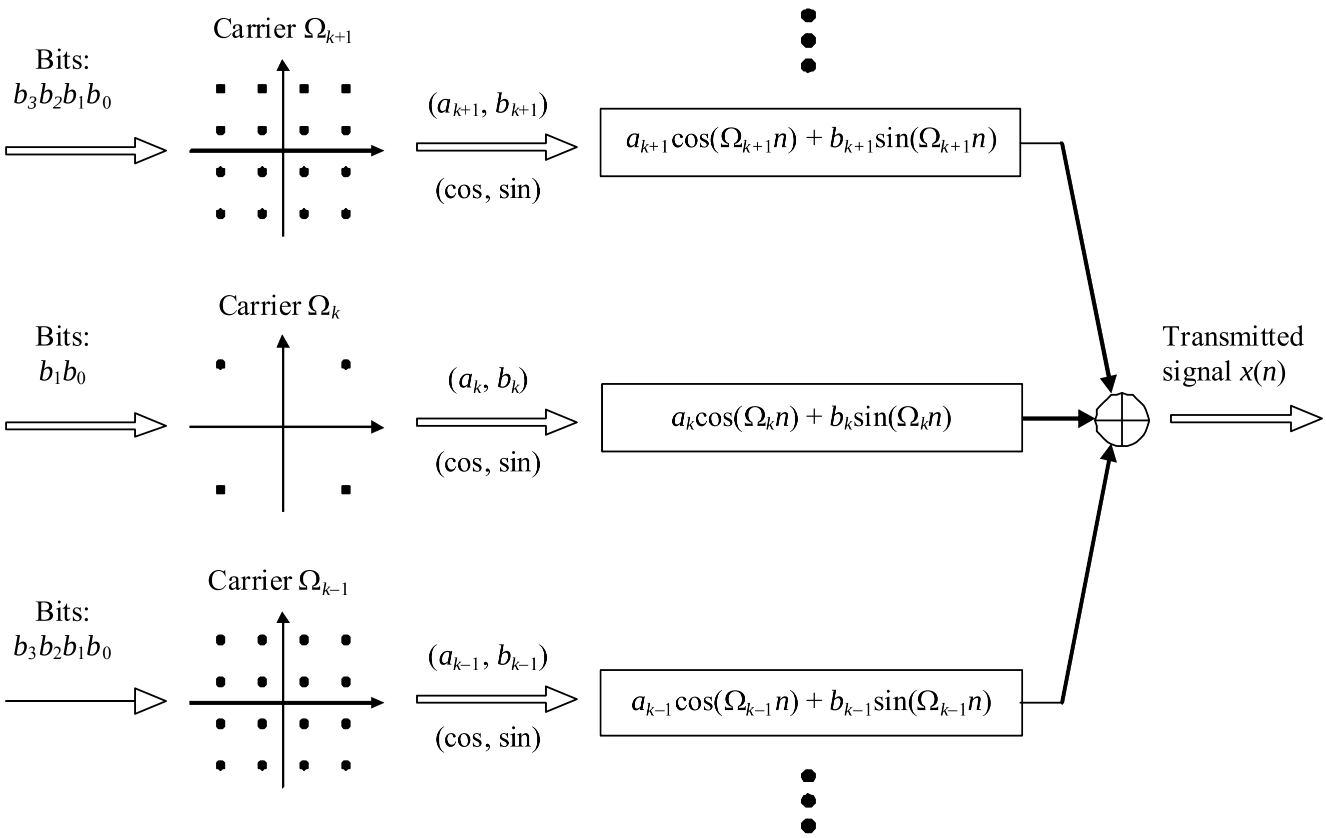


Basic principle of multicarrier wide-band modem:

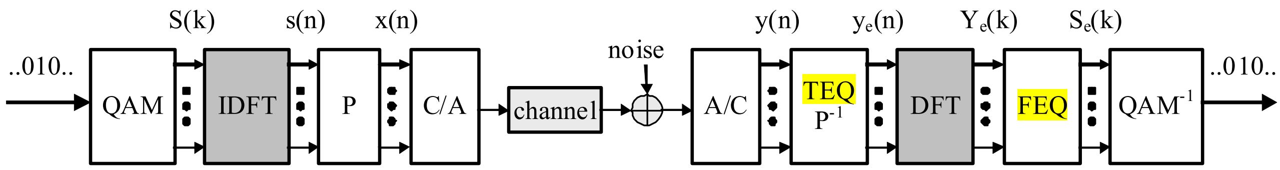
1) division of frequency band between many carriers that transmit data the same time (in parallel)

