Technical Specification Group Services and System Aspects **_TSGS#13(01)0452_**
Meeting #13, Beijing, China, 24-27 September 2001

Source: **TSG-SA WG4**

Title: CRs to TS 26.104 Corrections to encoder-decoder operations AMR-NB floating point (R99 and Release 4)

Document for: **Approval**

Agenda Item: **7.4.3**

The following CRs, agreed at the TSG-SA WG4 meeting #18, are presented to TSG SA #13 for approval.

Spec	CR	Rev	Phase	Subject	Cat	Vers	WG	Meeting	S4 doc
26.104	009	1	R99	Correction to make encoder and decoder memories independent	F	3.2.0	S4	TSG-SA WG4#18	S4-010410R
26.104	010	1	REL-4	Correction to make encoder and decoder memories independent	A	4.1.1	S4	TSG-SA WG4#18	S4-010410R
26.104	017		R99	Correction of decoder operation in error concealment of lost frames	F	3.2.0	S4	TSG-SA WG4#18	S4-010502
26.104	018		REL-4	Correction of decoder operation in error concealment of lost frames	A	4.1.1	S4	TSG-SA WG4#18	S4-010502

3GPP TSG-SA WG4 Meeting #18
Erlangen, Germany, 3 Sep – 7 Sep 2001

Tdoc S4 (01)0410R

CR-Form-v4

CHANGE REQUEST

⌘ **26.104**	**CR**	**009**	⌘ ev	**1**	⌘ Current version:	**3.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE **X** Radio Access Network ☐ Core Network **X**

Title:	⌘ Correction to make encoder and decoder memories independent		
Source:	⌘ TSG SA WG4		
Work item code:	⌘ Low bit rate codec for Multimedia Telephony	***Date:*** ⌘	24.9.2001
Category:	⌘ **F**	***Release:*** ⌘	R99

Use <u>one</u> of the following categories:
- ***F*** *(correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(addition of feature),*
- ***C*** *(functional modification of feature)*
- ***D*** *(editorial modification)*

Detailed explanations of the above categories can be found in 3GPP <u>TR 21.900</u>.

Use <u>one</u> of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘ Encoder module is dependent on decoder constant tables, some of the variables are unused

Summary of change: ⌘
1. Parameter sizes are moved from file rom_dec.h to interf_rom.h
   interf_rom.h is #included into rom_dec.h
2. gamma3_MR122 and gamma4 tables are combined into a single table in rom_dec.h
3. Remaining floating-point operations are converted to fixed-point operations in the decoder (sp_dec.c)
   struct: Cb_gain_averageState
   function: A_Refl, Log2, gc_pred, Cb_gain_average, sqrt_l_exp, Ex_ctrl, Inv_sqrt, agc2, agc, dtx_dec_activity_update

4. unused variables removed (sp_enc.c):
   Az_lsp
   search_2i40_9bits
   search_2i40_11bits
   search_8i40
   search_10i40

Consequences if not approved: ⌘ Encoder cannot be compiled without decoder files.
Same information is copied to multiple memory locations
Decoder can not be imported to fixed-point platforms without modifications

Clauses affected: ⌘ interf_rom.h, rom_dec.h, sp_dec.c, sp_enc.c

Other specs affected: ⌘
☐ Other core specifications ⌘
☐ Test specifications
☐ O&M Specifications

Other comments: ⌘	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

How the code is changed

1. Parameter sizes are moved from file rom_dec.h to interf_rom.h interf_rom.h is #included into rom_dec.h.

file interf_rom.h before the change (lines 28 to 37) (Blue lines are to be deleted)

```
/*
 * include files
 */
#include"typedef.h"
#include"rom_dec.h"

/*
 * definition of constants
 */
```

file interf_rom.h after the change (lines 28 to 45) (Red lines are to be inserted)

```
/*
 * include files
 */
#include"typedef.h"

/*
 * definition of constants
 */

/* number of parameters */
#define PRMNO_MR475 17
#define PRMNO_MR515 19
#define PRMNO_MR59  19
#define PRMNO_MR67  19
#define PRMNO_MR74  19
#define PRMNO_MR795 23
#define PRMNO_MR102 39
#define PRMNO_MR122 57
#define PRMNO_MRDTX 5
```

file interf_rom.h after the change (end of file) (Red lines are to be inserted)

```
/* parameter sizes (# of bits), one table per mode */
static const Word16 bitno_MR475[PRMNO_MR475] =
{
   8, 8, 7,    /* LSP VQ          */
   8, 7, 2, 8, /* first subframe  */
   4, 7, 2,    /* second subframe */
   4, 7, 2, 8, /* third subframe  */
   4, 7, 2     /* fourth subframe */
};
static const Word16 bitno_MR515[PRMNO_MR515] =
{
   8, 8, 7,    /* LSP VQ          */
   8, 7, 2, 6, /* first subframe  */
   4, 7, 2, 6, /* second subframe */
   4, 7, 2, 6, /* third subframe  */
   4, 7, 2, 6  /* fourth subframe */
};
static const Word16 bitno_MR59[PRMNO_MR59] =
{
   8, 9, 9,    /* LSP VQ          */
   8, 9, 2, 6, /* first subframe  */
   4, 9, 2, 6, /* second subframe */
   8, 9, 2, 6, /* third subframe  */
   4, 9, 2, 6  /* fourth subframe */
};
static const Word16 bitno_MR67[PRMNO_MR67] =
{
   8, 9, 9,       /* LSP VQ          */
   8, 11, 3, 7,   /* first subframe  */
   4, 11, 3, 7,   /* second subframe */
   8, 11, 3, 7,   /* third subframe  */
   4, 11, 3, 7    /* fourth subframe */
};
static const Word16 bitno_MR74[PRMNO_MR74] =
{
   8, 9, 9,        /* LSP VQ          */
   8, 13, 4, 7,    /* first subframe  */
   5, 13, 4, 7,    /* second subframe */
   8, 13, 4, 7,    /* third subframe  */
   5, 13, 4, 7     /* fourth subframe */
};
static const Word16 bitno_MR795[PRMNO_MR795] =
{
   9, 9, 9,           /* LSP VQ          */
   8, 13, 4, 4, 5,    /* first subframe  */
   6, 13, 4, 4, 5,    /* second subframe */
   8, 13, 4, 4, 5,    /* third subframe  */
   6, 13, 4, 4, 5     /* fourth subframe */
};
static const Word16 bitno_MR102[PRMNO_MR102] =
{
   8, 9, 9,                          /* LSP VQ          */
   8, 1, 1, 1, 1, 10, 10, 7, 7,  /* first subframe  */
   5, 1, 1, 1, 1, 10, 10, 7, 7,  /* second subframe */
   8, 1, 1, 1, 1, 10, 10, 7, 7,  /* third subframe  */
   5, 1, 1, 1, 1, 10, 10, 7, 7   /* fourth subframe */
};
static const Word16 bitno_MR122[PRMNO_MR122] =
{
   7, 8, 9, 8, 6,                             /* LSP VQ          */
   9, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 5,    /* first subframe  */
   6, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 5,    /* second subframe */
   9, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 5,    /* third subframe  */
   6, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 5     /* fourth subframe */
};
static const Word16 bitno_MRDTX[PRMNO_MRDTX] =
{
   3, 8, 9, 9, 6
};
```

file rom_dec.h before the change (lines 26 to 30)

```
/*
 * include files
 */
#include"typedef.h"
```

file rom_dec.h after the change (Red lines are to be inserted)

```
/*
 * include files
 */
#include"typedef.h"
#include"interf_rom.h"
```

file rom_dec.h before the change (lines 206 to 275) (Blue lines are to be deleted)

```
static const Word32 gamma4[M] =
{
   22938,
   16057,
   11240,
   7868,
   5508,
   3856,
   2699,
   1889,
   1322,
   925
};

/* parameter sizes (# of bits), one table per mode */
static const Word16 bitno_MR475[PRMNO_MR475] =
{
   8,
   8,
   7,
   /* LSP VQ          */ 8,
… … … …
… … … …
… … … …
};
static const Word16 bitno_MRDTX[PRMNO_MRDTX] =
{
   3,
   8,
   9,
   9,
   6
};

/* adaptive codebook gain quantization table (MR122, MR795) */
#define NB_QUA_PITCH 16
static const Word32 qua_gain_pitch[NB_QUA_PITCH] =
```

file rom_dec.h after the change

```
static const Word32 gamma4[M] =
{
   22938,
   16057,
   11240,
   7868,
   5508,
   3856,
   2699,
   1889,
   1322,
   925
};

/* adaptive codebook gain quantization table (MR122, MR795) */
#define NB_QUA_PITCH 16
static const Word32 qua_gain_pitch[NB_QUA_PITCH] =
```

2. gamma3_MR122 and gamma4 tables are combined into a single table in rom_dec.h

file rom_dec.h before the change (lines 167 to 179)

```
 static const Word32 gamma3_MR122[M] =
{
   22938,
   16057,
   11240,
   7868,
   5508,
   3856,
   2699,
   1889,
   1322,
   925
};
```

file rom_dec.h before the change (lines 206 to 218) (Blue lines are to be deleted)

```
static const Word32 gamma4[M] =
{
   22938,
   16057,
   11240,
   7868,
   5508,
   3856,
   2699,
   1889,
   1322,
   925
};
```

file rom_dec.h after the change

```
static const Word32 gamma4_gamma3_MR122[M] =
{
  22938,
  16057,
  11240,
  7868,
  5508,
  3856,
  2699,
  1889,
  1322,
  925
};
```

file sp_dec.c before the change (lines 5390 to 5409)

```
static void Post_Filter( