AVC-162

PROPOSAL PACKAGE

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IED JTC1/SC2/WG11
CODING OF MOVING PICTURES AND ASSOCIATED VIDEO

ISO/JEC JTEC1/SC2/WG11
MPEG91/226
NOV 1991

SOURCE : SONY (REG. #33)

TITLE : Proposal of encoding algorithm of MPEG II

PURPOSE : Proposal

Version

Nov. 7, 1991

1. Introduction

This document is the summary of the proposed coding algorithm for WG11/MPEG November meeting.

2. General Outline

The implemented coding configuration is based on the MPEG1 Video CD with improvements. Main improvements are:

(1) Adaptive frame/field MB structure
(2) Adaptive frame/field prediction
(3) Coding(Combination of DCT and Non Transform Codig)

3. Source Input Format(S.I.F)

The parameters for S.I.F. are:

Table1 Source Input Format

	Standard 525
Number of active lines	
Luminance (Y)	480
Chrominance (Cb, Cr)	480
Number of active pixels per line	
Luminance (Y)	720
Chrominance (Cb, Cr)	360
Frame Rate (Hz)	30
Frame aspect ratio (hor:ver)	4:3
Sampling Freq. ratio (Y:Cb:Cr)	4:2:2

4. Layering

4.1. GOP

The definition of a GOP (Group of pictures) is the same as in the MPEG1 Video CD. Figure1 shows the structure of GOP.

4.2. Picture

The three types of Pictures are the same as in the MPEG1 Video CD, I_picture(Intra coded picture), P_picture(Predictive coded picture) and B_picture(Bidirectionally predictive coded picture). Each type of picture consists of two fields, odd and even fields.

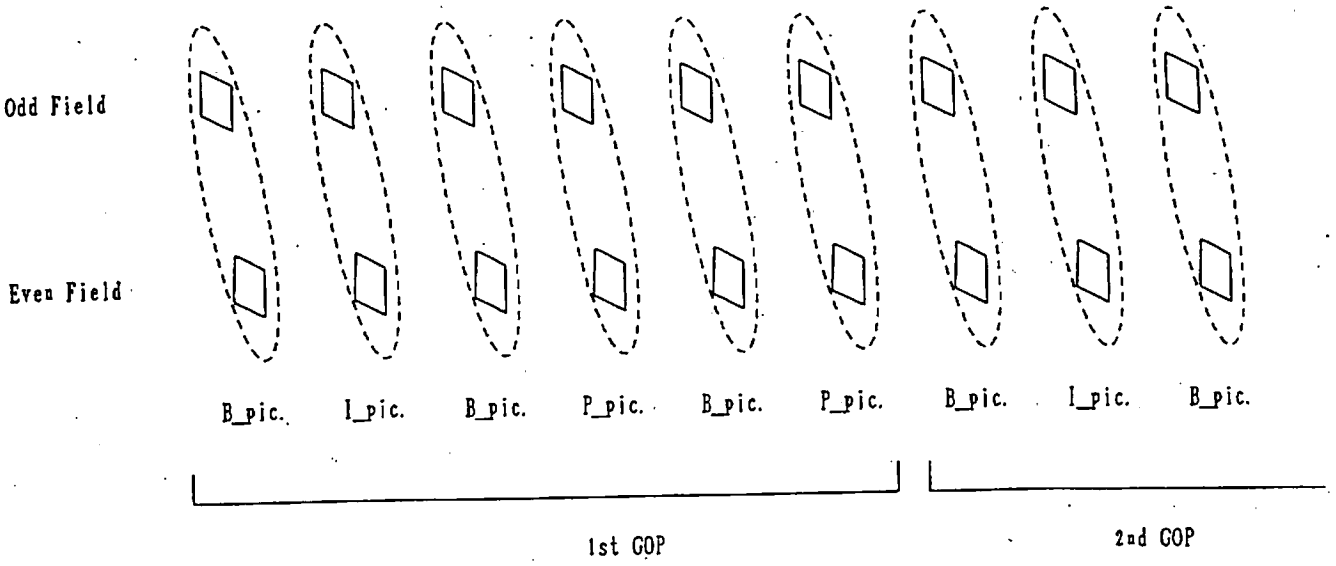


Figure1 Structure of GOP and Pictures

4.3. Slice

The definition of a Slice is the same as in the MPEG1 Video CD.

4.4. Macro Block

Three types of Macro Block are introduced, Frame_MB,Odd_field_MB and Even_field_MB. Figure2,3 and 4 shows the structure of each MB type.

(1) Frame_MB

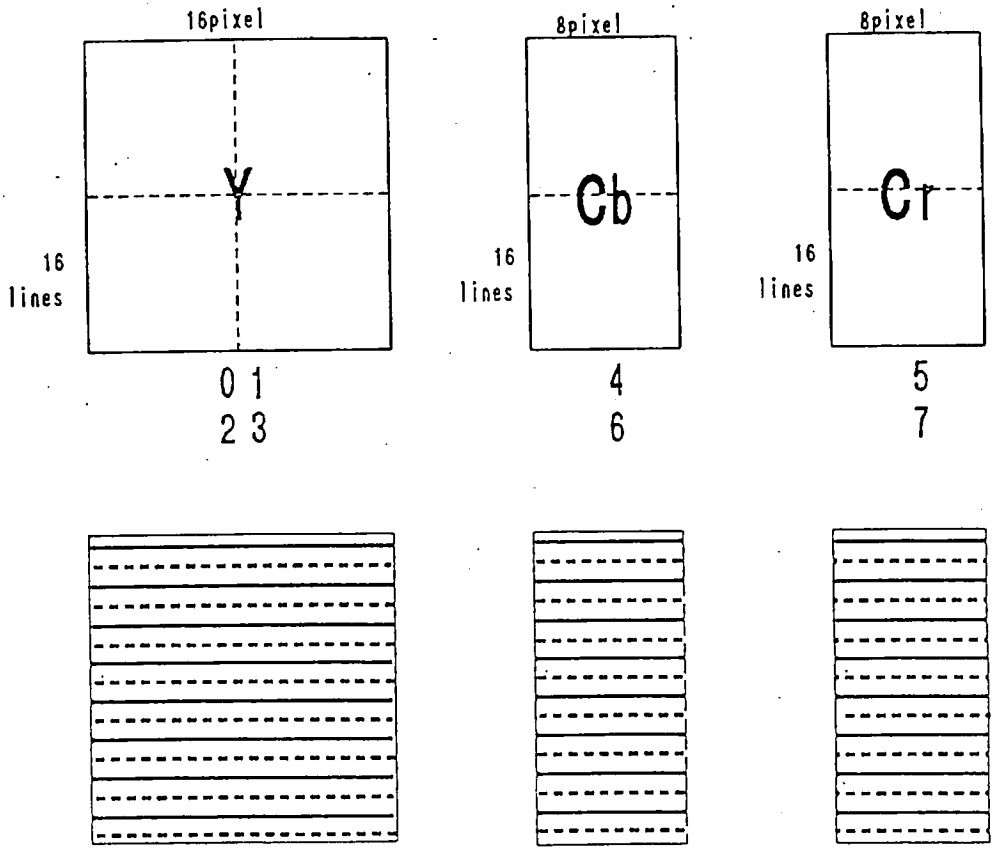


Figure2 Structure of Frame_MB

(2) Odd_field_MB

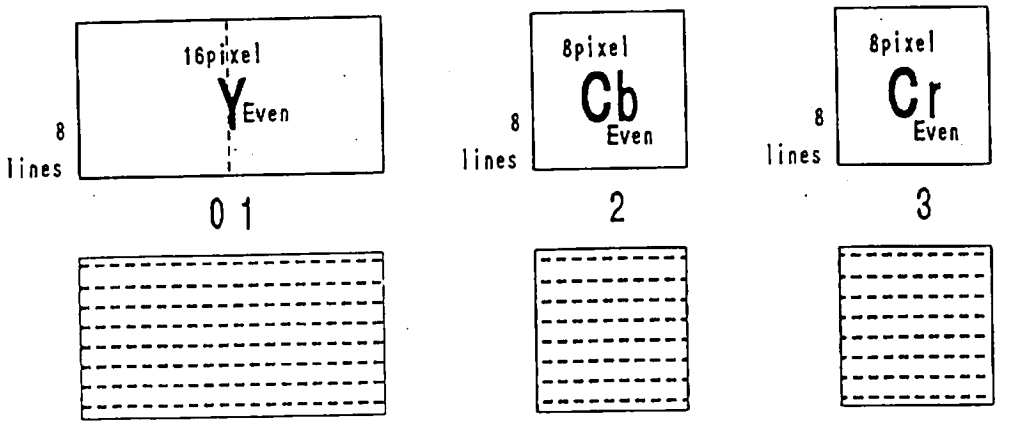


Figure4 Structure of Even_field_MB

(3) Even_field_MB

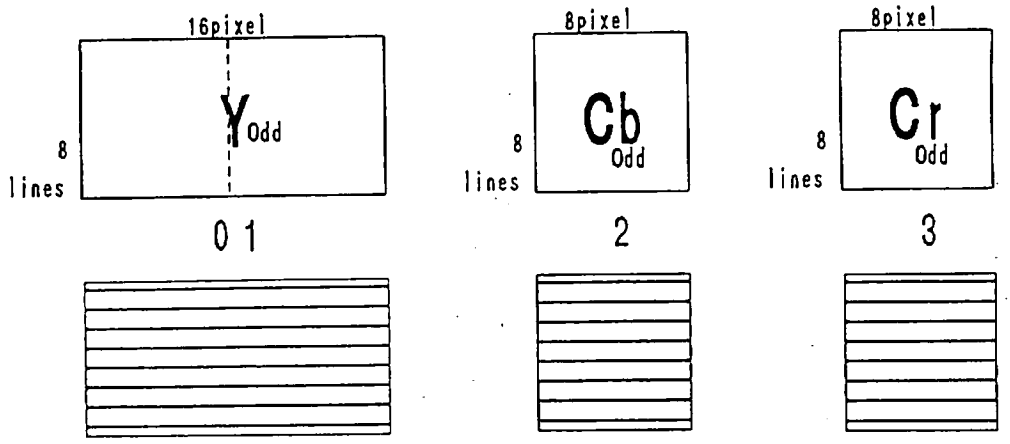


Figure3 Structure of Odd_field_MB

4.5. Block

The definition of a Block is the same as in the MPEG1 Video CD.

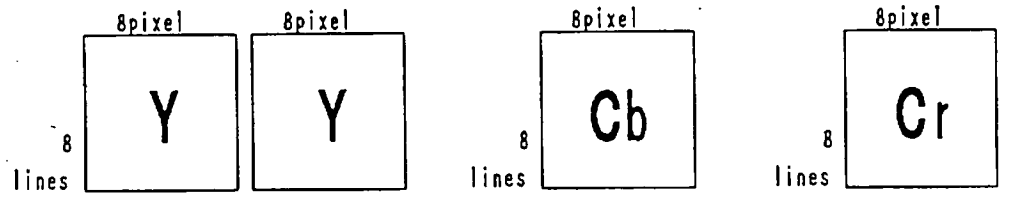
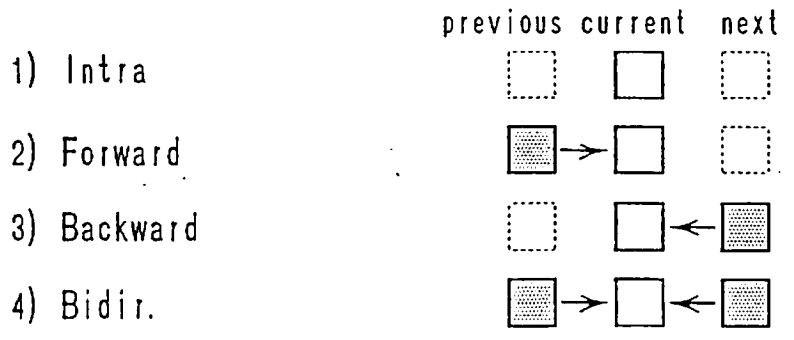


Figure5 Structure of Block

5. Adaptive frame/field prediction

The definition of Prediction_Mode is the same as in the MPEG1 Video CD. Four types of Prediction_Mode are:



5.1. Function of each prediction mode

Figure 6 shows function of prediction. Prediction mode is selected by MB_mode and picture_coding_type.

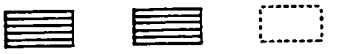
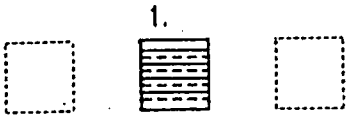
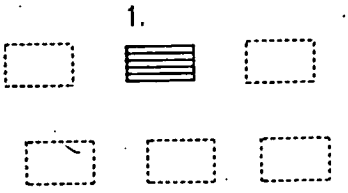
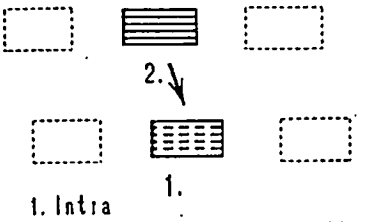
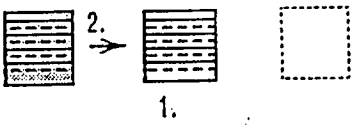
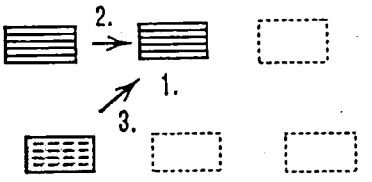
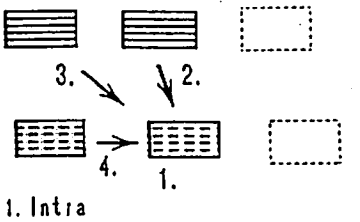
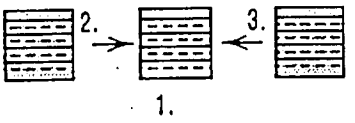
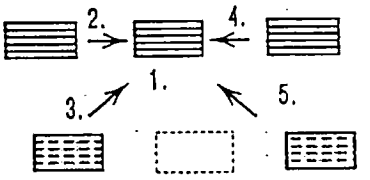
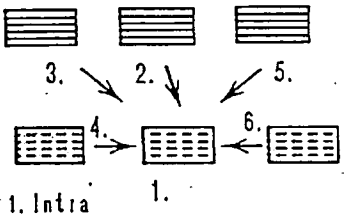
	Frame_MB	Odd_field_MB	Even_field_MB
	 Current Reference don't care	 Current Reference don't care	 Current Reference don't care
I-Pict.	 1. Intra	 1. Intra	 1. Intra 2. Forward_from_current_odd
P-Pict.	 1. Intra 2. Forward	 1. Intra 2. Forward_from_odd 3. Forward_from_even	 1. Intra 2. Forward_from_current_odd 3. Forward_from_odd 4. Forward_from_even
B-Pict.	 1. Intra 2. Forward 3. Backward 4. Bidir. (2+3)	 1. Intra 2. Forward_from_odd 3. Forward_from_even 4. Backward_from_odd 5. Backward_from_even 6. Bidir. (2+4) 7. Bidir. (2+5) 8. Bidir. (3+4) 9. Bidir. (3+5)	 1. Intra 2. Forward_from_current_odd 3. Forward_from_odd 4. Forward_from_even 5. Backward_from_odd 6. Backward_from_even 7. Bidir. (3+5) 8. Bidir. (3+6) 9. Bidir. (4+5) 10. Bidir. (4+6) 11. Bidir. (2+5) 12. Bidir. (2+6)

Figure6 function of prediction

5.2. Sequence of transmission

Macro Blocks are transmitted in two fields, odd field and even field.

- Frame_MBs with 8 blocks are transmitted in odd field.
- Frame_MBs with header only are transmitted in even field just for MB address increment.
- Odd_field_MBs with 4 blocks are transmitted in odd field.
- Even_field_MBs with 4 blocks are transmitted in even field.

The local decoded image of the odd field is available at next even field, So some Even_field_MBs are predicted from current odd field.

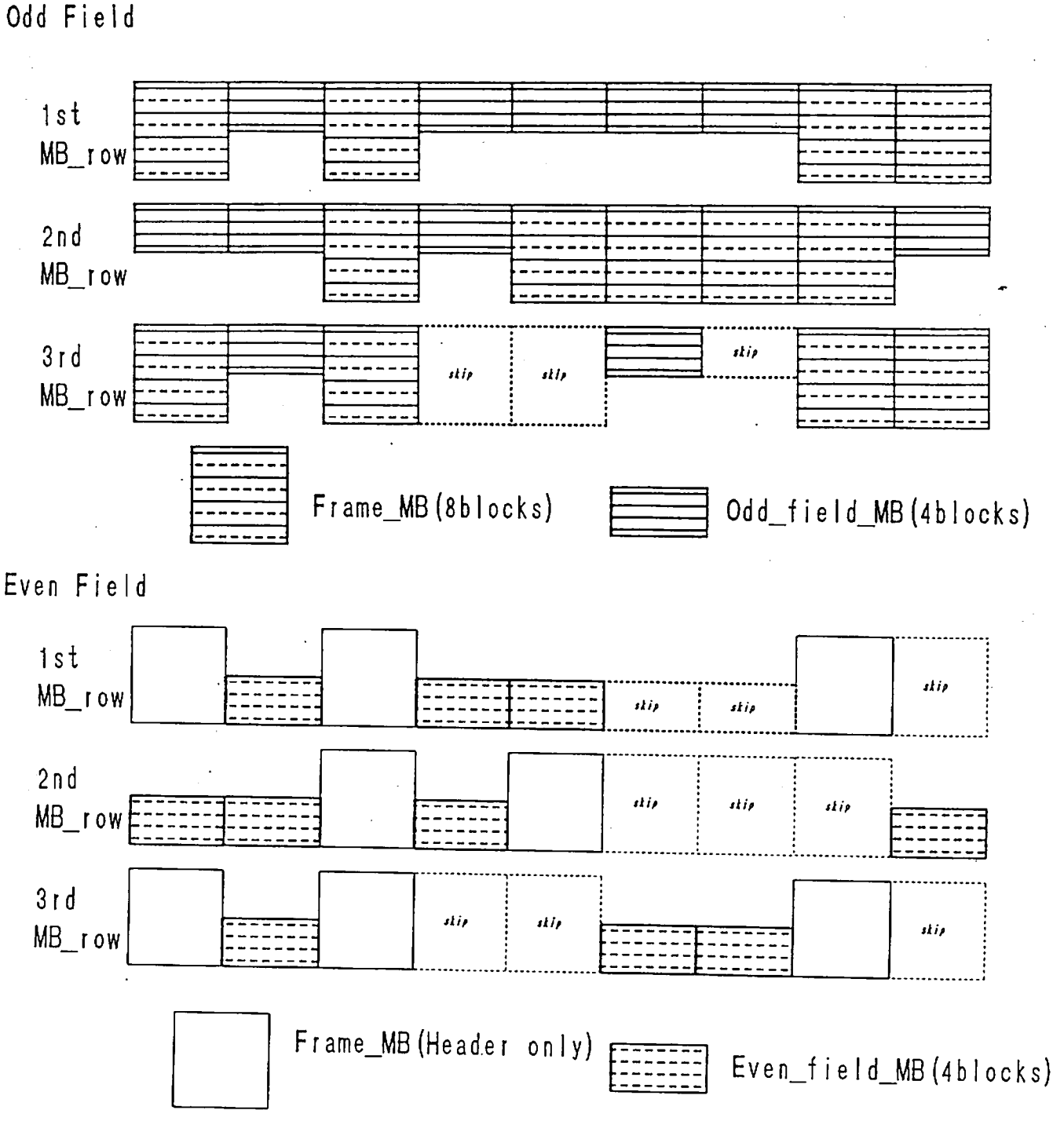


Figure7 sequence of transmit

6. Coding

The structure of the proposed algorithm is basically the same as that of the MPEG1 Video CD. The block diagrams of the encoder and decoder are shown in Figure 8_a and 8_b. The proposed algorithm features a combination of DCT and NTC(Non Transform Coding), i.e. either DCT or NTC is applied to 8x8 block.

6.1. DCT

In case of DCT, the coding algorithm is the same as in the MPEG1 Video CD. In case of DCT, the block diagrams of the encoder and the decoder are shown in Figure 9_a and 9_b.

6.2. NTC(Non Transform Coding)

In case of NTC, the block diagrams of the encoder and decoder are shown in Figure 10_a and 10_b.

Remark: NTC is derived from "Adaptive Dynamic Range Coding" reference [1].

6.2.1. Algorithm of NTC

Figure11_a and 11_b show examples of NTC. And Figure12 shows NTC block diagram. NTC consists of 4steps as follows.

(1) BASE value calculation

BASE value is a mean of the pixels in flat area, and transmitted in 8/9bit without quantization. Then BASE value is subtracted from all pixels before quantization.

(2) Stepsize setting and quantization

In intra MBs the stepsize is set in order to minimize the discontinuity at the block boundaries. In NON intra MBs the stepsize is set in order to keep original dynamic range. In case of NTC, blocks are quantized with a flat weighting matrix and with a dead zone equal to the stepsize, the quantizer is the same as used in DCT for non intra macro blocks in the MPEG1 Video CD.

(3) Scan mode selection

NTC has four scan modes.(see Figure13) NTC selects one scan mode which gives the fewest number of RUN LEVEL events.

(4) DPCM ON/OFF and VLC

NTC decides DPCM ON/OFF which gives fewer VLC events. If DPCM is ON, the scanned data are 1_dimensional DPCMed and then VLCed. Else if DPCM is OFF, the scanned data are directly VLCed. Figure14 shows the DPCM and iDPCM block diagrams.

6.2.2. Conditions for NTC

The blocks which satisfy the following conditions are coded in NTC.(see Figure12)

- blocks with edges and flat area.
- blocks with small low frequency DCT coefficients and large high frequency DCT coefficients.

These conditions are checked using pixel activity parameters and DCT activity parameters. The pixel activity parameters are the minimum,the maximum, the dynamic range and the mean of pixels. The DCT activity parameters are the mean and the center of energy of AC coefficients.

6.2.3. Data elements related to NTC

The following parameters are calculated in NTC. Table2 shows the data elements related to NTC.

(1)BASE
(2)stepsize
(3)Data scan mode
(4)DPCM on/off flag

Table2 Data elements related to NTC.

Parameter	Contents	Number of bits
DCT/NTC_and_DPCM_on /NTC_and_DPCM_off	indicated by block type	VLC
BASE flag	if BASE=0, BASE flag=0, else, BASE flag=1.	1
BASE	if BASE flag==1 if Intra MB, 1<=BASE<=255. if Inter MB, -255<=BASE<=255, except 0.	8 9
Stepsize for NTC	1<=Stepsize<=127	7
DATA SCAN MODE	No. 1~4 (see Figure13)	2

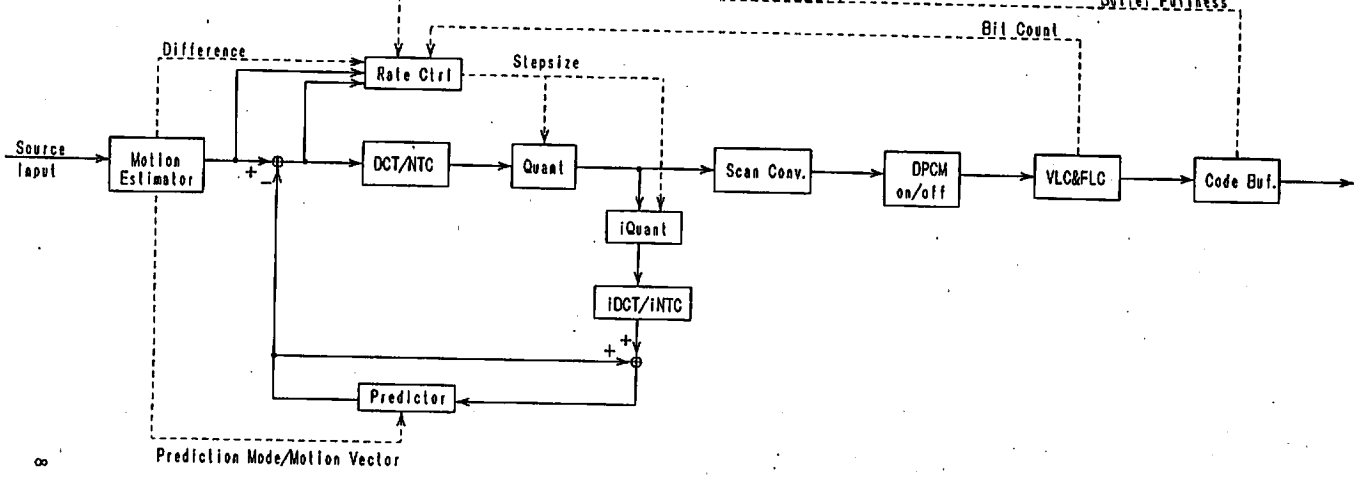


Figure 8-a Encoder Block Diagram

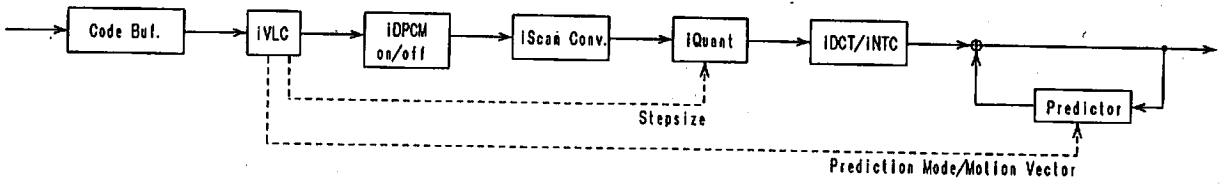


Figure 8-b Decoder Block Diagram

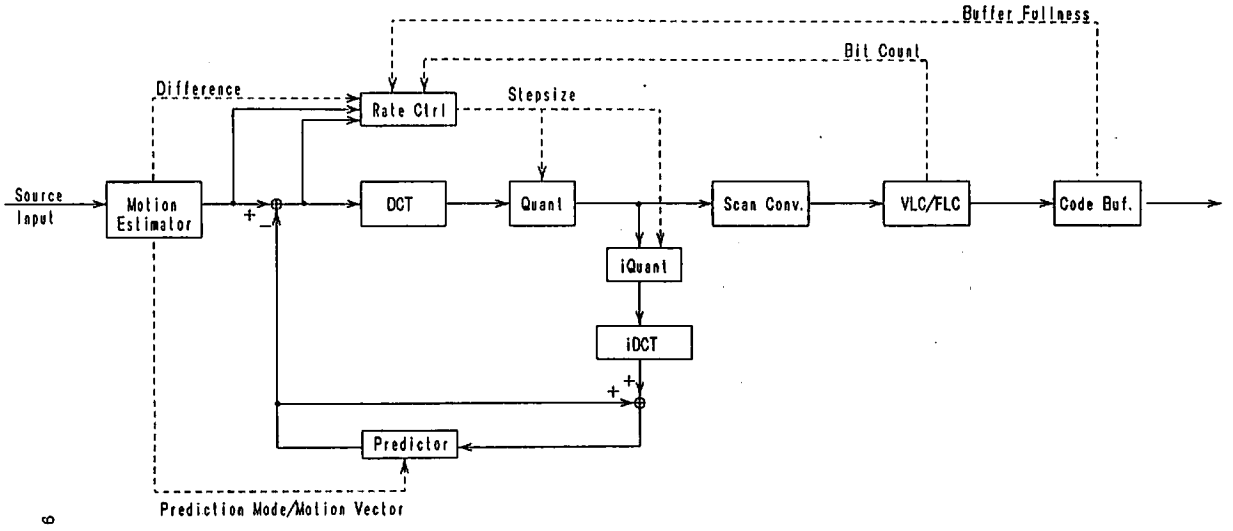


Figure 9-a DCT Encoder Block Diagram

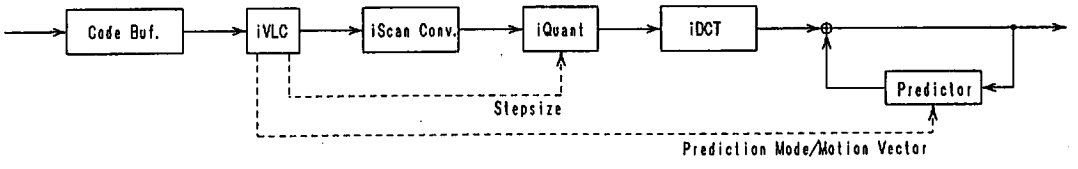


Figure 9-b DCT Decoder Block Diagram

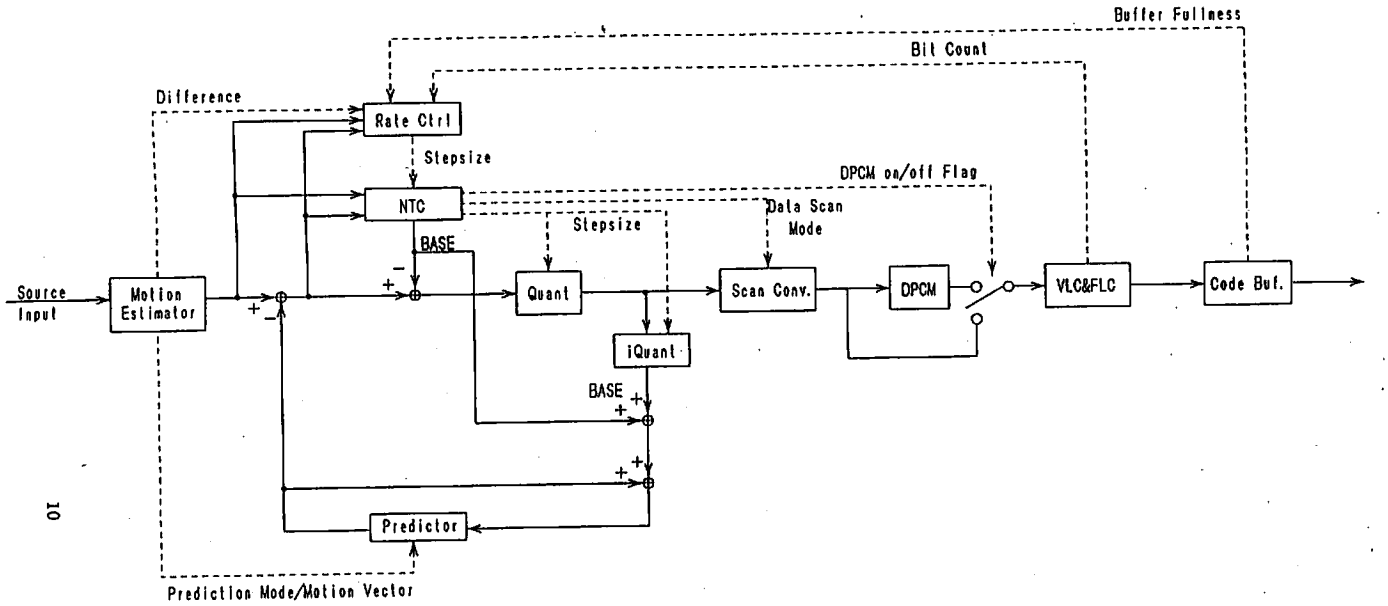


Figure 10-a NTC Encoder Block Diagram

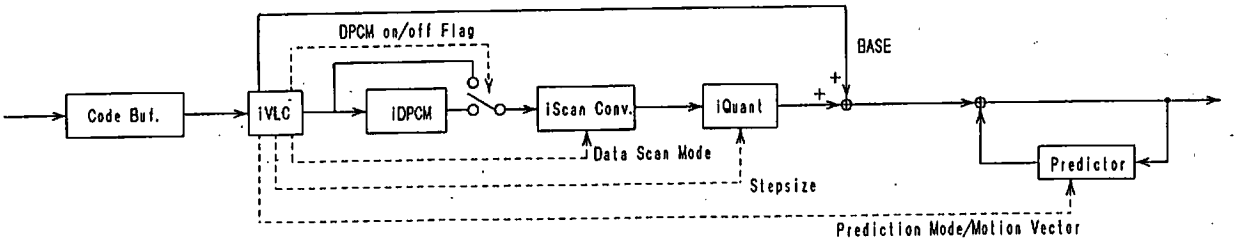


Figure 10-b NTC Decoder Block Diagram

(example. 1) a example of NTC block selecting 'DPCM OFF'.
<case of NTC>

8x8 block pixel data

199	197	198	198	199	199	197	198
198	196	200	196	195	198	197	198
196	197	196	196	197	198	196	197
105	200	197	198	197	198	194	195
203	196	200	196	199	196	198	199
137	219	198	199	199	196	198	197
198	199	196	198	199	196	199	198
197	201	197	199	198	198	198	199

Subtract BASE from each pixels.
BASE=198

1	-1	0	0	1	1	-1	0
0	-2	2	-2	-3	0	-1	0
-2	-1	-2	-2	-1	0	-2	-1
-93	2	-1	0	-1	0	-4	-3
5	-2	2	-2	1	-2	0	1
-61	21	0	1	1	-2	0	-1
0	1	-2	0	1	-2	1	0
-1	3	-1	1	0	0	0	1

QUANTIZED stepsize=6

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-7	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-5	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

DATA SCAN MODE:No. 4 and DPCM OFF.

0	0	0	-7	0	-5	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Add BASE to each pixels.
Reconstructed pixel block.

198	198	198	198	198	198	198	198
198	198	198	198	198	198	198	198
198	198	198	198	198	198	198	198
109	198	198	198	198	198	198	198
198	198	198	198	198	198	198	198
133	215	198	198	198	198	198	198
198	198	198	198	198	198	198	198
198	198	198	198	198	198	198	198

iVLC
iQUANTIZED

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-89	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-65	17	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

2-dimensional VLC
(3, -7), (1, -5), (4, 1), EOB

<Reference; case of DCT> Coding scheme depends on MPEG1 video CD.

8x8 block pixel data

199	197	198	198	199	199	197	198
198	196	200	196	195	198	197	198
196	197	196	196	197	198	196	197
105	200	197	198	197	198	194	195
203	196	200	196	199	196	198	199
137	219	198	199	199	196	198	197
198	199	196	198	199	196	199	198
197	201	197	199	198	198	198	199

DCT coef.

1565	-21	-23	-23	-21	-20	-14	-7
-2	1	2	5	4	6	9	2
20	23	24	22	20	17	11	5
5	3	0	-2	-3	-4	-5	0
-4	-7	-6	-3	0	0	2	1
-13	-16	-18	-12	-11	-10	-8	-5
-1	-2	-5	-5	-7	-9	-8	-5
23	29	33	31	30	24	18	12

QUANTIZED stepsize=6
with a default weighting matrix

196	-2	-2	-1	-1	-1	-1	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-1	-1	-1	-1	0	0	0	0
0	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0

ZZ SCANNING

196	-2	0	1	0	-2	-1	0
1	0	0	0	1	0	-1	-1
0	1	0	0	-1	0	-1	0
0	1	0	-1	0	0	1	0
0	-1	0	1	1	0	-1	0
0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	0

Reconstructed pixel block

196	191	192	206	199	192	203	196
188	211	197	190	203	202	200	195
194	194	193	204	200	194	203	195
114	204	209	187	205	198	187	199
205	204	193	199	201	199	204	194
149	209	203	187	204	201	193	197
188	187	193	208	199	190	203	196
203	203	193	198	201	198	204	195

iVLC
iQUANTIZED

1568	-23	-27	-15	-19	-19	-21	0
0	0	0	0	0	0	0	0
13	15	19	19	21	25	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-19	-19	-21	-23	0	0	0	0
0	0	0	0	0	0	0	0
19	21	25	27	33	41	0	0

2-dimensional VLC
(*, 196), (0, -2), (1, 1), (1, -2), (0, -1), (1, 1), (3, 1), (1, -1), (0, -1), (1, 1), (2, -1), (1, -1), (2, 1), (1, -1), (2, 1), (2, -1), (1, 1), (0, 1), (1, -1), (9, 1), (0, 1), (7, 1), (0, 1), EOB

Figure11_a Example of NTC

(example. 2) a example of NTC block selecting "DPCM ON".

<case of NTC>

8x8 block pixel data

199	197	198	198	199	199	197	198
198	196	200	196	195	198	197	198
196	101	196	196	197	198	196	197
105	106	197	198	197	198	194	195
98	104	200	196	199	196	198	199
100	98	198	199	199	196	198	198
254	254	196	198	199	196	199	198
254	253	197	199	198	198	198	199

Subtract BASE from each pixels.
BASE=198

1	-1	0	0	1	1	-1	0
0	-2	2	-2	-3	0	-1	0
-2	-97	-2	-2	-1	0	-2	-1
-93	-92	-1	0	-1	0	-4	-3
-100	-94	2	-2	1	-2	0	1
-98	-100	0	1	1	-2	0	0
56	56	-2	0	1	-2	1	0
56	55	-1	1	0	0	0	1

QUANTIZED stepsize=6

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	-8	0	0	0	0	0	0
-7	-7	0	0	0	0	0	0
-8	-7	0	0	0	0	0	0
-8	-8	0	0	0	0	0	0
4	4	0	0	0	0	0	0
4	4	0	0	0	0	0	0

DATA SCAN MODE:NO. 4 and DPCM ON.

0	0	0	-7	-1	0	12	0
0	0	-12	1	0	-1	8	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

2-dimensional VLC
(3, -7), (0, -1), (1, 12), (3, -12), (0, 1), (1, -1), (0, 8), EOB

IVLC
IQUANTIZED

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	-101	0	0	0	0	0	0
-89	-89	0	0	0	0	0	0
-101	-89	0	0	0	0	0	0
-101	-101	0	0	0	0	0	0
53	53	0	0	0	0	0	0
53	53	0	0	0	0	0	0

Add BASE to each pixels.
Reconstructed pixel block.

198	198	198	198	198	198	198	198
198	198	198	198	198	198	198	198
198	97	198	198	198	198	198	198
109	109	198	198	198	198	198	198
97	109	198	198	198	198	198	198
97	97	198	198	198	198	198	198
251	251	198	198	198	198	198	198
251	251	198	198	198	198	198	198

<Reference; case of DCT> Coding scheme depends on MPEG1 video CD.

8x8 block pixel data

199	197	198	198	199	199	197	198
198	196	200	196	195	198	197	198
196	101	196	196	197	198	196	197
105	106	197	198	197	198	194	195
98	104	200	196	199	196	198	199
100	98	198	199	199	196	198	198
254	254	196	198	199	196	199	198
254	253	197	199	198	198	198	199

DCT coef.

1525	-70	-48	-17	13	29	33	21
-27	-31	-20	-4	8	18	22	11
110	135	98	45	-5	-37	-46	-31
-28	-41	-35	-27	-16	-9	-5	2
-9	-14	-16	-14	-12	-12	-7	-5
12	18	12	11	5	0	-3	-3
-33	-41	-25	-2	17	27	27	16
21	26	25	19	13	7	3	3

QUANTIZED stepsize=6
with a default weighting matrix

191	-6	-3	-1	1	1	2	1
-2	-3	-1	0	0	1	1	0
8	8	5	2	0	-1	-2	-1
-2	-2	-2	-1	-1	0	0	0
-1	-1	-1	-1	-1	0	0	0
1	1	1	0	0	0	0	0
-2	-2	-1	0	1	1	1	0
1	1	1	1	0	0	0	0

ZZ SCANNING

191	-6	-2	8	-3	-3	-1	-1
8	-2	-1	-2	5	0	1	1
0	2	-2	-1	1	-2	1	-1
-1	0	1	2	1	1	-1	-1
-1	1	-2	1	1	-1	0	-1
0	-2	0	-1	0	0	0	0
1	1	1	0	0	0	0	0
1	0	0	1	0	0	0	0

2-dimensional VLC
(*, 191), (0, -6), (0, -2), (0, 8), (0, -3), (0, -3), (0, -1), (0, -1), (0, 8), (0, -2), (0, -1), (0, -2), (0, 5), (1, 1), 0, 1), (1, 2), (0, -2), (0, -1), (0, 1), (0, -2), (0, 1), (0, -1), (0, -1), (1, 1), (0, 2), (0, 1), (0, 1), (0, -1), (0, -1), (0, -1), (0, 1), (0, -2), (0, 1), (0, 1), (0, -1), (1, -1), (1, -2), (1, -1), (4, 1), (0, 1), (0, 1), (0, 1), (5, 1), (2, 1), EOB

IVLC
IQUANTIZED

1528	-71	-41	-15	19	19	43	25
-23	-35	-15	0	0	21	25	0
113	131	97	39	0	-25	-51	-27
-33	-33	-39	-19	-21	0	0	0
-15	-19	-19	-21	-23	0	0	0
19	19	21	0	0	0	0	0
-39	-39	-21	0	27	33	41	0
19	21	25	27	0	0	0	0

Reconstructed pixel block

203	190	204	199	186	206	202	199
201	200	195	196	206	198	199	201
200	108	187	191	195	208	187	210
105	106	208	193	196	195	192	206
93	104	198	184	201	205	195	189
114	92	194	202	199	196	181	198
254	255	202	190	211	189	203	203
253	242	199	202	202	189	196	206

Figure11_b Example of NTC

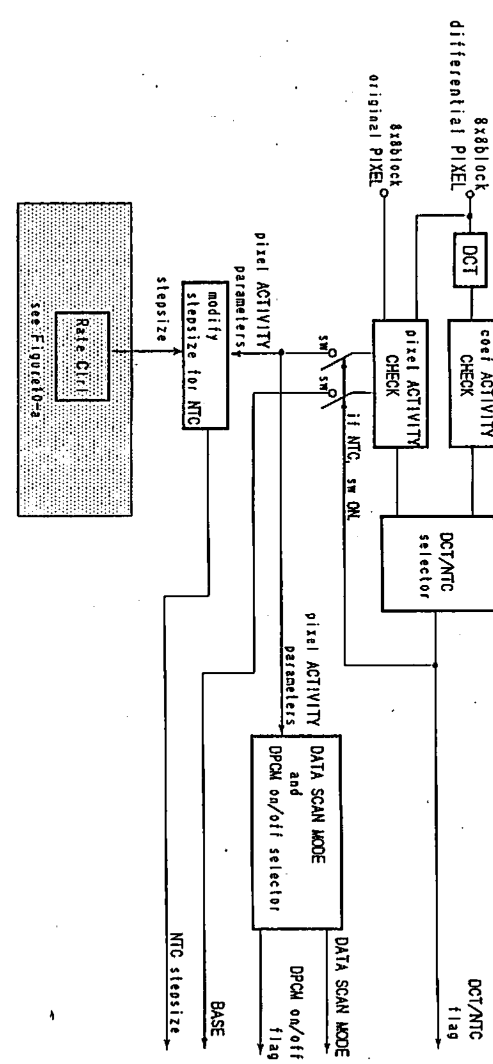
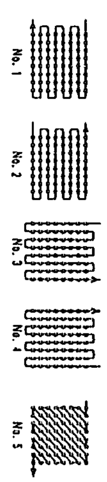


Figure12 NTC block diagram



In case of NTC, DATA SCAN MODE is selected from No. 1 to No. 4.
In case of DCT, No. 5 is used.

Figure13 DATA SCAN MODE

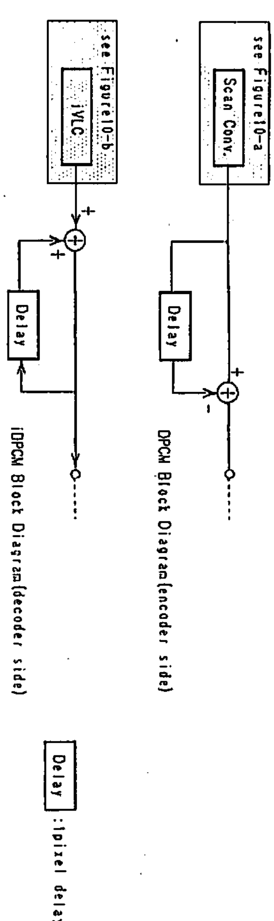


Figure14 DPCM and iDPCM Block Diagrams

7. Video Multiplex Coder

The video multiplex is arranged in a hierarchical structure with six layers. From top to bottom the layers are:

Sequence Layer
Group of Pictures Layer
Picture Layer
Field Layer
Slice Layer
Macroblock Layer
Block Layer

7.1. Sequence Layer

same as MPEG1 video CD.

7.2. Group of Pictures layer

same as MPEG1 video CD.

7.3. Picture Layer

same as MPEG1 video CD.

7.4. Field Layer

Field_StartCode --The field start code is the bit string '0000 0000 0000 0000 0000 1011 0000'.

7.5. Slice layer

same as MPEG1 video CD.

7.6. Macroblock Layer

macroblock_stuffing --same as MPEG1 video CD

macroblock_escape --same as MPEG1 video CD

macroblock_address_increment --same as MPEG1 video CD

macroblock_type --Variable length codes for the following information are shown in tables 3 through 5.

field/frame macroblock

macroblock_quant

macroblock_motion_forward

macroblock_motion_backward

macroblock_pattern

macroblock_intra

quantizer_scale --same as MPEG1 video CD

motion_horizontal_forward --same as MPEG1 video CD

motion_vertical_forward --same as MPEG1 video CD

motion_horizontal_backward --same as MPEG1 video CD

motion_vertical_backward --same as MPEG1 video CD

coded_block_pattern --same as MPEG1 video CD.

7.7. Block Layer

block_type --Variable length codes for block type are shown in the tables 6.

NTC_parameters --In case of NTC_and_DPCM_on and NTC_and_DPCM_off, the following NTC_parameters are coded.

ntc_quantize_scale--refer Stepsize for NTC in Table2.

ntc_quantize_base_flag--refer BASE flag in Table2.

ntc_quantize_base--refer BASE in Table2.

ntc_scan_conversion--refer DATA SCAN MODE in Table2.

block_data --In case of DCT, the block_data is the same as MPEG1 video CD. In case of NTC_and_DPCM_on and NTC_and_DPCM_off, the block_data is coded accoding to the Table 8a through 8c.

8. Bit Stream Syntax

```
/********************** Program **********************/
program(){

    startcode(Program_Startcode);

    do {
        sequence();
    }while (nextbits()==Sequence_StartCode);

    startcode(Program_End_Code);
}
/********************** Sequence Layer **********************/
sequence(){

    int i;

    startcode(Sequence_StartCode);

    horizontal_size (12);
    vertical_size   (12);
    pel_aspect_ratio( 4);
    picture_rate    ( 4);
    bit_rate        (18);
    marker_bit      ( 1,"1");
    buffer_size     (10);
    constrained_parameter_flag(1);

    while (nextbits()=="1"){
        load_quantizer_matrix_(1,"1");
        quantizer_matrix_id(3);
        for (i=0;i<64;i++){
            Nth_quantizer_matrix(8);
        }
    }
    load_quantizer_matrix_(1,"0");
    if (nextbits()==Extension_StartCode){
        startcode(Extension_StartCode);
        while (nextbits()!=OneOfStartCodes){
            user_data(8*N,EXTENSION_DATA);
        }
    }
    if (nextbits()==UserData_StartCode){
        startcode(UserData_StartCode);
        while (nextbits()!=OneOfStartCodes){
            user_data(8*N,USER_DATA);
        }
    }
    do {
        group_of_pictures();
    }while (nextbits()==Group_StartCode);
}
/************** Group of Pictures Layer **************/
group_of_pictures(){

    startcode(Group_StartCode);

    timecode_drop_frame_flag   (1);
    timecode_time_code_hours   (5);
    timecode_time_code_minutes (6);
    marker_bit(1,"1");
    timecode_time_code_seconds (6);
    timecode_time_code_pictures(6);
    closed_gop(1);
    broken_link(1);

    if (nextbits()==Extension_StartCode){
        startcode(Extension_StartCode);
        while (nextbits()!=OneOfStartCodes){
            user_data(8*N,EXTENSION_DATA);
        }
    }
    if (nextbits()==UserData_StartCode){
        startcode(UserData_StartCode);
        while (nextbits()!=OneOfStartCodes){
            user_data(8*N,USER_DATA);
        }
    }
    do {
        picture();
    }while (nextbits()==Picture_StartCode);
}
/****************** Picture Layer ******************/
picture(){

    startcode(Picture_StartCode);

    temporal_reference(10);
    picture_coding_type(3);
    buffer_fullness(16);

    if ((picture_coding_type==P_Picture)||
        (picture_coding_type==B_Picture)){
        full_pel_forward_vector(1);
        forward_f_code(3);
    }
    if (picture_coding_type==B_Picture){
        full_pel_backward_vector(1);
        backward_f_code(3);
    }
    if (nextbits()==Extension_StartCode){
        startcode(Extension_StartCode);
        while (nextbits()!=OneOfStartCodes){
            user_data(8*N,EXTENSION_DATA);
        }
    }
    if (nextbits()==UserData_StartCode){
        startcode(UserData_StartCode);
        while (nextbits()!=OneOfStartCodes){
            user_data(8*N,USER_DATA);
        }
    }

    field(First);
    field(Second);
}
/****************** Field Layer ******************/
field(field_cycle_ID){

    startcode(Field_StartCode);

    do {
        slice();
    }while(nextbits()==Slice_StartCode);
}
/****************** Slice Layer ******************/
slice(){

    startcode(Slice_StartCode);

    quantizer_scale_for_dct(5);

    do {
        macroblock();
    }while(nextbits()!=OneOf_StartCodes);
}
/**************** Macroblock Layer ********************/
macroblock(field_cycle_identifier){

    while(nextbits()==macroblock_stuffing)
        macroblock_address_increment(VLC,Macroblock_Stuffing);

    while(nextbits()==macroblock_escape)
        macroblock_address_increment(VLC,Macroblock_Escape);

    macroblock_address_increment(VLC);
    switch(picture_coding_type){
      case I_Picture:
        if (field_cycle_ID==First){
            macroblock_type_I_picture(VLC);
        }else /* Second Field */ {
            macroblock_type_P_picture(VLC);
        }
        break;
      case P_Picture:
        macroblock_type_P_picture(VLC);
        break;
      case B_Picture:
        macroblock_type_B_picture(VLC);
        break;
        }
    }
    if (macroblock_type&Macroblock_Quant)
        macroblock_quant(5);

    if (macroblock_type&Macroblock_Motion_Forward){
        if (macroblock_type&Field_Mode){
            if ((picture_coding_type==B_Picture)||
                (field_cycle_ID==First))
                forward_reference_field_for_frame_mode_picture(1);
            else
                forward_reference_field_for_field_mode_picture(VLC)
;
        }

        motion_horizontal_forward_code(VLC);
        if ((forward_f!=1)&&(motion_horizontal_forward_code!=0))
            motion_horizontal_forward_r(FLC);

        motion_vertical_forward_code(VLC);
```



```
        if ((forward_f!=1)&&(motion_vertical_forward_code!=0))
            motion_vertical_forward_r(FLC);
    }
    if (macroblock_type&Macroblock_Motion_Backward){
        if (macroblock_type&Field_Mode)
            backward_reference_field(1);

        motion_horizontal_backward_code(VLC);
        if ((backward_f!=1)&&(motion_horizontal_backward_code!=0))
            motion_horizontal_backward_r(FLC);

        motion_vertical_backward_code(VLC);
        if ((backward_f!=1)&&(motion_vertical_backward_code!=0))
            motion_vertical_backward_r(FLC);
    }
    if (macroblock_type&Macroblock_Pattern){
        block_num=4;
        for (block_no=0;block_no<block_num;block_no++){
            block(macroblock_type,block_no);
        }
    }
}
/****************** Block Layer ******************/
block(macroblock_type,block_no){
    block_type(VLC);
    if (block_type&DCT){
        if (isMacroblock_Intra(macroblock_type)){

            if (isY(block_no)){
                dct_dc_size_y(VLC);
                if (dct_dc_size_y!=0){
                    dct_dc_diff_y(FLC);
                }
            }else{
                dct_dc_size_c(VLC);
                if (dct_dc_size_c!=0){
                    dct_dc_diff_c(FLC);
                }
            }

            while(nextbits()!=End_Of_Block){
                dct_coef_intra(VLC);
            }
            dct_coef_intra(VLC,End_Of_Block);
        }else /* inter */{

            while(nextbits()!=End_Of_Block){
                dct_coef_inter(VLC);
            }
            dct_coef_inter(VLC,End_Of_Block)
        }
    }else /* block_type&NTC */{

        ntc_quantize_scale(7);
        ntc_quantize_base_flag(1);
        if (ntc_quantize_base_flag==1){
          if (isMacroblock_Intra(macroblock_type)){
            ntc_quantize_base(8);
          }
          else{
            ntc_quantize_base(9);
          }
        ntc_scan_conversion(2);

        while(nextbits()!=End_Of_block){
            ntc_pixel(VLC);
        }
        ntc_pixel(VLC,End_Of_Block);
    }
}
/********************** block_data **********************/
coef_or_pixel()
{
    if (nextbits()==Escape){

        escape(VLC);
        run(6);
        level(8);

        if      (nextbits()=="00000000"){
            escape2(8,"00000000");
        }
        else if (nextbits()=="10000000"){
            escape2(8,"10000000");
        }

        level(8);

    }else{
        bi_dimentional_vlc(VLC);
        sign(1);
    }
}
/********************** startcode **********************/
startcode(ParamStartCode)
{
    while(!(32_bits_aligned())){
        zero_bit(1,"0");
    }
    start_code_header(24,"0000 0000 0000 0000 0000 0001");
    start_code_body  ( 8,ParamStartCode);
}
```

9. Variable Length Coding Tables

9.1. Macroblock Addressing

same as MPEG1 video CD.

9.2. Macroblock Type

Table 3. Variable length code for macroblock_type in I-picture

VLC code	field:0 frame:1	macroblock_ quant
"00"	0	0
"01"	0	1
"10"	1	0
"11"	1	1

Table 4. Variable length code for macroblock_type in P-picture

VLC code	field:0 frame:1	macroblock_ quant	macroblock_ motion forward	macroblock_ pattern	intra
"000"	0	0	1	1	0
"0010"	0	0	0	1	0
"0011"	0	0	1	0	0
"0100"	0	0	0	0	1
"0101"	0	1	1	1	0
"0110"	0	1	0	1	0
"0111"	0	1	0	0	1
"1000"	1	0	1	1	0
"1001"	1	0	0	1	0
"1010"	1	0	1	0	0
"1011"	1	0	0	0	1
"1100"	1	1	1	1	0
"1101"	1	1	0	1	0
"1110"	1	1	0	0	1
"1111"	1	0	0	0	0

Table 5. Variable length code for macroblock_type in B-picture

VLC code	field:0 frame:1	macroblock_ quant	macroblock_ motion forward	macroblock_ motion backward	macroblock_ pattern	intra
"0000"	0	0	1	1	0	0
"0001"	0	0	1	1	1	0
"0010"	0	0	0	1	0	0
"0011"	0	0	0	1	1	0
"0100"	0	0	1	0	0	0
"0101"	0	0	1	0	1	0
"0110"	0	0	0	0	0	1
"0111"	0	1	1	1	1	0
"1000"	0	1	0	1	1	0
"10010"	0	1	1	0	1	0
"10011"	0	1	0	0	1	0
"10100"	1	0	1	1	1	0
"10101"	1	0	1	1	0	0
"10110"	1	0	0	1	1	0
"10111"	1	0	0	0	1	0
"11000"	1	0	1	1	0	0
"11001"	1	0	1	0	1	0
"11010"	1	0	0	1	0	0
"11011"	1	1	1	1	1	0
"11100"	1	1	0	1	1	0
"11101"	1	1	1	0	1	0
"11110"	1	1	0	0	0	1
"11111"	1	0	0	0	0	0

9.3. Block type

Table 6. Variable length codes for block_type

VLC code	coding mode
"00"	DCT
"10"	NTC and DPCM ON
"11"	NTC and DPCM OFF

9.4. Macroblock Pattern

same as MPEG1 video CD.

9.5. Motion Vectors

same as MPEG1 video CD.

9.6. DCT coeficients

Variable length codes for dct_dc_size_luminance and dct_dc_size_chrominance is same as MPEG1 video CD.

There are two variable length code tables for dct_coefficients. One is table for intra_coded_macroblock, it is shown in Table 7a through 7d. And the other is table for non_intra_coded_macroblock, it is same as MPEG video CD. And method of encoding of run and level following ESC(escape code) is same as MPEG video CD.

Table7a. Variable length codes for dct_coeff in intra_coded_macroblock.

dct_coeff variable length code (NOTE)	run	level
"00" s	0	1
"010" s	0	2
"011" s	1	1
"1000" s	EOB	
"1001" s	0	3
"10100" s	ESC	
"10101" s	0	4
"10110" s	1	2
"10111" s	2	1
"11000" s	3	1
"110010" s	0	5
"110011" s	0	6
"110100" s	4	1
"110101" s	5	1
"110110" s	6	1
"110111" s	7	1
"1110000" s	0	7
"1110001" s	0	8
"1110010" s	1	3
"1110011" s	8	1
"1110100" s	9	1
"1110101" s	10	1
"1110110" s	11	1
"11101110" s	0	9
"11101111" s	0	10
"11110000" s	1	4
"11110001" s	2	2
"11110010" s	12	1
"11110011" s	13	1
"111101000" s	0	11
"111101001" s	0	12
"111101010" s	0	13
"111101011" s	1	5
"111101100" s	2	3
"111101101" s	3	2
"111101110" s	4	2
"111101111" s	5	2
"111110000" s	14	1
"1111100010" s	0	14
"1111100011" s	0	15
"1111100100" s	0	16
"1111100101" s	0	17
"1111100110" s	1	6
"1111100111" s	6	2
"1111101000" s	7	2
"1111101001" s	8	2
"1111101010" s	9	2
"1111101011" s	10	2
"1111101100" s	15	1
"11111011010" s	0	18
"11111011011" s	0	19
"11111011100" s	0	20
"11111011101" s	0	21
"11111011110" s	1	7
"11111011111" s	1	8
"11111100000" s	2	4
"11111100001" s	3	3
"11111100010" s	4	3
"11111100011" s	11	2
"11111100100" s	12	2
"11111100101" s	13	2
"11111100110" s	16	1
"11111100111" s	17	1
"11111101000" s	20	1
"11111101001" s	25	1
"11111101010" s	32	1
"111111010110" s	0	22
"111111010111" s	0	23
"111111011000" s	0	24
"111111011001" s	0	25
"111111011010" s	1	9
"111111011011" s	1	10
"111111011100" s	2	5
"111111011101" s	3	4
"111111011110" s	5	3
"111111011111" s	8	3
"111111100000" s	9	3
"111111100001" s	14	2
"111111100010" s	18	1
"111111100011" s	19	1
"111111100100" s	21	1

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

Table7b. Variable length codes for dct_coeff in intra_coded_macroblock(continued).

dct_coeff variable length code (NOTE)	run	level
"111111100101" s	22	1
"111111100110" s	23	1
"111111100111" s	24	1
"111111101000" s	26	1
"111111101001" s	29	1
"111111101010" s	30	1
"111111010110" s	0	26
"111111010111" s	0	27
"111111011000" s	0	28
"111111011001" s	0	29
"111111011010" s	0	30
"111111011011" s	1	11
"111111011100" s	1	12
"111111011101" s	2	6
"111111011110" s	3	5
"111111011111" s	4	4
"111111100000" s	6	3
"111111100001" s	7	3
"111111100010" s	10	3
"111111100011" s	11	3
"111111100100" s	12	3
"111111100101" s	13	3
"111111100110" s	27	1
"111111100111" s	28	1
"111111101000" s	31	1
"111111010010" s	0	31
"111111010011" s	0	32
"111111010100" s	0	33
"111111010101" s	0	34
"111111010110" s	0	35
"111111010111" s	1	13
"111111011000" s	1	14
"111111011001" s	1	15
"111111011010" s	2	7
"111111011011" s	2	8
"111111011100" s	3	6
"111111011101" s	5	4
"111111011110" s	5	5
"111111011111" s	7	4
"111111100000" s	8	4

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

dct_coeff variable length code (NOTE)	run	level
"111111110000" s	9	4
"111111110001" s	12	4
"111111110010" s	13	4
"111111110011" s	14	3
"111111110100" s	33	1
"111111110101" s	34	1
"111111110110" s	0	36
"111111110111" s	0	37
"111111111000" s	0	38
"111111111001" s	1	16
"111111111010" s	1	17
"111111111011" s	1	18
"111111111100" s	1	19
"111111111101" s	1	20
"111111111110" s	3	7
"111111111111" s	3	8
"111111110110" s	4	5
"111111110111" s	4	6
"111111111010" s	6	4
"111111111011" s	7	5
"111111111100" s	9	5
"111111111101" s	10	4
"111111111110" s	10	5
"111111111111" s	11	4
"111111110000" s	12	5
"111111110001" s	13	5
"111111110010" s	35	1
"111111111000110" s	0	39
"111111111000111" s	0	40
"111111111001000" s	1	26
"111111111001001" s	2	9
"111111111001010" s	2	11
"111111111001011" s	2	12
"111111111001100" s	3	9
"111111111001101" s	3	10
"111111111001110" s	4	7
"111111111001111" s	5	6
"111111111010000" s	5	7
"111111111010001" s	5	9
"111111111010010" s	5	11

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

Table7c Variable length codes for dct_coeff in intra_coded_macroblock(continued).

dct_coeff variable length code (NOTE)	run	level
"111111111010011" s	6	5
"111111111010100" s	9	6
"111111111010101" s	12	6
"111111111010110" s	13	6
"111111111010111" s	13	7
"111111111011000" s	14	4
"111111111011001" s	15	2
"111111111011010" s	16	2
"111111111011011" s	20	2
"111111111011100" s	21	2
"111111111011101" s	25	2
"111111111011110" s	36	1
"111111111010111110" s	1	21
"111111111011111" s	1	22
"111111111000000" s	1	23
"111111111000001" s	1	24
"111111111000010" s	1	25
"111111111000011" s	1	27
"111111111000100" s	1	28
"111111111000101" s	2	10
"111111111000110" s	2	13
"111111111000111" s	2	14
"111111111001000" s	3	11
"111111111001001" s	3	12
"111111111001010" s	3	16
"111111111001011" s	4	8
"111111111001100" s	4	10
"111111111001101" s	5	8
"111111111001110" s	5	14
"111111111001111" s	6	6
"111111111010000" s	7	6
"111111111010001" s	8	5
"111111111010010" s	8	6
"111111111010011" s	11	5
"111111111010100" s	11	6
"111111111010101" s	12	7
"111111111010110" s	12	8
"111111111010111" s	13	8
"111111111011000" s	14	5
"111111111011001" s	14	6

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

dct_coeff variable length code (NOTE)	run	level
"111111111011010" s	17	2
"111111111011011" s	18	2
"111111111011100" s	23	2
"111111111011101" s	24	2
"111111111011110" s	26	2
"111111111011111" s	29	2
"111111111100000" s	30	2
"111111111100001" s	37	1
"111111111100010" s	38	1
"111111111100011" s	39	1
"111111111111001000" s	1	31
"111111111111001001" s	1	33
"111111111111001010" s	1	34
"111111111111001011" s	1	36
"111111111111001100" s	1	37
"111111111111001101" s	1	38
"111111111111001110" s	3	13
"111111111111001111" s	3	14
"111111111111010000" s	3	15
"111111111111010001" s	3	17
"111111111111010010" s	3	20
"111111111111010011" s	4	9
"111111111111010100" s	4	11
"111111111111010101" s	4	12
"111111111111010110" s	4	14
"111111111111010111" s	4	15
"111111111111011000" s	5	10
"111111111111011001" s	7	8
"111111111111011010" s	7	19
"111111111111011011" s	8	7
"111111111111011100" s	8	8
"111111111111011101" s	8	9
"111111111111011110" s	9	7
"111111111111011111" s	9	8
"111111111111100000" s	10	6
"111111111111100001" s	10	7
"111111111111100010" s	11	8
"111111111111100011" s	12	9
"111111111111100100" s	12	10
"111111111111100101" s	13	9

NOTE- The last bit 's' denotes the sign o the level, '0' for positive, '1' for negati

Table7d. Variable length codes for dct_coeff in intra_coded_macroblock(continued).

dct_coeff variable length code (NOTE)	run	level
"1111111111111100110"s	13	10
"1111111111111100111"s	22	2
"1111111111111101000"s	24	3
"1111111111111101001"s	27	2
"1111111111111101010"s	28	2
"1111111111111101011"s	32	2
"1111111111111011000"s	1	30
"1111111111111011001"s	2	15
"1111111111111011010"s	3	19
"1111111111111011011"s	3	23
"1111111111111011100"s	3	24
"1111111111111011101"s	4	16
"1111111111111011110"s	5	12
"1111111111111011111"s	6	7
"1111111111111100000"s	7	7
"1111111111111100001"s	7	11
"1111111111111100010"s	11	7
"1111111111111100011"s	11	11
"1111111111111100100"s	12	11
"1111111111111100101"s	14	7
"1111111111111100110"s	19	2
"1111111111111100111"s	29	3
"1111111111111101000"s	33	2
"1111111111111010010"s	1	29
"1111111111111010011"s	1	32
"1111111111111010100"s	1	40
"1111111111111010101"s	2	16
"1111111111111010110"s	2	18
"1111111111111010111"s	2	23
"1111111111111011000"s	2	33
"1111111111111011001"s	3	18
"1111111111111011010"s	3	27
"1111111111111011011"s	5	13
"1111111111111011100"s	5	15
"1111111111111011101"s	5	18
"1111111111111011110"s	5	22
"1111111111111011111"s	6	8
"1111111111111100000"s	6	9
"1111111111111100001"s	6	10
"1111111111111100010"s	6	14

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

dct_coeff variable length code (NOTE)	run	level
"1111111111111100011"s	7	9
"1111111111111100100"s	7	10
"1111111111111100101"s	7	16
"1111111111111100110"s	7	29
"1111111111111100111"s	8	10
"1111111111111101000"s	9	9
"1111111111111101001"s	9	11
"1111111111111101010"s	10	8
"1111111111111101011"s	11	9
"1111111111111101100"s	12	13
"1111111111111101101"s	13	11
"1111111111111101110"s	13	12
"1111111111111101111"s	14	8
"1111111111111110000"s	14	11
"1111111111111110001"s	15	4
"1111111111111110010"s	16	3
"1111111111111110011"s	17	3
"1111111111111110100"s	22	3
"1111111111111110101"s	23	3
"1111111111111110110"s	23	4
"1111111111111110111"s	24	4
"1111111111111111000"s	25	3
"1111111111111111001"s	26	3
"1111111111111111010"s	26	4
"1111111111111111011"s	30	3
"1111111111111111100"s	31	2
"1111111111111111101"s	32	4
"1111111111111111110"s	34	2
"1111111111111111111"s	35	2

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

9.7. NTC coefficients

Table8a. Variable length codes for NTC_coeff.

NTC coeff variable length code (NOTE)	run	level
"00"s	EOB	
"010"s	0	1
"0110"s	0	2
"0111"s	1	1
"1000"s	2	1
"10010"s	3	1
"10011"s	4	1
"10100"s	5	1
"10101"s	6	1
"101100"s	0	3
"101101"s	7	1
"101110"s	8	1
"101111"s	9	1
"110000"s	10	1
"110001"s	15	1
"1100100"s	0	4
"1100101"s	1	2
"1100110"s	11	1
"1100111"s	12	1
"1101000"s	13	1
"1101001"s	14	1
"1101010"s	16	1
"11010110"s	ESC	
"11010111"s	0	5
"11011000"s	0	6
"11011001"s	2	2
"11011010"s	3	2
"11011011"s	4	2
"11011100"s	17	1
"11011101"s	18	1
"11011110"s	19	1
"11011111"s	20	1
"11100000"s	21	1
"11100001"s	22	1
"11100010"s	23	1
"11100011"s	24	1
"11100100"s	25	1
"11100101"s	30	1
"11100110"s	31	1
"11100111"s	32	1

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

NTC coeff variable length code (NOTE)	run	level
"11101000"s	48	1
"111010010"s	0	7
"111010011"s	1	3
"111010100"s	5	2
"111010101"s	6	2
"111010110"s	7	2
"111010111"s	8	2
"111011000"s	26	1
"111011001"s	27	1
"111011010"s	28	1
"111011011"s	29	1
"111011100"s	33	1
"111011101"s	34	1
"111011110"s	35	1
"111011111"s	36	1
"111100000"s	37	1
"111100001"s	38	1
"111100010"s	39	1
"111100011"s	40	1
"111100100"s	41	1
"111100101"s	42	1
"111100110"s	44	1
"111100111"s	45	1
"111101000"s	46	1
"111101001"s	47	1
"111101010"s	49	1
"111101011"s	52	1
"111101100"s	54	1
"111101101"s	56	1
"111101110"s	63	1
"1111011110"s	0	8
"1111011111"s	1	4
"1111100000"s	2	3
"1111100001"s	4	3
"1111100010"s	9	2
"1111100011"s	11	2
"1111100100"s	13	2
"1111100101"s	14	2
"1111100110"s	15	2
"1111100111"s	43	1

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

Table8b. Variable length codes for NTC_coeff(continued).

NTC coeff variable length code (NOTE)	run	level
"1111101000" s	50	1
"1111101001" s	51	1
"1111101010" s	53	1
"1111101011" s	55	1
"1111101100" s	57	1
"1111101101" s	58	1
"1111101110" s	59	1
"1111101111" s	60	1
"1111110000" s	61	1
"1111110001" s	62	1
"1111110010" s	1	5
"1111110011" s	2	4
"1111110110" s	3	3
"1111110111" s	5	3
"1111111000" s	6	3
"1111111001" s	7	3
"1111111010" s	8	3
"1111111011" s	10	2
"1111111100" s	12	2
"1111111101" s	16	2
"1111111110" s	32	2
"1111111111110" s	0	9
"1111111111111" s	2	5
"1111111100000" s	4	4
"1111111100001" s	9	3
"1111111100010" s	12	3
"1111111100011" s	13	3
"1111111100100" s	14	3
"1111111100101" s	17	2
"1111111100110" s	23	2
"1111111100111" s	24	2
"1111111101000" s	30	2
"1111111101001" s	31	2
"1111111101010" s	55	2
"1111111010110" s	1	6
"1111111010111" s	8	4
"1111111011000" s	10	3
"1111111011001" s	13	4
"1111111011010" s	15	3
"1111111011011" s	16	3

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

NTC coeff variable length code (NOTE)	run	level
"1111111011100" s	18	2
"1111111011101" s	19	2
"1111111011110" s	25	2
"1111111011111" s	29	2
"1111111100000" s	33	2
"1111111100001" s	40	2
"1111111100010" s	45	2
"1111111100011" s	47	2
"1111111100100" s	48	2
"1111111100101" s	57	2
"1111111100110" s	58	2
"1111111100111" s	63	2
"1111111010000" s	1	7
"1111111010001" s	2	6
"1111111010010" s	2	7
"1111111010011" s	3	4
"1111111010100" s	5	4
"1111111010101" s	5	5
"1111111010110" s	6	5
"1111111010111" s	7	4
"1111111011000" s	11	3
"1111111011001" s	12	4
"1111111011010" s	14	4
"1111111011011" s	21	2
"1111111011100" s	22	2
"1111111011101" s	23	3
"1111111011110" s	27	2
"1111111011111" s	27	3
"1111111100000" s	31	3
"1111111100001" s	32	4
"1111111100010" s	34	2
"1111111100011" s	37	2
"1111111100100" s	39	2
"1111111100101" s	41	2
"1111111100110" s	47	3
"1111111100111" s	56	2
"1111111101000" s	60	2
"1111111101001" s	62	2
"1111111101010100" s	1	8
"1111111101010101" s	3	5

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

Table8c Variable length codes for NTC_coeff(continued).

NTC coeff variable length code (NOTE)	run	level
"1111111110010110" s	4	5
"1111111110010111" s	4	6
"1111111111011000" s	5	6
"1111111111011001" s	6	4
"1111111111011010" s	6	6
"1111111111011011" s	6	8
"1111111111011100" s	7	6
"1111111111011101" s	8	5
"1111111111011110" s	9	6
"1111111111011111" s	10	4
"1111111111100000" s	11	4
"1111111111100001" s	13	6
"1111111111100010" s	14	7
"1111111111100011" s	15	4
"1111111111100100" s	16	4
"1111111111100101" s	20	2
"1111111111100110" s	23	4
"1111111111100111" s	25	3
"1111111111101000" s	26	2
"1111111111101001" s	34	4
"1111111111101010" s	36	2
"1111111111101011" s	36	3
"1111111111101100" s	38	2
"1111111111101101" s	42	2
"1111111111101110" s	42	3
"1111111111101111" s	43	2
"1111111111110000" s	43	3
"1111111111110001" s	45	3
"1111111111110010" s	46	2
"1111111111110011" s	47	4
"1111111111110100" s	48	4
"1111111111110101" s	49	2
"1111111111110110" s	50	2
"1111111111110111" s	51	2
"1111111111111000" s	52	2
"1111111111111001" s	53	2
"1111111111111010" s	53	3
"1111111111111011" s	54	2
"1111111111111100" s	56	3
"1111111111111101" s	59	2
"1111111111111110" s	61	2
"1111111111111111" s	62	3

NOTE- The last bit 's' denotes the sign of the level, '0' for positive, '1' for negative.

Method of encoding of run and level following ESC(escape code) is same as MPEG v ideo CD.

10. Compatibility Feature

· Forward compatibility at the proposed algorithm with MPEG1 is achieved by using MPEG1 VLC tables.

· Backward compatibility with MPEG1 is under study.

11. Random Access Feature

Random access is implemented in the same way as MPEG1. At random access the decoding starts at the I_picture in a GOP, and the decoding proceeds to the target picture.

The worst case is access to the last coded B_picture in the GOP. A GOP consists of 12 pictures. So 12 pictures period, i.e. 400msec is the worst case access time.

· Remark: The media access time is ignored.

12. Coding/decoding Delay

12.1. Coding Delay

The total coding delay is the sum of the delays in each module, as follows.

Coding Delay = Reorder&Motion_estimator + DCT + Quant + Scan + VLC + Buffer

· At the Reorder&Motion_estimator, 3frame_delay, i.e. 3/30sec occurs at B_picture because the maximum distance between P_pictures is 4. No delay occurs at I/P_pictures.

· From DCT to VLC, a delay of a number of MacroBlock_periods occurs. This delay is relatively small.

· At the Buffer, the VBV delay occurs. This VBV delay is 768*1024bit_delay for a 4Mbps' bitstream and 1664*1024bit_delay for a 9Mbps' bitstream.

12.2. Decoding Delay

The total decoding delay is the sum of the delays in each module, as follows.

Decoding Delay = Buffer + iVLC + iScan + iQuant + iDCT + Motion_compensator&Reorder

· At the Buffer, the VBV delay occurs. This VBV delay is 1213568(=2000000-768*1024)bit_delay for a 4Mbps' bitstream and 2808064(=4000000-1664*1024)bit_delay for a 9Mbps' bitstream.

· From iVLC to iDCT, a delay of a number of MacroBlock_periods occurs. This delay is relatively small.

· At Reorder&Motion_compensator, 3frame_delay, i.e. 3/30sec occurs at I/P_picture because maximum distance between P_pictures is 4. No delay occurs at B_picture.

13. Other Feature

Fast Forward/Reverse

In Fast Forward/Reverse play back, only I_pictures are decoded. An I_picture needs less than 800Kbits at 4Mbps and less than 1800Kbits at 9Mbps. So the media drive reads the I_picture in less than 6 frame periods.

The decoder repeats the decoded I_picture 6 times while reading the next I_picture. And the I_picture exists in every GOP, i.e. 12 frames. So in Fast Forward/Reverse play back, the decoded pictures are displayed 2(=12/6)times faster than normal play back.

Remark: The media access time is ignored.

14. Statistics

· Number of bits and SNR for each frame are shown in figure as follows.

	bit/sec	figure	bit/sec	figure
· Flower garden	4M	fig15	9M	fig16
Popple	x	x	9M	fig17
Table tennis	4M	fig18	9M	fig19
Mobile and Calendar	4M	fig20	9M	fig21

· Cumulative bit count once every 0.4 second(excluding the last 0.2 second) for each sequence are shown in figure 22.

· Several items averaged over each sequence following the formats in SM3/RM8 tables as a guidline are shown in figure as follows.

	bit/sec	figure
Flower garden	4M,9M	fig23
Popple	9M	fig24
Table tennis	4M,9M	fig25
Mobile and Calendar	4M,9M	fig26

· Listing for each sequence which shown the corresponding coded bit stream file in a format "ls -l" output are

```
-rw-rw-rw-  1 nakajima  2466582 Nov  7 16:20 flower.4M.bbs
-rw-rw-rw-  1 nakajima  5596819 Nov  7 16:20 flower.9M.bbs
-rw-rw-rw-  1 nakajima  5811544 Nov  7 16:20 popple.9M.bbs
-rw-rw-rw-  1 nakajima  2463282 Nov  7 16:20 tennis.4M.bbs
-rw-rw-rw-  1 nakajima  5584182 Nov  7 16:20 tennis.9M.bbs
-rw-rw-rw-  1 nakajima  2445000 Nov  7 16:20 mobile.4M.bbs
-rw-rw-rw-  1 nakajima  5549794 Nov  7 16:20 mobile.9M.bbs
```

15. Implementation

15.1 Introduction

This description is the implementation study of the the proposed coding algorithm for WG11/MPEG November meeting.

15.2 Decoder implementation

Figure27 shows the block diagram of the decoder.

● Code Buf.

a.memory size and width
 off chip ram
  size = 4 Mbits
  width = 1 bit
b.memory bandwidth
 9 Mbit/sec(in) + 9Mbit/sec(out) = 18Mbit/sec.
c.number and width of additions per second
 none
d.number and width of multiplications per second
 none
e.table sizes
 none
f.number of table lookups per second
 none
g.a functional description
 as same as VBV function

● iVLC

a.memory size and width
b.memory bandwidth
 none
c.number and width of additions per second
 shift_bit_calc in shifter:

  width = num_bit_vlc_length = 5bit and 5bit
  number = max_coef_event_per_block times num_block_sec
    = 65x90x60x2x30 = 2106000 times/sec.

d.number and width of multiplications per second
 none
e.table sizes
 e.1 coef vlc table for inter predict mb in DCT mode :
  fixed
  num_entry = 113
  max_width = 16
  size = 113x16 = 1808 bit
 e.2 coef vlc table for intra predict mb in DCT mode:
  fixed
  num_entry = 310
  max_width = 20
  size = 310x20 = 6200 bit
 e.3 coef vlc table for all mb in non DCT mode:
  fixed
  num_entry = 202
  max_width = 15
  size = 202x15 = 3030 bit
 e.4 motion vector vlc table
  fixed
  num_entry = 17
  max_width = 10
  size = 17x10 = 170 bit
 e.5 mbtype vlc table for I-pict
  fixed
  num_entry = 42
  max_width = 11
  size = 42x11 = 462 bit
 e.6 dc_coef vlc table
  fixed
  num_entry = 18
  max_width = 8
  size = 18x8 bit = 144 bit
f. number of table lookups per second
 f.1 coef vlc table:
  max_number = max_coef_event_per_block x num_block_sec
    = 65 x 90x60x2x30 = 21060000 times/sec.
 f.2 motion vector vlc table
  max_number = max_num_mc_mb_sec
    = 45x60x30 = 81000 times/sec.
 f.3 dc_coef vlc table
  max_number = max_num_intra_block_sec
    = 90x60x2x30 = 324000 times/sec.
g.a functional description
 refer to Figure28

● Inverse Differential

a.memory size and width
b.memory bandwidth
 none
c.number and width of additions per second
 c.1.motion vector inverse dpcm
  width = 9 bit and 9 bit
  max number = mb_num_sec = 45x60x30 = 81000 times/sec.
 c.2.DC_coef inverse dpcm
  width = 9 bit and 9 bit
  max number = block_num_sec = 90x60x2x30 times/sec.
 c.3.NTC_coef inverse dpcm
  width = 9 bit and 9 bit
  max number = num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec.
d.number and width of multiplications per second
 none
e.table sizes
f.number of table lookups per second
 none
g.a functional description
  motion vector : as MPEG-1
  DC_coef : as MPEG-1
  NTC_coef : refer Figure29

● iScan Conv.

a.memory size and width
  input buffer
   size = num_coef_block x length_q_coef x 2(bank)
      = 64 x 9 x 2 = 1152 bit
   width= 9 bit
b.memory bandwidth
  input buffer
   bandwidth = length_q_coef x num_coef_pict x 30
      = 9 x (720x480x2) x 30 = 186.62 Mbit/sec.
c.number and width of additions per second
  none
d.number and width of multiplications per second
  none
e.table sizes
  scan type table:
   on chip fix.
   size = bit_coef_address x 64 x num_scan_type
      = 6x64x5=1920 bit
f.number of table lookups per second
  number = num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec.
g.a functional description
  refer Figure30

● iQuant

a.memory size and width
b.memory bandwidth
  none
c.number and width of additions per second
  c.1 addition BASE
   width = 9 bit and 9 bit
   max_number = num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec.
  c.2 addition dead_zone_level :
   width = 12 bit and 12 bit
   max_number = num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec.
d.number and width of multiplications per second
  d.1 iQuant:
   width = length_q_coef by bit_stepsize = 9bit by 7 bit
   number = num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec.
  d.2 iweight:
   width = length_coef by bit_weight = 12bit by 8 bit
   number = num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec.
e.table sizes
  weight:
   on chip ram, downloadable
   size = width x 64 x num_type_matrix
      = 8 x 64 x 1 = 512 bit
f.number of table lookups per second
   number = num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec.
g.a functional description
   refer Figure31

● iDCT

a.memory size and width
 h/v conv
  on chip
  size = length_coef x 64 x 2 = 16 x 64 x 2 = 16x64x2=2048bit
  width = 16 bit
b.memory bandwidth
  bandwidth = length_coef x num_coef_sec
      = 16 x (720x480x2) x 30 = 331.776 Mbit/sec.
c.number and width of additions per second
   width = 23 bits and 23 bits
   7x8x2 x 90x60x2 x 30 = 36288000 times/sec.
d.number and width of multiplications per second
   width = 16 bits by 14 bits
   8x8x2 x 90x60x2 x 30 = 41472000 times/sec.
e.table sizes
   fixed
   14 x 64 = 896 bit
f.number of table lookups per second
   8x8x2 x 90x60x2 x 30 = 41472000 times/sec.
g.a functional description
   as MPEG-1

● Predictor

a.memory size and width
  Forward and backward decoded pict_memory for prediction.
  SIZE = width x pixel_pict x 3 = 8 x 720 x 480 x 2 x 3 = 16588800bit
  width = 8bit
b.memory bandwidth
  bandwidth = width x pixel_pict x 30 = 158.203125Mbit/sec
c.number and width of additions per second
  addition for a bidirectional predicted MB
  width = 8bit and 8bit
  max_number = 720x480x2x30 = 20736000 times/sec
d.number and width of multiplications per second
  none
e.table sizes
  none
f.number of table lookups per second
  none
g.a functional descriptions
  refer Figure32

●Line Scan Conversion

a.memory size and width
  a.1 scan conv(mb->line)
   size = length_pixel x pixel_line x mb_height x bank x (Y and C)
      = 8 x 16 x 720 x 2 x 2 = 360x1024 bit
   width = 8 bit
b.memory bandwidth
   bandwidth = width x pixel_rate_display x 1(read or write)
      = 8 x 13.5MHz = 108 Mbit/sec.

● Prediction

as same as Inverse Prediction

● DCT

as same as Inverse DCT

● Scan conv.

as same as Block Invers scan conversion

● Quant

a.memory size and width
b.memory bandwidth
none
c.number and width of additions per second
c.1 subtraction BASE
width =length_pixel and bit_BASE = 9bit and 9 bit
max number=64x90x60x2x30=20736000 times/second
d.number and width of multiplications per second
d.1 weighting
width = length_coef by bit_weight
= 12bit by 20 bit
max number= num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec
d.2 Quant
width = length_coef by bit_stepsize = 12bit by 8 bit
number = num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec
e.table sizes
e.1 weight
on chip ram, downloadable
size = width x 64 x num_type_matrix
= 20 x 64 x 1 = 1280 bit
f.number of table lookups per second
number = num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec.
g.a functional description
refer Figure36

● Differential

a.memory size and width
b.memory bandwidth
none
c.number and width of additions per second
c.1 motion vector dpcm
width = 10 bit and 10 bit
number = mb_num_sec = 45x60x30 = 81000 times/sec.
c.2 DC_coef dpcm
width = 9 bit and 9 bit
number = mb_num_sec = 90x60x2x30 = 324000 times/sec.
c.3 NTC coef dpcm
width = 9 bit and 9 bit
max number = num_coef_sec = 64 x 90x60x2x30 = 20736000 times/sec.
d.number and width of multiplications per second
none
e.table sizes
f.number of table lookups per second
c.number and width of additions per second
none
d.number and width of multiplications per second
none
e.table sizes
none
f.number of table lookups per second
none
g.a functional description
MB to line conversion

15.4 Encoder implementation

Figure33 shows the block diagram of the encoder.

● Line Scan Conversion

as same as Decoder

● Motion Estimator(with Picture Reorder)

a.memory size and width
a.1 Original pict memory for Picture Reorder and Motion Estimation
5x720x480x2x8 = 27648000bit
a.2 Delay memory for Motion Estimation
8x16x8x720 = 737280bit
width 8bit
b.memory bandwidth
bandwidth = width x pixel_pict x 30 x 1 = 158.203125Mbit/sec
c.number and width of additions per second
maximum case of B-picture (Forward frame_distance=1, Backward frame_distance=3)

pixel_MB x num_vector x 3
=pixel_MB x (search_range x 2 x frame_distance +1)^2 x 3

frame mode forward ; 16x16 x (8 x 2 x 1 + 1)^2 x 3 = 221952 times
backward; 16x16 x (8 x 2 x 3 + 1)^2 x 3 = 1843968 times
field mode forward ; 221952 x 5/2 = 554880 times
backward; 1843968 x 4/2 = 3687936 times

total 6308736 times/MB

half_pel
pixel_MB x num_calc_ave. x 3 + pixel_MB x 3 =
16x16 x 8 x 3 + 16x16 x 3 = 6192 times/MB

then (6308736 + 6192)x 45 x 30 x 30 = 243906Mtimes/sec

d.number and width of multiplications per second
none
e.table sizes
none
f.number of table lookups per second
none
g.functional descriptions
refer Figure34 and 35

none
g.a functional description
motion vector : as MPEG-1
DC_coef : as MPEG-1
NTC : refer refer Figure37

● VLC&FLC

a.memory size and width
b.memory bandwidth
c.number and width of additions per second
d.number and width of multiplications per second
e.table sizes
f.number of table lookups per second
as same as Inverse VLC
g.a functional description
refer Figure38

● Code Buf.

as same as decoder Code Buffer

● Rate Ctrl

a.memory size and width
1.memory for MB
on chip
size = length_pixel x MB_size x 2(Original & Differential)
= 8x16x16x2x2 = 8192bit
width = 8 bit
b.memory bandwidth
bandwidth = width x MB x 2(Original & Differential)
= 8x64x8x2x(45x30) = 10.6Mbit/sec.
c.number and width of additions per second
c.1 additions for the filter
width = 8 bit and 8 bit
max_number = 2 x num_pixel = 2x720x480x2x30 = 41472000times/sec.

c.2 additions for the activity
width = 8 bit and 8 bit
max_number = 64xMB_num = 720x480x2x30 = 20736000times/sec.
c.3 addition for stepsize adjustment
width = 8 bit and 8 bit
max_number = 1xMB_num = 45x30x30 = 40500times/sec.
d.number and width of multiplications per second
d.1 multiplications for Filter
width = 8 bit and 8 bit
max_number = 3xnum_pixel = 3x720x480x2x30 = 62208000times/sec.
d.2 multiplications for Stepsize Modification
width = 8 bit and 8 bit
max_number = 2xMB_num = 2x45x30x30 = 81000times/sec.
e.table sizes
none
f.number of table lookups per second
none
g.a functional description

Rate Control is a typical encoder issue, so an example of rate control is attached as follows.

Basically stepsize is set by SM3 rate control for each slice.
Then stepsize is modified by the activity of target MB(Macro Block).

(1)The activity is calculated at target MB.
The activity is calculated with a filter.
An example of the filter is 1_dimensional 3tap filter.
The filter is applied to MB horizontally and vertically.
(2)Stepsize is modified according to the activity.
Stepsize is modified with a following formula.

$$MB\_stepsize = A \cdot Slice\_stepsize + B \cdot activity \quad A,B:constant$$

Nov 7 21:54 1991 Flower_garden_4Mbit/sec Page 1

No.	bit	SNRY	SNRCb	SNRCr
0	529230	33.165	35.112	35.429
1	36539	27.870	33.312	34.314
2	51003	30.359	34.221	34.784
3	117898	26.236	32.939	33.932
4	52193	29.934	34.061	34.693
5	62977	29.066	33.696	34.418
6	275649	30.428	33.224	33.868
7	71086	28.462	32.868	33.779
8	44300	28.740	33.034	33.824
9	192947	28.869	32.223	33.220
10	41547	28.767	32.463	33.425
11	67283	28.359	32.583	33.420
12	551221	32.848	34.966	35.095
13	54395	28.172	33.124	33.872
14	45709	29.940	34.010	34.569
15	168829	29.246	33.188	34.082
16	46570	29.868	33.958	34.613
17	66229	29.005	33.646	34.320
18	205056	29.337	32.689	33.498
19	86900	27.988	32.796	33.755
20	52492	28.620	32.773	33.666
21	210915	29.384	32.592	33.351
22	57261	28.371	32.235	33.182
23	39613	28.283	32.537	33.364
24	551487	32.760	34.897	34.970
25	58175	28.693	33.179	33.788
26	48945	29.453	33.944	34.283
27	154872	29.353	33.307	33.991
28	46483	29.782	33.919	34.446
29	54136	28.811	33.561	34.261
30	181980	29.695	32.970	33.700
31	51085	28.794	33.048	33.815
32	34778	28.745	33.087	33.880
33	137231	28.307	32.075	33.098
34	81124	27.950	32.528	33.438
35	48634	27.677	32.305	33.407
36	564923	32.664	34.814	34.780
37	42462	27.735	31.854	33.080
38	30907	29.711	33.959	34.374
39	166875	29.898	33.505	34.158
40	68011	29.090	33.959	34.405
41	59062	28.566	33.515	34.238
42	262588	30.098	32.899	33.627
43	53391	28.700	32.825	33.745
44	49111	28.209	33.042	33.771
45	195696	29.233	32.369	33.174
46	59974	28.492	32.574	33.318
47	53349	28.420	32.430	33.207
48	586604	32.531	34.483	34.496
49	74081	27.972	32.341	33.290
50	42851	28.510	33.571	34.022
51	150617	29.907	33.249	33.994
52	45190	29.506	33.596	34.036
53	77921	28.331	33.472	34.096
54	318101	31.079	33.340	33.846

Nov 7 21:54 1991 Flower_garden_4Mbit/sec Page 2

55	64201	28.802	33.102	33.858
56	54876	28.649	33.130	33.743
57	190371	29.349	32.465	33.338
58	59605	28.662	32.828	33.546
59	68987	28.317	32.495	33.400
60	591664	32.474	34.393	34.627
61	69370	28.232	32.166	33.194
62	61744	28.790	33.668	34.115
63	219446	30.284	33.350	33.831
64	54481	29.048	33.641	34.041
65	67850	28.765	33.525	34.013
66	211887	30.082	33.001	33.445
67	43687	28.989	32.792	33.469
68	42560	29.133	32.968	33.414
69	181989	28.388	31.956	32.816
70	64543	28.219	32.392	33.161
71	41122	27.629	32.166	33.001
72	612466	32.160	34.248	34.357
73	37018	27.719	31.610	32.621
74	61386	28.655	32.823	33.482
75	588539	28.783	32.686	33.483
76	67148	28.228	33.067	33.744
77	48179	28.267	32.855	33.554
78	184740	28.358	31.977	32.960
79	60170	27.696	32.201	33.178
80	41655	27.641	32.147	33.050
81	181911	28.211	31.666	32.651
82	36274	27.834	31.662	32.693
83	66145	27.854	31.758	32.748
84	694934	32.373	34.285	34.271
85	46534	27.578	31.786	32.846
86	34114	29.086	33.495	33.798
87	96320	28.393	32.760	33.328
88	42753	29.631	33.489	33.754
89	44383	28.996	33.251	33.654
90	208297	28.243	31.800	32.576
91	68117	27.700	32.161	32.969
92	27211	27.665	32.167	32.983
93	133590	27.122	31.044	32.151
94	61123	27.618	31.368	32.315
95	31707	26.771	31.279	32.380
96	693207	32.288	34.237	34.266
97	63904	26.527	30.982	32.233
98	60787	27.358	32.681	33.398
99	134781	28.377	32.668	33.257
100	61938	27.795	33.163	33.584
101	82948	27.743	33.048	33.611
102	231766	28.712	31.980	32.630
103	66747	26.929	32.148	32.952
104	79247	27.393	32.118	32.814
105	231976	28.715	31.737	32.344
106	67597	26.838	31.707	32.489
107	84927	27.327	31.726	32.422
108	624821	32.168	34.048	34.050
109	59049	27.119	31.991	32.685
110	55055	27.729	33.036	33.405

Nov 7 21:54 1991 Flower_garden_4Mbit/sec Page 3

111	161216	28.824	32.673	33.179
112	61092	28.443	33.147	33.444
113	83191	27.708	32.920	33.367
114	276746	29.640	32.493	33.014
115	67357	27.618	32.003	32.789
116	60330	27.641	32.665	33.051
117	213311	28.804	31.977	32.609
118	58441	28.160	32.140	32.780
119	38737	27.804	32.060	32.763
120	616839	32.174	34.187	34.179
121	49127	27.991	31.692	32.844
122	40604	28.424	33.046	33.477
123	163591	28.606	32.688	33.091
124	63881	28.721	33.344	33.681
125	61406	28.091	32.721	33.306
126	196464	28.047	31.782	32.664
127	39725	27.434	32.006	32.787
128	68612	27.013	32.096	32.802
129	203468	27.805	31.402	32.204
130	40957	27.294	31.369	32.316
131	35575	27.396	31.869	32.406
132	651336	30.642	33.019	33.171
133	64189	27.114	31.093	32.052
134	28984	27.193	31.813	32.462
135	79757	28.130	32.081	32.674
136	43297	28.719	32.378	32.826
137	48674	28.191	32.263	32.820
138	184720	28.108	31.668	32.308
139	38809	27.367	31.660	32.404
140	23668	27.616	31.761	32.411
141	638846	31.970	33.901	33.801
142	49931	27.886	31.331	32.296
143	28136	29.038	32.933	33.224
144	120642	27.426	32.129	32.861
145	47209	29.873	33.169	33.395
146	62626	28.166	32.788	33.114
147	202119	27.776	31.499	32.349
148	36841	26.468	31.586	32.468
149	28173	26.715	31.670	32.472

Figure 15

7 21:55 1991 Flower_garden_9Mbit/sec Page 1

No.	bit	SNRY	SNRCb	SNRCr
0	907410	37.949	39.046	38.910
1	77995	30.393	35.467	36.306
2	163396	34.453	37.251	37.495
3	274896	32.494	35.552	36.165
4	118195	33.615	37.065	37.302
5	118236	33.411	36.778	37.070
6	548009	35.646	37.043	36.917
7	135366	32.973	36.037	36.238
8	121485	33.207	36.088	36.341
9	464831	34.646	36.171	36.214
10	115639	33.220	35.737	35.963
11	129952	32.989	36.122	36.224
12	1395876	41.197	42.210	41.914
13	141455	33.177	37.002	37.195
14	103219	33.828	37.822	38.344
15	420491	34.874	37.061	37.290
16	144780	34.501	38.338	38.699
17	103072	33.092	37.753	37.896
18	499269	35.380	36.994	36.844
19	226147	33.489	36.328	36.486
20	116786	32.984	36.207	36.421
21	512530	35.525	37.006	36.794
22	116972	33.068	36.015	36.054
23	106788	33.301	36.421	36.408
24	1412321	41.037	42.102	41.814
25	122743	33.234	37.288	37.417
26	116091	33.268	37.678	38.001
27	343457	34.135	36.779	36.998
28	105198	33.736	38.010	38.201
29	120847	32.961	36.799	37.201
30	389147	34.546	36.621	36.629
31	108338	33.089	36.325	36.483
32	107157	32.995	36.253	36.564
33	348896	33.645	35.739	35.782
34	158419	32.437	35.745	35.981
35	145329	32.424	35.490	35.887
36	1450292	41.074	42.033	41.664
37	79218	32.510	35.680	35.915
38	83169	34.100	37.737	38.119
39	360911	34.765	37.251	37.498
40	126314	33.155	37.819	38.121
41	136682	32.742	36.749	37.154
42	523956	35.300	36.807	36.683
43	95204	32.165	35.794	36.414
44	130130	32.911	36.493	36.661
45	507687	35.291	36.605	36.371
46	114710	33.091	36.369	36.149
47	117988	32.790	36.102	35.981
48	1490931	40.853	41.801	41.497
49	143221	32.828	36.143	36.291
50	106210	32.396	37.118	37.818
[illegible]	330697	34.710	37.181	37.534
[illegible]	109809	33.358	37.672	38.366
[illegible]	153881	32.868	37.046	37.493
[illegible]	646300	36.560	37.583	37.355

Nov 7 21:55 1991 Flower_garden_9Mbit/sec Page 2

55	128784	32.805	36.698	36.887
56	129195	32.723	36.373	36.626
57	496422	35.339	36.689	36.437
58	114060	32.968	36.515	36.503
59	183154	33.488	36.206	36.136
60	1496459	40.790	41.698	41.406
61	95949	32.133	35.654	36.025
62	130039	33.035	37.206	37.269
63	462161	36.443	37.238	37.286
64	122252	33.187	37.706	37.893
65	125142	33.058	37.377	37.396
66	465186	35.220	36.846	36.644
67	89094	33.408	36.522	36.649
68	92625	33.180	36.414	36.431
69	398906	33.316	35.185	35.272
70	125787	32.811	35.874	35.869
71	136431	32.233	35.198	35.394
72	1561127	40.661	41.594	41.300
73	84914	31.639	34.907	35.172
74	79360	32.392	36.420	36.598
75	375972	33.602	36.013	36.273
76	119344	32.426	36.664	36.956
77	120000	32.275	35.842	36.190
78	391761	33.326	36.227	35.349
79	90366	32.191	35.527	35.675
80	133553	32.104	35.106	35.334
81	386533	33.184	34.932	34.939
82	61705	31.668	34.678	34.840
83	122405	32.304	34.913	35.123
84	1231572	38.863	39.668	39.273
85	78158	32.908	36.506	36.392
86	101843	33.317	36.860	36.741
87	236686	32.474	35.932	36.197
88	88746	33.620	37.137	37.184
89	100904	33.005	36.585	36.794
90	390134	32.441	34.547	34.662
91	146507	32.045	35.190	35.398
92	68049	31.193	34.750	35.069
93	349359	32.276	34.258	34.302
94	94896	31.646	34.284	34.416
95	97211	31.370	34.267	34.463
96	1233314	38.820	39.629	39.218
97	127247	31.217	34.835	34.655
98	131987	31.472	35.591	35.720
99	310403	32.807	35.607	35.913
100	156814	32.291	36.398	36.536
101	174944	32.096	35.950	36.262
102	489575	33.831	35.393	35.306
103	140968	31.210	35.177	35.388
104	185499	31.671	35.023	35.210
105	601302	33.914	35.241	35.087
106	136369	31.279	34.828	34.913
107	170634	31.889	34.964	34.912
108	1298580	38.796	39.573	39.145
109	175980	31.991	35.663	35.626
110	132543	31.612	35.946	35.993

Nov 7 21:55 1991 Flower_garden_9Mbit/sec Page 3

111	401456	34.177	36.274	36.317
112	138513	32.513	36.668	36.696
113	139321	31.257	35.850	36.038
114	673662	34.940	36.230	36.997
115	163140	31.833	35.015	35.167
116	120709	31.686	36.618	36.619
117	471408	33.941	36.432	36.281
118	112836	32.767	35.762	36.620
119	129407	32.211	35.235	36.184
120	1285963	38.776	39.629	39.186
121	88929	31.686	34.777	34.926
122	79104	32.704	36.764	36.664
123	373829	33.393	35.733	36.767
124	167516	33.627	37.069	37.027
125	124604	32.137	35.486	35.786
126	421796	32.953	34.922	34.994
127	101209	31.259	34.862	34.996
128	127284	31.177	34.962	36.034
129	626691	33.649	35.240	34.957
130	79233	32.135	35.086	34.900
131	108386	32.343	36.129	34.922
132	1334227	38.638	39.489	38.962
133	93634	31.324	34.654	34.664
134	95163	31.702	35.670	36.790
135	195064	32.441	35.878	36.245
136	82983	32.248	36.421	36.721
137	124434	32.296	36.101	36.484
138	368945	32.314	34.747	34.976
139	70464	30.906	34.648	35.302
140	90038	31.276	34.827	35.139
141	1338858	38.688	39.468	38.950
142	72269	32.260	35.669	36.630
143	88743	33.207	36.576	36.675
144	337857	32.331	35.176	35.364
145	87598	33.097	36.701	36.896
146	103627	32.604	36.220	36.229
147	446839	32.730	34.722	34.617
148	88963	31.362	34.862	34.808
149	143873	31.661	34.776	34.719

Figure. 16

Nov 7 21:55 1991 Popple_9Mbit/sec Page 1

No.	bit	SNRY	SNRCb	SNRCr
0	791921	37.784	44.790	44.346
1	196067	34.896	40.928	40.702
2	314375	35.914	42.097	41.550
3	143950	34.259	40.209	40.163
4	402970	36.506	42.130	41.648
5	156683	34.527	40.332	40.185
6	358368	36.225	41.545	41.257
7	149817	34.603	40.207	40.239
8	366018	36.118	41.395	41.079
9	150886	34.563	40.139	40.057
10	372113	36.050	41.239	40.906
11	156721	34.531	40.007	39.940
12	817182	37.730	44.523	44.190
13	157316	34.663	40.410	40.281
14	353663	36.150	42.208	41.633
15	153756	34.616	40.802	40.796
16	363574	36.073	41.618	41.287
17	154567	34.467	40.240	40.158
18	371974	36.143	41.488	41.169
19	150933	34.569	40.037	40.121
20	370389	36.135	41.407	41.062
21	154724	34.362	39.990	39.948
22	373660	36.118	41.248	40.957
23	149704	34.602	40.029	40.053
24	796127	37.911	44.792	44.263
25	151650	34.604	40.380	40.399
26	342427	36.208	42.474	41.899
27	149151	34.717	40.718	40.645
28	359009	36.212	41.876	41.489
29	155117	34.510	40.397	40.363
30	363063	36.177	41.649	41.273
31	152282	34.564	40.105	40.173
32	371535	36.167	41.526	41.248
33	150525	34.492	40.195	40.134
34	369757	36.190	41.313	40.985
35	149671	34.497	39.970	40.046
36	793930	37.882	44.903	44.347
37	146525	34.575	40.509	40.584
38	341222	36.218	42.237	41.786
39	141247	34.728	40.794	40.812
40	362071	36.160	41.684	41.431
41	170988	34.770	40.503	40.656
42	365370	36.167	41.638	41.240
43	146935	34.459	40.048	40.169
44	331297	35.793	41.025	40.767
45	154278	34.363	39.968	40.046
46	335196	35.869	40.875	40.727
47	148690	34.355	39.887	39.980
48	813018	37.819	44.799	44.154
49	169835	34.870	40.702	40.658
50	341870	36.102	42.212	41.738
51	146892	34.598	40.647	40.717
52	373346	36.134	41.669	41.297
53	173662	34.751	40.541	40.591
54	373802	36.018	41.228	40.874

Nov 7 21:55 1991 Popple_9Mbit/sec Page 2

55	178107	34.480	40.137	40.196
56	374615	36.084	41.307	40.807
57	181146	34.410	39.909	39.866
58	359143	36.178	41.292	40.838
59	178330	34.625	40.023	39.823
60	773482	37.905	44.894	44.321
61	172436	34.829	40.626	40.461
62	395859	36.642	42.299	41.746
63	183349	34.497	40.728	40.719
64	396777	36.631	41.971	41.547
65	193887	34.431	40.133	40.306
66	398568	36.637	42.001	41.590
67	199800	34.467	40.002	40.276
68	403592	36.592	41.894	41.419
69	186973	34.736	40.352	40.284
70	396139	36.672	41.913	41.469
71	184255	34.790	40.159	40.283
72	702695	38.115	44.984	44.613
73	149797	34.387	39.767	40.120
74	343792	36.849	42.384	41.685
75	148843	34.895	40.205	40.348
76	357214	36.738	42.213	41.393
77	149880	35.049	39.660	39.754
78	338729	36.600	41.734	40.865
79	133597	35.078	39.497	39.411
80	330862	36.429	41.275	40.583
81	148051	35.041	39.347	39.133
82	342288	36.369	41.200	40.647
83	158516	36.016	38.689	38.747
84	659516	38.384	45.630	44.605
85	136820	34.617	38.025	38.268
86	323433	36.307	40.693	40.311
87	143104	34.770	37.712	37.983
88	346266	36.075	39.896	39.836
89	162682	34.432	36.875	37.320
90	352714	35.710	39.011	39.216
91	163992	33.962	36.083	36.633
92	356202	35.244	38.135	38.564
93	160224	33.368	35.090	35.622
94	377198	34.660	37.641	37.922
95	168206	32.633	34.427	35.072
96	932405	38.695	44.685	44.175
97	163540	32.048	34.134	34.854
98	355337	34.040	37.220	37.793
99	163083	31.780	33.574	34.310
100	393380	33.468	36.921	36.702
101	162783	30.643	32.693	33.256
102	404442	32.920	35.317	35.946
103	172709	30.638	32.384	33.059
104	395534	32.752	34.853	35.428
105	180677	30.357	31.630	32.221
106	391108	32.511	34.281	35.036
107	171851	29.955	31.319	32.042
108	1175934	39.102	44.157	43.878
109	170133	30.254	31.891	32.525
110	334768	32.227	34.588	35.165

Nov 7 21:55 1991 Popple_9Mbit/sec Page 3

111	174488	30.367	31.762	32.379
112	368171	32.247	33.643	34.433
113	177653	29.702	31.085	31.622
114	364752	31.929	33.280	34.088
115	176317	29.859	31.077	31.727
116	362535	31.866	33.369	33.909
117	174969	30.164	30.802	31.460
118	383296	32.086	33.443	34.110
119	175273	29.644	30.739	31.283
120	1236428	38.937	43.923	44.048
121	170140	29.642	31.709	32.144
122	353438	31.810	34.663	35.164
123	173635	29.926	31.945	32.616
124	392995	31.691	33.798	34.377
125	169385	29.019	31.336	31.788
126	392701	31.364	33.555	34.166
127	181877	29.477	31.218	31.842
128	384998	31.609	33.671	34.171
129	187168	29.496	31.209	31.686
130	1236853	38.992	43.966	43.831
131	163369	29.244	31.712	32.092
132	364449	31.368	34.280	34.824
133	172438	29.545	31.806	32.240
134	405401	31.496	34.072	34.681
135	181180	29.360	31.400	31.716
136	384663	31.419	33.584	34.046
137	166639	28.785	31.176	31.680
138	380306	31.070	33.444	33.940
139	183407	28.937	30.975	31.433
140	589339	34.458	38.328	38.672
141	180968	29.804	31.626	31.943
142	386625	31.666	34.037	34.618
143	169881	29.108	31.686	32.046
144	393886	31.217	33.764	34.348
145	181899	29.086	31.216	31.708
146	400018	31.422	33.496	34.180
147	181221	29.467	31.381	31.792
148	400593	31.346	33.398	34.021
149	171936	28.634	30.716	31.277

Figure. 17

Nov 7 21:55 1991 Table_tennis_4Mbit/sec Page 1

No.	bit	SNRY	SNRCb	SNRCr
0	459717	26.259	38.479	39.822
1	48042	25.468	38.268	39.124
2	237394	26.130	38.537	39.519
3	36987	25.559	38.334	39.282
4	87836	25.313	38.477	38.662
5	131410	26.048	38.552	39.239
6	148543	25.394	38.253	38.531
7	23688	25.084	38.235	38.193
8	135003	25.543	38.211	38.326
9	60882	25.022	38.204	38.429
10	225530	26.335	38.204	38.774
11	50355	25.397	38.092	38.422
12	493925	26.955	38.609	40.063
13	92803	26.141	38.278	39.031
14	152170	26.009	38.482	39.479
15	85969	25.881	38.443	39.507
16	250008	26.689	38.477	39.534
17	51547	25.684	38.397	39.106
18	197294	26.452	38.345	39.144
19	77936	25.804	38.333	39.118
20	172427	25.992	38.045	38.714
21	46489	25.960	38.245	38.830
22	238874	26.359	37.906	38.545
23	136054	25.759	37.980	38.659
24	476097	28.735	39.148	40.446
25	133668	25.930	38.268	39.069
26	289715	29.302	39.003	39.849
27	90423	27.211	38.797	39.437
28	179263	28.625	38.711	38.988
29	91909	27.864	38.772	38.985
30	175707	28.920	38.584	38.662
31	73843	28.006	38.627	38.708
32	145302	29.315	38.344	38.138
33	63748	28.508	38.621	38.527
34	127427	29.804	38.334	38.001
35	60357	29.063	38.292	38.164
36	318904	31.933	40.335	41.006
37	46152	30.228	39.351	39.313
38	84099	31.002	39.445	39.323
39	47329	30.691	39.714	39.795
40	117685	31.496	39.162	38.804
41	56720	31.101	39.219	38.923
42	122088	31.796	38.823	38.379
43	52930	31.112	38.886	38.310
44	124134	32.085	38.642	37.898
45	54146	31.323	38.447	37.658
46	124580	32.203	38.306	37.791
47	48852	31.358	38.217	37.312
48	307256	34.421	40.299	40.742
49	55091	32.159	38.896	38.221
50	115216	32.898	38.737	38.523
51	51660	32.265	39.079	38.764
52	138427	32.802	38.198	38.129
53	54719	32.012	38.142	37.863
54	142553	32.792	38.023	37.966

Nov 7 21:55 1991 Table_tennis_4Mbit/sec Page 2

55	53817	32.059	37.887	37.755
56	125370	32.381	37.596	37.562
57	55514	31.994	37.599	37.510
58	134402	32.266	37.473	37.414
59	52962	31.682	37.439	37.401
60	362366	34.751	39.745	40.212
61	51582	32.389	38.206	38.529
62	118527	32.968	38.660	38.834
63	48892	32.701	38.869	39.197
64	162969	33.060	38.432	38.549
65	56236	32.373	38.442	38.468
66	595796	34.745	40.302	40.401
67	52460	31.330	37.440	37.447
68	87100	33.439	39.911	39.887
69	26608	33.009	39.914	39.696
70	162814	34.138	40.014	40.040
71	38780	33.389	39.903	39.649
72	313138	31.428	38.374	38.176
73	79402	32.647	39.218	38.927
74	174445	32.658	38.603	38.693
75	53723	31.844	38.511	38.312
76	74969	32.216	38.392	38.312
77	38570	32.105	38.469	38.339
78	124988	32.786	38.481	38.419
79	48702	32.641	38.476	38.455
80	110042	32.845	38.481	38.415
81	47006	32.624	38.453	38.410
82	105002	32.892	38.550	38.430
83	48423	32.684	38.522	38.414
84	423131	33.277	39.329	39.522
85	79485	33.317	39.167	39.303
86	159830	33.678	39.358	39.486
87	55779	32.994	39.291	39.334
88	73645	33.239	39.240	39.321
89	46244	33.134	39.274	39.289
90	130868	33.718	39.303	39.485
91	42486	33.294	39.257	39.376
92	121810	33.748	39.348	39.486
93	60319	33.627	39.332	39.435
94	85352	33.679	39.263	39.375
95	41351	33.358	39.311	39.384
96	507322	33.167	39.678	39.839
97	73316	33.461	39.195	39.242
98	213897	33.078	39.426	39.629
99	61522	32.868	39.460	39.534
100	117205	31.603	39.104	39.129
101	40737	32.197	39.278	39.373
102	155656	31.897	39.031	39.114
103	69883	32.130	39.107	39.258
104	114606	31.384	38.825	38.860
105	42051	31.661	38.972	39.027
106	118141	31.304	38.710	38.753
107	74421	31.766	38.819	38.974
108	507659	33.187	39.625	39.951
109	31463	31.827	39.529	39.725
110	94021	32.151	39.419	39.608

Nov 7 21:55 1991 Table_tennis_4Mbit/sec Page 3

111	41314	32.203	39.474	39.677
112	126264	32.174	39.322	39.542
113	65469	32.198	39.357	39.621
114	96039	31.724	39.190	39.343
115	26024	31.665	39.278	39.368
116	127894	31.943	39.135	39.260
117	70989	31.896	39.182	39.349
118	123200	31.489	38.896	39.096
119	33169	31.432	38.872	39.149
120	507139	33.132	39.527	39.847
121	47910	32.144	39.432	39.611
122	112764	31.430	39.089	39.118
123	62081	31.926	39.301	39.409
124	164776	31.963	39.091	39.181
125	95813	30.976	38.936	38.984
126	155356	31.677	38.776	38.780
127	63805	30.733	38.901	38.982
128	188018	31.987	38.720	38.760
129	59845	30.374	38.465	38.622
130	604092	34.180	40.089	40.630
131	31698	32.285	39.688	39.870
132	113632	32.661	39.663	39.800
133	58197	32.608	39.822	40.076
134	133839	32.132	39.044	39.190
135	63775	31.173	39.468	39.629
136	154640	32.166	38.899	39.098
137	66253	31.088	38.616	38.760
138	151787	32.164	38.791	39.072
139	34182	31.849	38.767	38.916
140	697453	34.149	40.006	40.408
141	78500	31.700	39.182	39.608
142	131703	32.188	39.280	39.662
143	65937	31.442	39.496	39.794
144	163206	32.460	38.934	39.225
145	70146	32.172	39.237	39.358
146	198111	32.707	38.680	38.886
147	74455	31.011	38.857	39.013
148	173864	32.230	38.483	38.667
149	25949	29.137	38.391	38.408

Figure. 18

Nov 7 21:55 1991 Table_tennis_9Mbit/sec Page 1

No.	bit	SNRY	SNRCb	SNRCr
0	1127410	35.519	41.668	43.397
1	49915	30.051	40.448	41.454
2	446777	31.584	40.611	42.540
3	266015	32.664	40.816	42.535
4	319862	29.954	40.775	42.104
5	205935	30.569	40.636	42.067
6	328405	29.960	40.461	41.689
7	64991	29.459	40.356	41.297
8	373468	30.078	40.321	41.475
9	206682	29.558	40.229	41.477
10	422519	30.597	40.310	41.767
11	108966	28.872	40.006	41.136
12	1134252	35.639	41.825	43.528
13	196864	29.591	40.730	41.869
14	428153	31.172	40.771	42.446
15	156481	29.914	40.769	42.236
16	502671	31.890	40.559	42.381
17	133685	29.741	40.455	41.805
18	421668	31.378	40.381	42.000
19	138864	29.828	40.316	41.750
20	453136	31.449	40.009	41.652
21	120403	30.060	40.138	41.521
22	519200	31.583	39.652	41.141
23	237813	29.707	39.497	40.919
24	1201136	37.750	43.136	44.536
25	292058	29.846	39.840	41.145
26	514892	33.677	41.060	42.565
27	205858	30.436	40.469	41.814
28	463911	33.338	40.560	41.982
29	191107	30.939	40.258	41.432
30	400014	33.141	40.224	41.514
31	149472	30.828	40.008	41.085
32	353073	33.177	40.104	41.128
33	154408	31.752	40.039	41.065
34	309563	33.277	39.935	40.832
35	124125	31.851	39.783	40.643
36	923483	38.784	44.082	45.153
37	98892	32.999	41.200	42.045
38	225863	34.173	41.560	42.401
39	109034	33.617	41.868	42.691
40	263492	34.559	40.942	41.649
41	117478	33.649	41.007	41.648
42	274432	34.874	40.708	41.275
43	117069	33.827	40.529	41.017
44	268723	35.024	40.617	40.881
45	101275	33.812	40.196	40.394
46	272195	35.171	40.406	40.895
47	116073	34.221	40.200	40.349
48	749350	39.065	43.789	44.781
49	124113	34.962	41.100	41.221
50	290787	36.149	41.551	42.035
51	117235	35.004	41.365	41.725
52	302155	35.922	40.867	41.454
53	128247	34.756	40.501	40.909
54	301540	35.738	40.502	41.077

Nov 7 21:55 1991 Table_tennis_9Mbit/sec Page 2

55	102317	34.639	40.175	40.614
56	267222	35.391	40.116	40.630
57	121582	34.740	40.134	40.628
58	277318	35.224	39.964	40.603
59	109116	34.327	39.812	40.329
60	802634	39.096	43.185	43.973
61	146687	35.210	40.804	41.488
62	304116	36.132	41.211	41.969
63	103400	35.004	41.136	42.013
64	362678	36.301	40.937	41.653
65	142669	35.080	40.743	41.371
66	1265306	39.745	44.241	44.802
67	369849	36.767	41.124	41.736
68	745352	40.727	44.005	44.839
69	29206	34.710	42.800	42.885
70	100548	35.227	42.639	42.951
71	69098	36.120	42.710	43.040
72	1091320	39.382	43.922	44.685
73	143018	36.694	42.810	43.250
74	270933	36.186	42.366	43.403
75	115908	36.236	42.297	43.132
76	230472	35.783	42.174	43.115
77	106439	36.086	42.139	42.938
78	252743	36.116	42.123	43.004
79	96672	35.337	42.082	42.931
80	213969	36.834	41.897	42.729
81	100478	36.256	41.928	42.694
82	249086	36.136	42.031	42.894
83	101340	35.192	41.888	42.643
84	1153127	39.969	44.429	45.150
85	129598	36.817	42.828	43.709
86	248189	36.310	42.846	43.883
87	103834	36.348	42.840	43.686
88	237787	36.312	42.681	43.700
89	101320	36.378	42.600	43.409
90	233839	36.229	42.677	43.604
91	94902	36.608	42.658	43.492
92	238642	36.300	42.646	43.546
93	133699	36.736	42.624	43.424
94	237058	36.372	42.640	43.609
95	97658	36.409	42.538	43.271
96	1221118	39.610	43.960	44.939
97	122334	36.199	42.182	42.769
98	277789	36.101	42.303	43.307
99	79536	36.934	42.395	43.214
100	301691	36.856	41.794	42.726
101	157719	36.870	41.988	42.944
102	286086	36.735	41.374	42.201
103	79826	36.227	41.686	42.306
104	278876	36.667	41.060	41.863
105	110723	36.327	41.247	41.960
106	241714	36.239	40.694	41.405
107	116142	36.146	40.898	41.638
108	1218987	39.630	43.966	44.957
109	74927	36.783	42.337	43.310
110	197206	36.006	42.528	43.572

Nov 7 21:55 1991 Table_tennis_9Mbit/sec Page 3

111	111764	36.068	42.618	43.689
112	255018	36.967	42.187	43.262
113	77077	36.383	42.236	43.256
114	221256	36.441	41.835	42.814
115	108916	36.360	41.994	42.884
116	230086	36.319	41.643	42.497
117	86967	33.786	41.667	42.432
118	260620	36.270	41.263	42.098
119	113354	36.203	41.341	42.263
120	1223835	39.495	43.973	44.872
121	94998	36.272	42.049	42.962
122	218682	36.419	41.813	42.618
123	84835	36.420	42.400	43.280
124	368673	36.434	41.456	42.211
125	180827	33.732	41.030	41.746
126	306762	36.671	40.671	41.182
127	163313	33.925	40.766	41.367
128	400467	36.664	40.669	41.064
129	110412	33.008	40.179	40.454
130	1248188	39.669	44.066	45.060
131	97266	36.769	42.047	42.740
132	275660	36.608	42.230	43.181
133	136311	36.905	42.472	43.427
134	270841	36.808	40.951	41.610
135	101943	33.303	41.692	42.417
136	360616	36.468	40.788	41.314
137	135139	33.682	40.198	40.681
138	337530	36.432	40.734	41.318
139	81498	36.526	40.437	41.028
140	1233624	39.608	43.966	45.035
141	186122	36.833	41.426	42.364
142	303261	36.236	41.481	42.610
143	134832	33.764	42.035	43.048
144	317323	36.217	40.736	41.489
145	146384	36.552	41.184	41.913
146	428017	36.997	40.487	41.062
147	145173	33.561	40.619	41.296
148	378154	36.484	40.247	40.862
149	67031	31.393	40.233	40.614

Figure. 19

Nov 7 21:55 1991 Mobile_and_Calendar_4Mbit/sec Page 1

No.	bit	SNRY	SNRCb	SNRCr
0	704789	30.364	34.677	35.587
1	85896	26.128	33.309	33.969
2	69835	26.347	33.601	34.388
3	238310	27.901	33.667	34.503
4	69749	26.635	33.684	34.433
5	70330	26.404	33.428	34.261
6	193246	26.427	32.703	33.472
7	75012	25.961	33.013	33.697
8	74520	25.570	32.738	33.456
9	176973	25.869	32.107	32.959
10	70163	25.263	32.299	33.064
11	65656	25.108	32.181	33.019
12	621360	29.231	33.886	34.804
13	63047	25.932	32.562	33.323
14	63624	26.049	32.717	33.642
15	132878	25.879	32.854	33.762
16	56386	25.577	33.229	34.110
17	55040	26.400	33.294	34.126
18	155952	25.398	32.028	32.979
19	60000	25.964	32.311	33.204
20	56496	25.831	32.759	33.546
21	172014	26.312	31.327	32.452
22	59618	25.598	31.674	32.559
23	55136	25.376	32.169	32.991
24	632256	29.139	33.912	34.749
25	54071	26.614	32.701	33.523
26	50432	26.730	32.860	33.733
27	124310	27.642	33.064	33.749
28	49458	27.046	33.328	33.978
29	48088	26.483	33.508	34.170
30	143526	27.466	32.792	33.488
31	53282	25.763	32.786	33.273
32	51656	25.801	33.273	33.853
33	109094	26.512	32.401	33.124
34	55074	25.700	32.618	33.166
35	52496	25.491	32.613	33.432
36	605745	28.672	33.529	34.304
37	58782	26.474	32.449	33.065
38	55648	26.294	33.310	33.978
39	122614	26.759	32.603	33.194
40	55842	26.222	33.100	33.649
41	55560	26.001	33.105	33.532
42	170633	26.308	32.156	32.826
43	58727	25.193	32.203	32.569
44	55056	25.316	32.681	33.212
45	149644	25.610	31.657	32.366
46	57956	25.068	31.614	32.218
47	54920	24.936	32.325	32.974
48	607125	28.751	33.479	34.226
49	58114	25.516	32.321	32.690
50	54104	26.309	33.002	33.598
51	123188	26.806	32.650	33.188
52	58956	26.332	33.098	33.609
53	59960	26.079	33.137	33.601
54	157224	26.121	31.976	32.673

Nov 7 21:55 1991 Mobile_and_Calendar_4Mbit/sec Page 2

55	61984	25.313	32.192	32.689
56	61344	25.279	32.339	32.862
57	149575	25.411	31.334	32.030
58	60465	24.830	31.665	32.284
59	60050	24.493	31.706	32.305
60	611326	28.708	33.423	34.232
61	60550	25.216	31.639	32.048
62	56032	25.939	32.473	33.194
63	134111	26.074	32.472	33.124
64	56614	26.108	32.813	33.400
65	56446	26.623	32.682	33.238
66	181771	25.648	31.849	32.503
67	54405	25.120	32.152	32.696
68	64976	24.906	32.109	32.646
69	189407	26.322	31.354	32.125
70	62801	24.781	31.552	32.068
71	62824	24.632	31.570	32.086
72	616556	28.622	33.393	34.061
73	67547	24.813	31.500	32.221
74	56848	25.672	32.726	33.219
75	147176	25.821	32.213	32.737
76	67664	25.614	32.765	33.218
77	63056	25.097	32.458	32.883
78	161422	26.062	31.855	32.474
79	63842	24.934	32.124	32.622
80	60776	25.098	31.884	32.466
81	156971	25.032	31.453	32.151
82	60657	25.026	31.677	32.254
83	59880	25.020	31.616	32.242
84	619460	28.814	33.371	34.172
85	57915	26.048	32.190	32.793
86	58520	25.967	32.088	32.900
87	135684	26.982	32.409	33.125
88	57300	25.723	33.032	33.652
89	59944	26.123	32.632	33.312
90	185862	26.576	31.705	32.466
91	63434	25.270	32.045	32.677
92	61872	25.584	31.888	32.524
93	179470	26.268	31.311	32.049
94	66308	24.833	31.441	32.185
95	60950	25.414	31.475	32.103
96	623362	28.781	33.381	34.078
97	56301	25.926	31.973	32.604
98	55960	26.510	33.024	33.475
99	150959	27.018	33.061	33.694
100	57153	26.596	33.347	33.744
101	58192	26.422	33.027	33.510
102	171774	26.442	32.653	33.168
103	55338	25.963	32.653	33.204
104	56032	25.842	32.540	32.989
105	131462	26.434	31.808	32.369
106	55762	25.287	32.168	32.620
107	61608	25.630	32.013	32.461
108	623819	28.936	33.367	34.002
109	59020	26.050	32.042	32.333
110	56080	26.007	32.823	33.114

Nov 7 21:55 1991 Mobile_and_Calendar_4Mbit/sec Pag

111	143318	27.276	32.648	33.109
112	57930	26.064	33.113	33.421
113	62000	26.313	32.635	32.918
114	145102	27.362	32.326	32.874
115	60330	25.765	32.190	32.521
116	64840	26.863	32.148	32.590
117	130679	26.721	31.436	32.025
118	58273	26.208	31.611	32.044
119	64424	26.056	32.266	32.611
120	616562	29.227	33.368	34.164
121	60567	26.044	31.749	32.040
122	56328	26.661	32.373	32.728
123	131860	27.656	32.893	33.670
124	59203	26.686	32.881	33.309
125	60803	26.326	32.743	33.002
126	137625	27.127	32.266	32.963
127	59367	26.908	32.324	32.734
128	57848	26.050	32.149	32.679
129	126065	26.926	31.634	32.380
130	58685	26.093	32.029	32.621
131	60626	26.068	31.944	32.467
132	680720	29.012	32.989	33.906
133	58703	26.473	31.627	32.184
134	56264	27.027	31.972	32.681
135	111729	27.666	32.721	33.676
136	64828	27.308	32.322	33.094
137	63396	26.983	32.716	33.633
138	129945	26.592	32.028	32.726
139	66063	26.348	32.177	32.832
140	63848	26.186	32.024	32.670
141	603228	28.765	32.907	33.742
142	66076	26.229	32.273	32.864
143	64224	26.867	32.379	32.879
144	108403	27.360	32.055	32.786
145	66509	26.861	32.639	33.249
146	64040	26.292	32.678	33.139
147	123531	26.885	31.741	32.390
148	69298	25.828	31.763	32.104
149	74035	26.318	32.423	33.148

Nov 7 21:55 1991 Mobile_and_Calendar_9Mbit/sec Page 1

No.	bit	SNRY	SNRCb	SNRCr
0	1738563	38.072	42.036	42.253
1	247491	30.348	36.682	37.541
2	171698	29.562	36.721	37.497
3	478211	32.849	37.614	38.478
4	143366	30.076	37.031	37.760
5	168855	30.008	36.551	37.428
6	435014	31.604	36.166	37.141
7	149842	29.562	35.784	36.689
8	162924	29.359	35.534	36.498
9	398658	31.237	35.396	36.504
10	141233	29.285	35.203	36.085
11	149944	29.206	34.959	35.935
12	1615218	37.573	41.255	41.574
13	128541	29.839	36.011	36.838
14	145224	29.875	35.799	36.897
15	296015	31.741	36.931	37.879
16	109137	30.554	36.838	37.730
17	127064	30.772	37.204	37.895
18	362071	31.628	35.506	36.618
19	135833	30.326	35.716	36.643
20	120656	30.073	36.189	36.885
21	374371	31.444	34.857	35.949
22	110285	29.883	35.011	35.824
23	116980	29.669	35.376	36.171
24	1465787	36.770	40.304	40.801
25	96188	31.123	36.670	37.232
26	94528	30.638	35.845	36.689
27	278230	32.366	37.650	38.176
28	88354	31.388	37.029	37.686
29	95120	31.319	37.474	38.035
30	317053	32.614	37.112	37.766
31	137843	29.790	36.134	36.593
32	121616	30.186	36.985	37.500
33	251548	31.470	36.543	37.216
34	123236	29.644	36.166	36.611
35	128464	29.536	36.019	36.797
36	1473206	36.708	40.332	40.767
37	132401	29.740	36.012	36.660
38	126064	30.154	37.018	37.621
39	282511	31.658	37.266	37.920
40	125353	30.728	37.035	37.631
41	119988	30.238	36.955	37.444
42	361726	31.469	36.284	37.107
43	136078	29.792	35.611	36.274
44	119144	29.632	36.423	36.932
45	332842	30.774	35.396	36.356
46	131208	29.508	34.882	35.604
47	120222	29.534	35.811	36.379
48	1488726	36.824	40.292	40.676
49	108649	30.241	36.047	36.567
50	114344	30.571	36.731	37.274
51	275297	31.542	37.022	37.734
52	118919	30.547	36.839	37.460
53	117240	30.702	37.170	37.576
54	347147	31.311	35.871	36.753

Nov 7 21:55 1991 Mobile_and_Calendar_9Mbit/sec Page 2

55	143429	29.670	35.597	36.286
56	141592	29.760	35.815	36.474
57	322766	30.398	34.832	35.724
58	139754	29.211	34.907	35.694
59	128961	28.976	34.874	35.541
60	1364767	36.075	39.416	39.936
61	115802	29.618	35.162	35.784
62	119082	30.128	35.873	36.625
63	314262	30.794	36.064	36.920
64	118279	30.198	36.394	37.140
65	127167	29.857	35.874	36.610
66	407969	30.998	35.399	36.259
67	140823	29.210	35.068	35.842
68	126436	29.140	35.062	35.818
69	404302	30.643	34.860	35.809
70	131126	29.007	34.692	35.244
71	129640	28.935	34.536	35.338
72	1492543	36.766	40.245	40.689
73	144384	29.337	35.073	35.785
74	138853	29.637	36.072	36.640
75	359019	31.078	36.191	37.049
76	131418	29.697	36.354	36.923
77	138486	29.668	35.723	36.398
78	352682	31.483	36.647	36.416
79	150686	29.600	35.365	36.085
80	126816	29.609	35.169	35.918
81	361504	31.659	35.266	36.108
82	119568	29.560	34.866	35.600
83	116880	29.782	35.016	35.730
84	1496938	36.889	40.269	40.696
85	110437	30.278	36.074	36.631
86	150702	30.565	35.550	36.285
87	330876	32.503	36.838	37.644
88	140241	30.153	37.161	37.697
89	125661	30.504	36.489	37.205
90	404130	32.113	35.769	36.624
91	156746	30.035	35.616	36.380
92	134328	29.664	35.328	36.125
93	410715	31.992	35.373	36.200
94	135995	29.497	34.965	35.717
95	126208	29.765	34.944	35.671
96	1501774	36.870	40.275	40.705
97	102969	30.656	36.495	36.819
98	103264	31.366	37.294	37.594
99	326694	32.343	37.532	38.236
100	103526	31.314	37.538	38.001
101	118176	31.231	36.678	37.291
102	372370	32.050	36.616	37.313
103	107014	30.576	36.130	36.806
104	99912	30.696	35.788	36.484
105	335480	31.629	35.588	36.361
106	126584	30.104	35.432	36.058
107	115792	30.208	35.433	36.077
108	1504351	37.044	40.288	40.683
109	121048	30.272	35.794	36.379
110	127964	30.310	36.492	36.847

Nov 7 21:56 1991 Mobile_and_Calendar_9Mbit/sec Page

111	339616	32.619	36.916	37.614
112	117382	30.404	37.106	37.492
113	133814	30.453	36.684	37.105
114	322674	32.903	36.782	37.475
115	141120	30.136	35.624	36.212
116	146814	30.069	35.677	36.237
117	321356	31.916	36.466	36.198
118	118068	31.084	36.286	36.960
119	114008	30.886	36.852	36.426
120	1498069	37.277	40.431	40.794
121	121909	30.480	35.469	35.946
122	116360	31.028	36.899	36.576
123	282388	32.836	37.486	38.191
124	134505	30.628	36.458	37.119
125	123793	30.913	36.667	37.117
126	330474	32.282	36.237	37.168
127	116936	30.887	36.034	36.666
128	103073	30.821	35.789	36.666
129	306114	31.936	36.334	36.272
130	104482	30.687	36.627	36.373
131	116316	30.422	35.600	36.277
132	1372637	36.746	39.659	40.136
133	102618	30.690	35.872	36.488
134	118520	31.123	36.649	36.600
135	237244	32.742	37.133	37.833
136	108008	31.678	36.397	37.165
137	96528	31.643	37.109	37.766
138	316661	32.027	36.983	36.644
139	108419	30.938	36.107	36.748
140	94068	30.926	35.769	36.413
141	1407007	36.501	39.677	40.093
142	94423	30.877	36.403	37.021
143	103221	30.786	36.844	36.488
144	268423	32.465	36.453	37.166
145	120306	31.978	36.494	37.361
146	99300	31.709	36.909	37.816
147	345872	32.961	36.188	36.996
148	74117	29.219	34.863	35.369
149	88596	30.033	36.329	36.869

Figure. 21

Nov 7 21:11 1991 Flower_Garden_4Mbit/sec

sec.	bits
0.0	529230
0.4	2093873
0.8	3669329
1.2	5131695
1.6	6749725
2.0	8476190
2.4	10127335
2.8	11634332
3.2	13021668
3.6	14803106
4.0	16533469
4.4	18018235
4.8	19365868

Nov 7 21:09 1991 Table_tennis_4Mbit/sec

sec.	bits
0.0	459117
0.4	2139312
0.8	4117580
1.2	5867846
1.6	7053817
2.0	8395914
2.4	10110726
2.8	11439129
3.2	12843590
3.6	14432583
4.0	15774568
4.4	17464357
4.8	19163728

Nov 7 21:12 1991 Mobile_and_Calendar_4Mbit/sec

sec.	bits
0.0	704789
0.4	2515839
0.8	4078286
1.2	5475518
1.6	6978025
2.0	8494315
2.4	10100808
2.8	11696007
3.2	13306628
3.6	14840988
4.0	16348536
4.4	17797023
4.8	19192728

Nov 7 21:31 1991 Popple_9Mbit/sec

sec.	bits
0.0	791921
0.4	4377071
0.8	7927458
1.2	11435575
1.6	14892412
2.0	18516642
2.4	22329871
2.8	25590956
3.2	29194200
3.6	33324578
4.0	37423261
4.4	41583259
4.8	45156782

Nov 7 21:11 1991 Flower_Garden_9Mbit/sec

sec.	bits
0.0	907410
0.4	4571286
0.8	8475116
1.2	11991030
1.6	15757930
2.0	19796122
2.4	23600821
2.8	26778306
3.2	29764113
3.6	33588435
4.0	37433562
4.4	40955269
4.8	43903711

Nov 7 19:50 1991 Table_tennis_9Mbit/sec

sec.	bits
0.0	1127410
0.4	5055197
0.8	9565271
1.2	13647235
1.6	16361111
2.0	19305377
2.4	24030606
2.8	27064791
3.2	30142335
3.6	33415758
4.0	36376783
4.4	39928866
4.8	43518805

Nov 7 21:41 1991 Mobile_and_Calendar_9Mbit/sec

sec.	bits
0.0	1738563
0.4	6001017
0.8	9482969
1.2	12668355
1.6	16164618
2.0	19487483
2.4	23114914
2.8	26752148
3.2	30479959
3.6	33895991
4.0	37396924
4.4	40624911
4.8	43670251

Figure. 22

< Sequence : Flower garden 4Mbit/s >

1. S/N ratio

	all	I	P	B
snry	28.75	32.32	28.90	28.23
snrcb	32.74	34.35	32.42	32.65
snrcb	33.41	34.41	33.17	33.36

2. mean Quantizer Scale

	all	I	P	B
mstepg	18.67	8.23	14.41	21.61

3. Bit usage

	all	I	P	B
P_head	72	71	69	73
S_head	2203	2195	2198	2206
mb_inc	2317	2718	2592	2164
mb_type	6078	7276	7927	5238
quant	2170	4302	3153	1530
mvect	10592	6105	15862	9226
cbp	6966	7757	12214	4921
eob	6021	30704	7849	2136
intradc	3230	36333	326	0
coefy	73184	368542	113490	19874
coefu	6462	51910	6348	596
coefv	4400	40371	2990	245
others	7856	26829	11655	3984
total	131551	585113	186673	52193

< Sequence : Flower garden 9Mbit/s >

1. S/N ratio

	all	I	P	B
snry	33.50	39.70	33.99	32.52
snrcb	36.44	40.61	35.99	36.07
snrcb	36.54	40.25	36.03	36.25

2. mean Quantizer Scale

	all	I	P	B
mstepg	8.82	2.73	6.73	10.38

3. Bit usage

	all	I	P	B
P_head	72	71	69	73
S_head	2198	2195	2197	2198
mb_inc	2549	2719	2635	2495
mb_type	7995	7142	8735	7832
quant	2495	2497	2972	2318
mvect	13967	5346	16612	14109
cbp	10479	7448	14805	9272
eob	9622	34167	13066	5157
intradc	3888	40618	1468	8
coefy	183582	832473	284757	61792
coefu	25655	186246	31047	2783
coefv	24124	192642	24808	1963
others	11871	27737	16618	8054
total	298497	1341301	419789	118054

Figure. 23

< Sequence : Popple 4Mbit/s >

1. S/N ratio

	all	I	P	B
snry	34.04	37.98	34.68	32.83
snrcb	38.27	44.18	38.81	36.80
snrcb	38.35	43.80	38.79	37.04

2. mean Quantizer Scale

	all	I	P	B
mstepg	11.36	2.37	7.98	15.71

3. Bit usage

	all	I	P	B
P_head	71	71	69	73
S_head	2196	2191	2201	2194
mb_inc	2832	2695	2847	2843
mb_type	10287	7801	9043	11746
quant	6383	4420	7664	5664
mvect	21408	4747	22159	23676
cbp	8850	4080	10786	8075
eob	10087	37161	10227	5279
intradc	5719	43301	3954	664
coefy	147242	448018	185467	63508
coefu	37842	139351	46818	12827
coefv	38200	143030	45489	14004
others	18832	33728	21562	13993
total	309949	870594	368286	164546

Figure. 24

< Sequence : Table Tennis 4Mbit/s >

1. S/N ratio

	all	I	P	B
snry	30.97	31.97	31.16	30.63
snrcb	38.86	39.47	38.79	38.81
snrcb	39.02	40.04	38.93	38.93

2. mean Quantizer Scale

	all	I	P	B
mstepq	13.03	7.99	11.43	15.23

3. Bit usage

	all	I	P	B
P_head	71	71	70	73
S_head	2196	2202	2189	2201
mb_inc	2454	2408	2477	2444
mb_type	6503	4949	6752	6566
quant	1810	3242	2201	1238
mvect	8845	2819	9651	9223
cbp	5935	2928	8722	4151
eob	7070	35822	6947	2187
intradc	4497	38510	2338	386
coefy	77323	293688	94359	25737
coefu	2716	17271	2683	220
coefv	3981	20609	4231	892
others	7974	27688	9101	3626
total	131375	452207	151721	58944

< Sequence : Table Tennis 9Mbit/s >

1. S/N ratio

	all	I	P	B
snry	34.45	38.69	34.65	33.55
snrcb	41.45	43.54	41.27	41.23
snrcb	42.33	44.61	42.19	42.05

2. mean Quantizer Scale

	all	I	P	B
mstepq	6.86	2.47	5.81	8.49

3. Bit usage

	all	I	P	B
P_head	71	71	69	73
S_head	2196	2191	2195	2199
mb_inc	2432	2394	2416	2452
mb_type	7813	5140	7789	8295
quant	1995	2324	2281	1702
mvect	10942	2436	9983	13209
cbp	10510	3070	13067	9686
eob	11343	38703	12859	5348
intradc	5295	40406	4000	280
coefy	206689	785252	244293	75319
coefu	13410	104230	9459	934
coefv	12867	86717	10942	1657
others	12260	29017	14583	7434
total	297823	1101951	333936	128588

Figure. 25

< Sequence : Mobile and calendar 4Mbit/s >

1. S/N ratio

	all	I	P	B
snry	26.32	29.00	26.61	25.87
snrcb	32.47	33.51	32.19	32.44
snrcb	33.10	34.31	32.90	33.02

2. mean Quantizer Scale

	all	I	P	B
mstepq	27.44	11.42	24.02	30.78

3. Bit usage

	all	I	P	B
P_head	73	71	71	74
S_head	2200	2200	2198	2201
mb_inc	2335	2720	2591	2190
mb_type	5514	6637	6252	5094
quant	4104	12243	5536	2516
mvect	7582	4562	8717	7554
cbp	4474	6021	7738	3065
eob	4709	30463	4578	1409
intradc	3586	39549	621	8
coefy	65843	357758	80298	22546
coefu	5520	48514	3268	763
coefv	12740	57213	14588	6275
others	11720	52456	13097	5918
total	130400	620407	149553	59613

< Sequence : Mobile and calendar 9Mbit/s >

1. S/N ratio

	all	I	P	B
snry	31.20	36.93	31.84	30.22
snrcb	36.42	40.33	36.20	35.99
snrcb	37.11	40.75	37.02	36.66

2.mean Quantizer Scale

	all	I	P	B
mstepq	12.45	2.83	8.90	15.02

3. Bit usage

	all	I	P	B
P_head	73	71	70	74
S_head	2198	2193	2191	2201
mb_inc	2480	2588	2614	2416
mb_type	6818	5350	7425	6784
quant	6358	9958	8297	5172
mvect	8847	2644	8083	9936
cbp	8313	3660	11661	7679
eob	8388	35060	10858	4007
intradc	4833	48437	2477	36
coefy	173604	900296	214510	63999
coefu	25522	223953	19336	2014
coefv	29946	205120	27566	8053
others	18609	54484	23949	11973
total	295989	1493814	339037	124344

Figure. 26

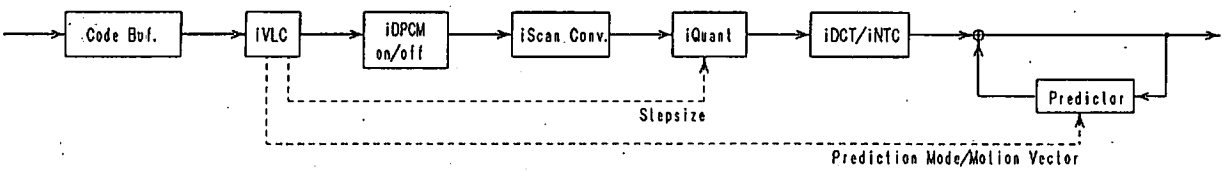


Figure27 Decoder Block Diagram

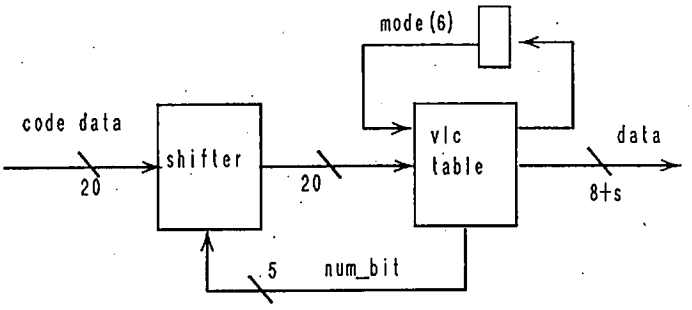


Figure28 Inverse VLC

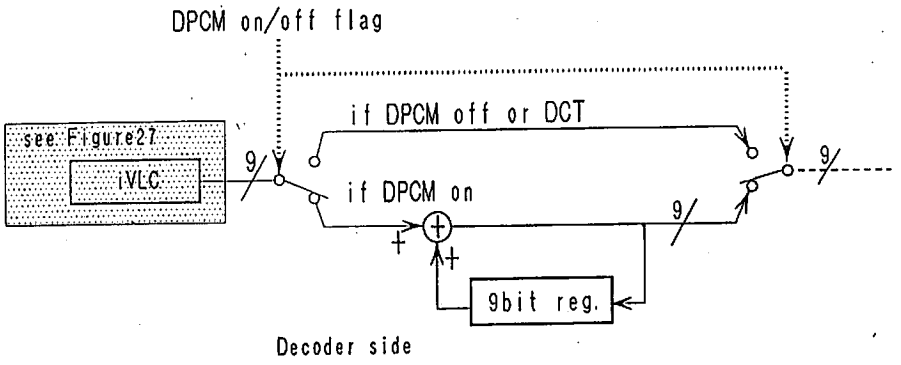
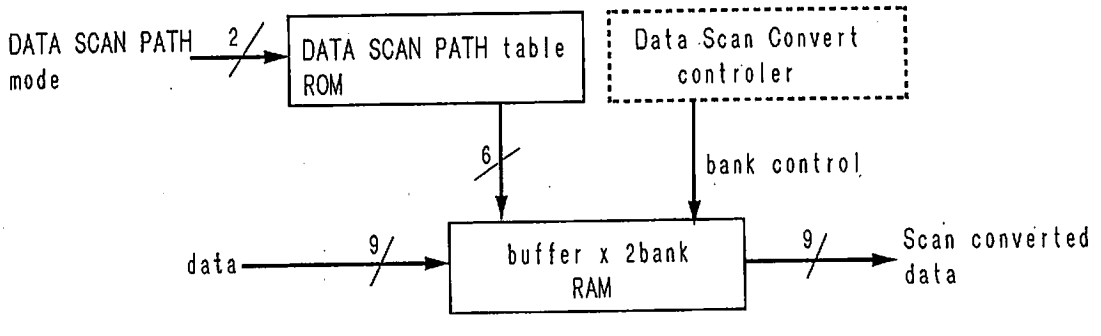


Figure29 NTC coef invers dpcm



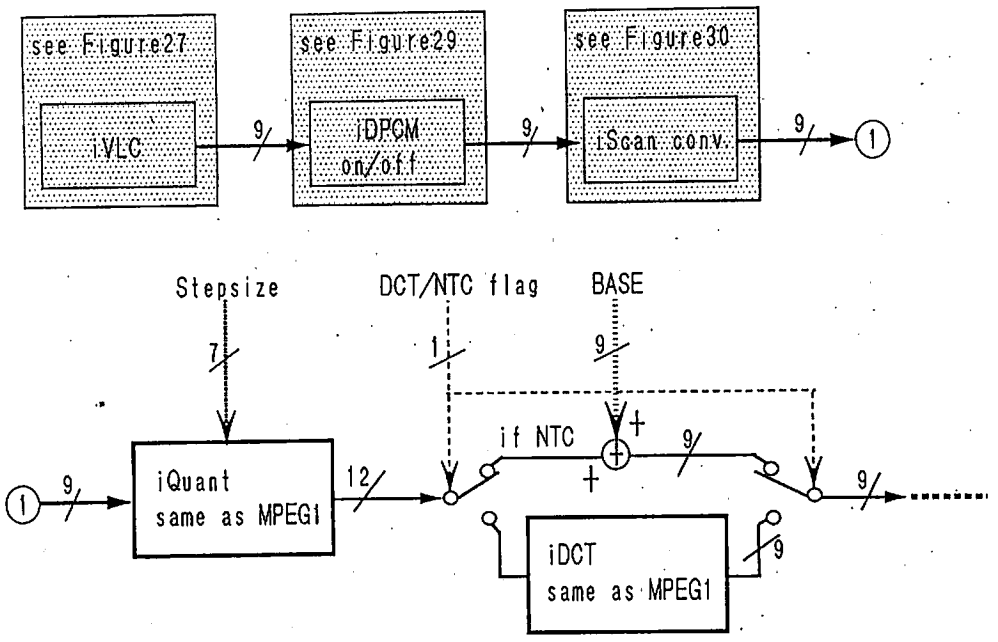


Figure31 iDCT/iNTC and iQuantizer

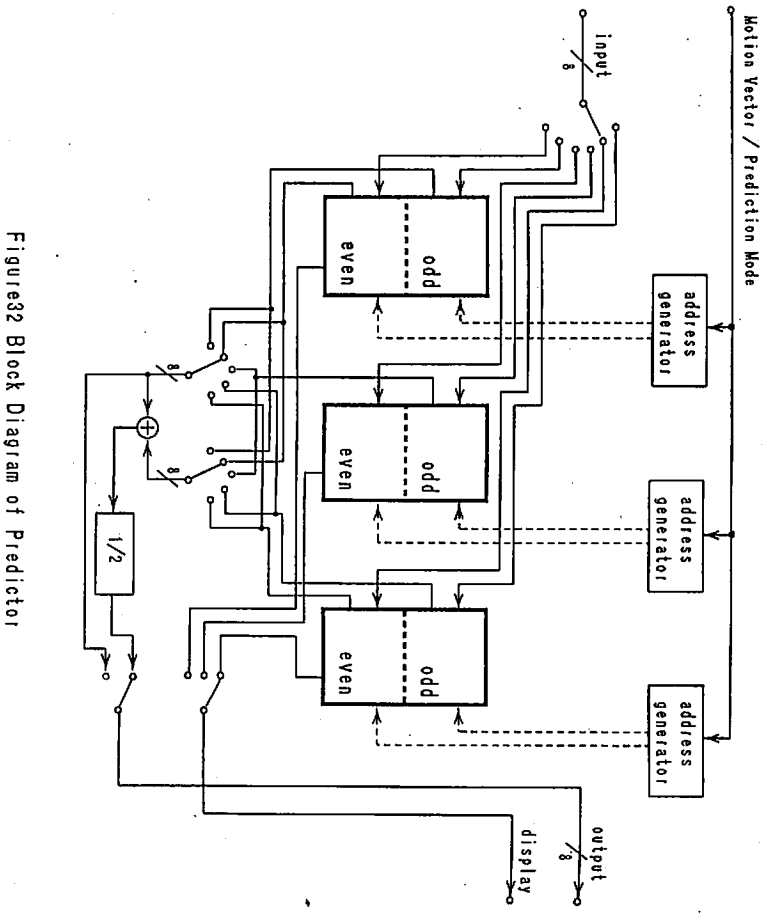


Figure32 Block Diagram of Predictor

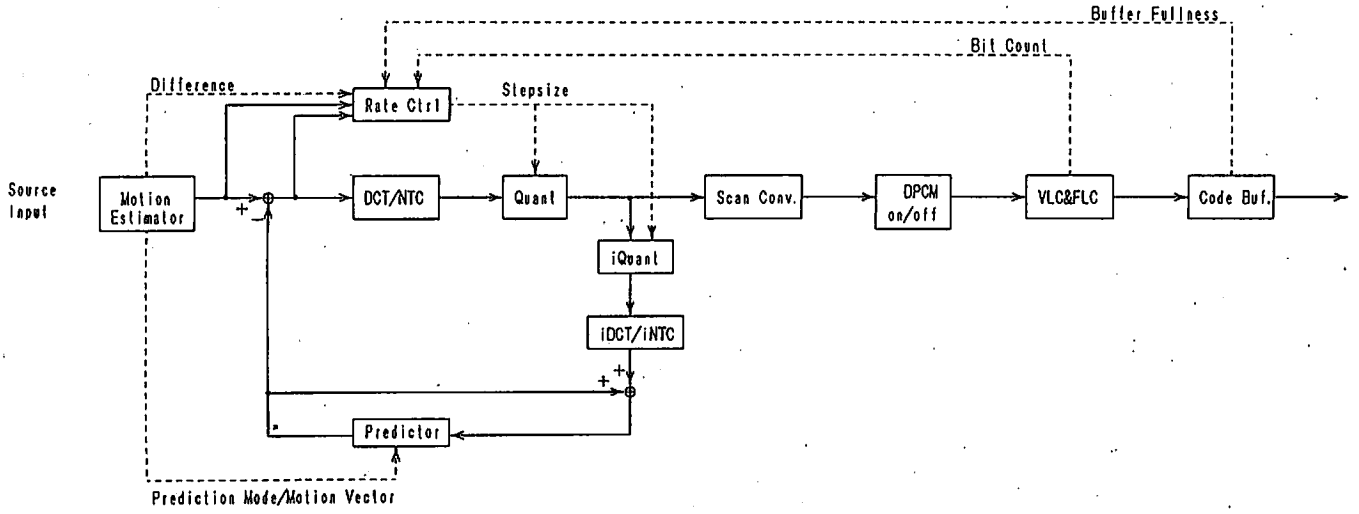


Figure33 Encoder Block Diagram

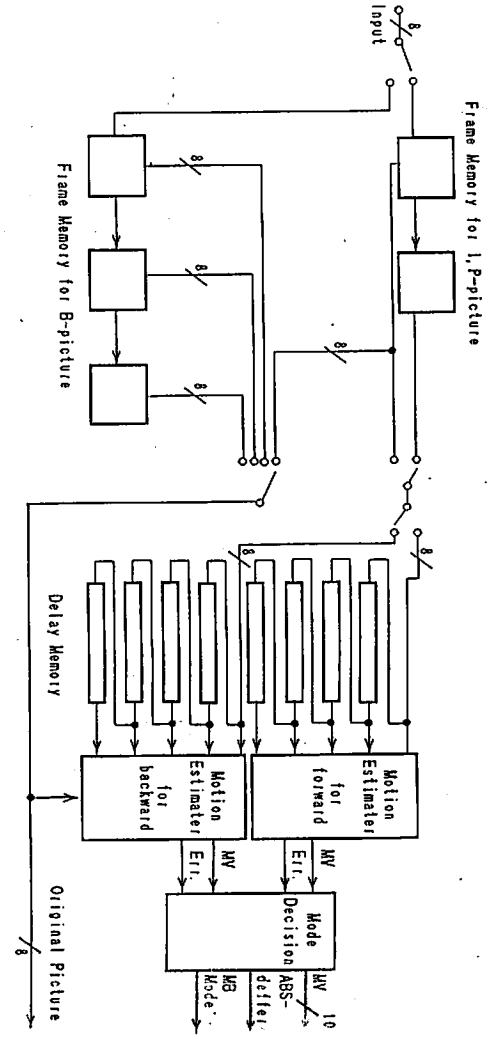


Figure34 Block Diagram of Picture Reorder and Motion Estimater

```
for( i=0 ; i<num_frame_mode ; i++ ){
    if( diff_tmp_frame > diff_frame[i] ){
        mode_tmp_frame = mode_frame[i];
        diff_tmp_frame = diff_frame[i];
    }
}

for( i=0 ; i<num_field_mode ; i++ ){
    if( diff_tmp_field > diff_field[i] ){
        mode_tmp_field = mode_field[i];
        diff_tmp_field = diff_field[i];
    }
}

if( diff_tmp_frame < diff_tmp_field ){
    return( mode_tmp_frame );
}else{
    return( mode_tmp_field );
}
```

Figure35 Function Description of MODE decision

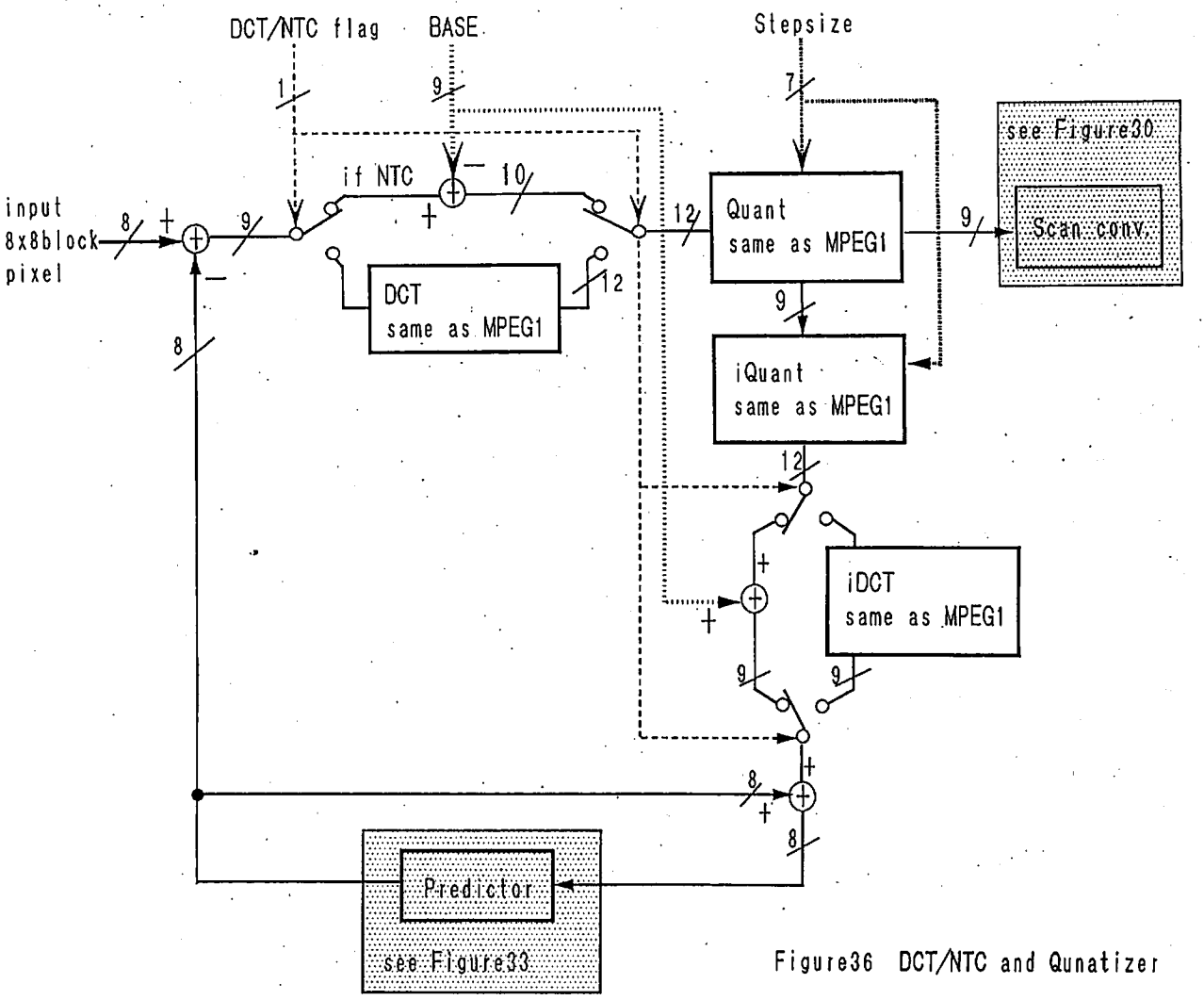


Figure36 DCT/NTC and Qunatizer

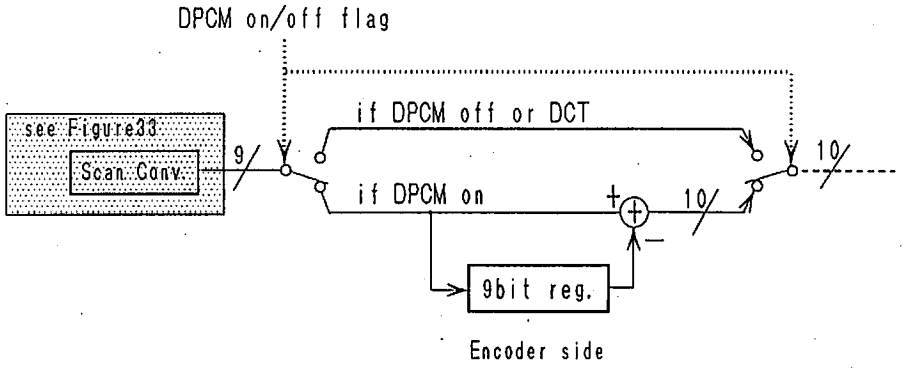


Figure37 NTC coef. dpcm

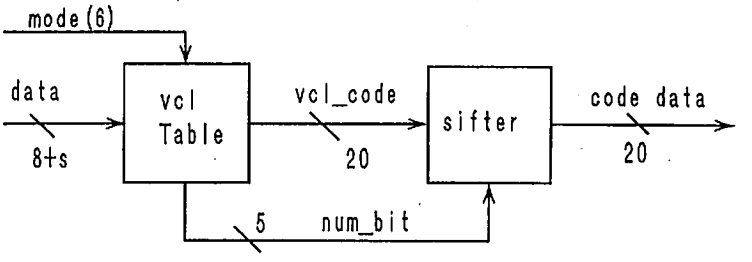


Figure38 VLC