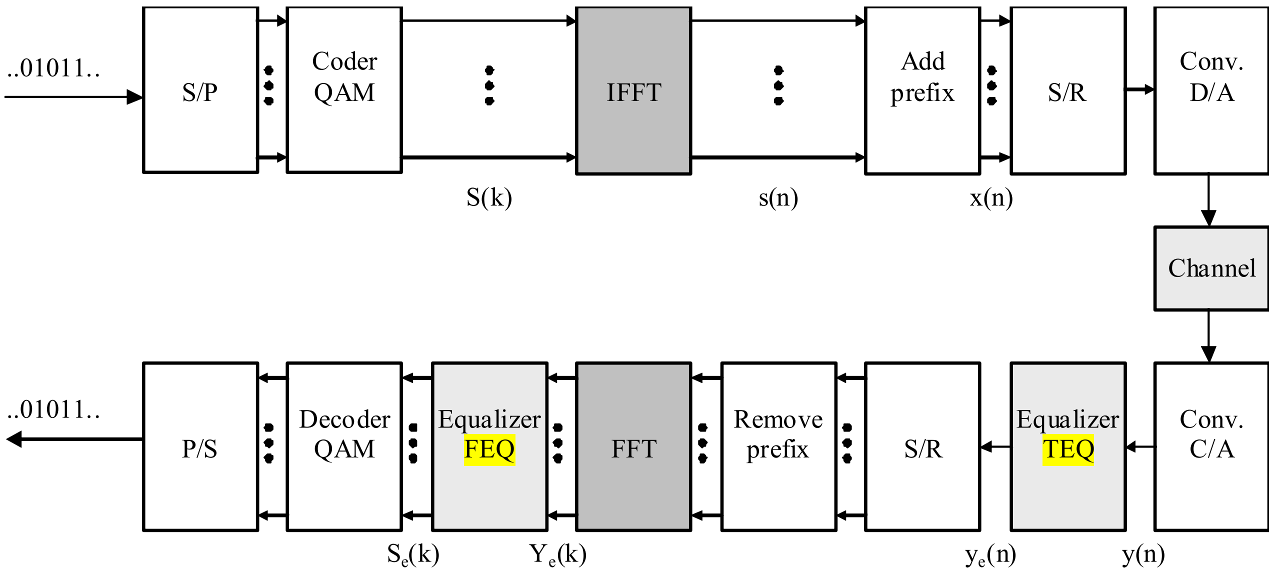


2) generation of the transmitted signal being a sum of many modulated carriers

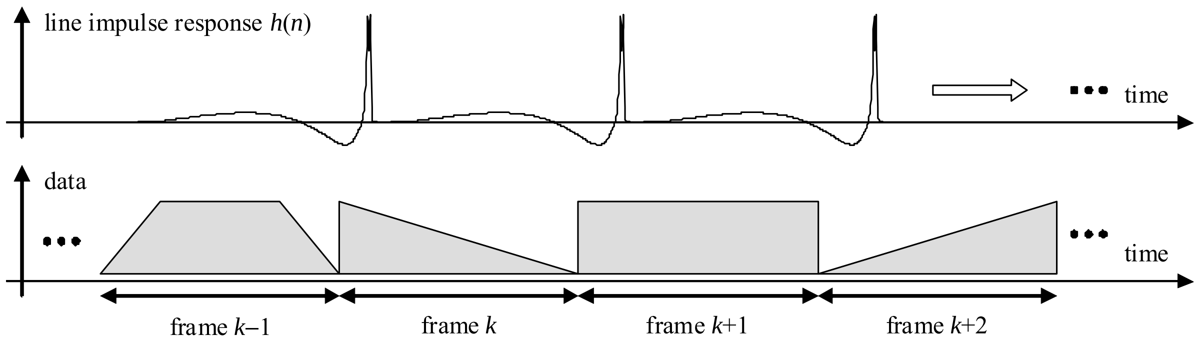


Simplified diagram of multi-carrier transmission scheme. Denotations: QAM i QAM^{-1} – signal state (constellation) encoder and decoder, DFT and IDFT – direct and inverse discrete Fourier transform, P and P^{-1} – addition and subtraction of the cyclic prefix (repetition of signal fragment), D/A and A/D – digital-to-analog and analog-to-digital converters, TEQ – time equalizer shortening the channel impulse response, FEQ – frequency equalizer, removing amplitude and phase disturbances introduced by the transmission line (channel)

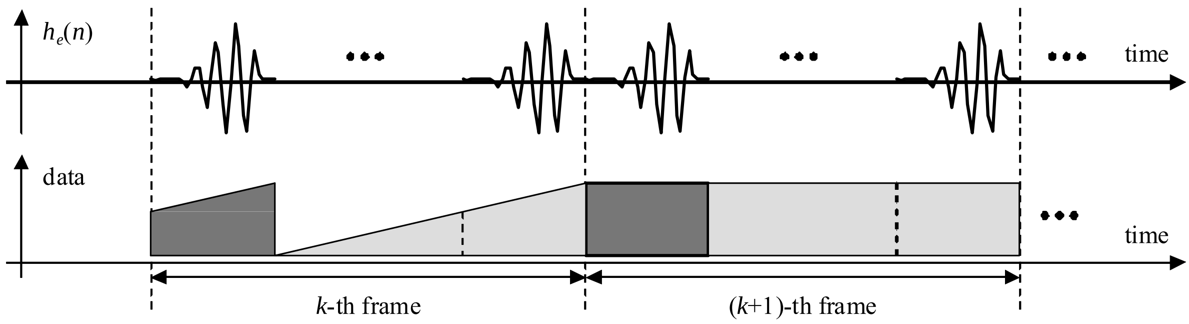




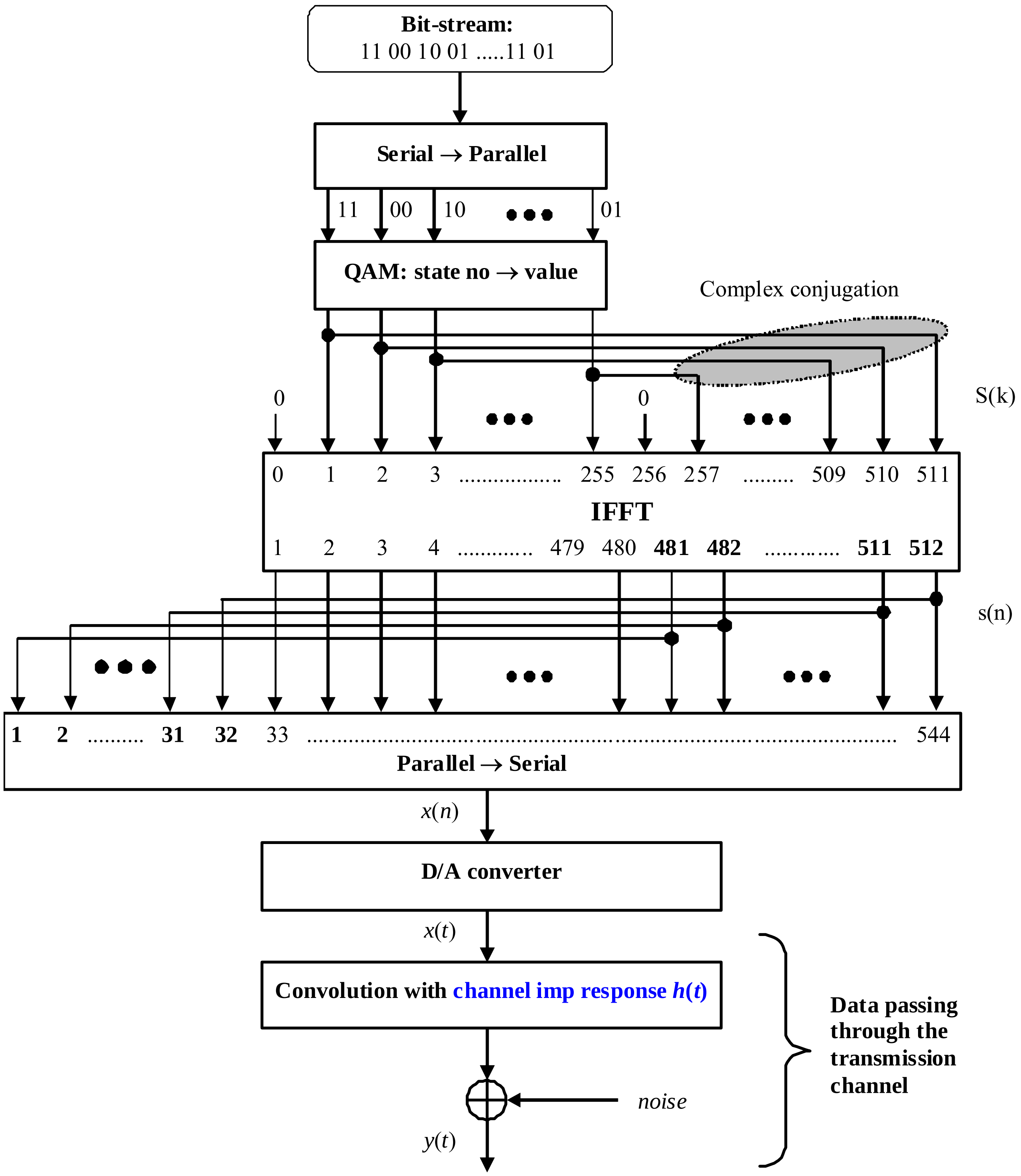
Block-based transmission:



Cyclic prefix addition (reduction of cross-symbol interference)



Block diagram of the ADSL transmitter and data passing through the channel



Block diagram of the ADSL receiver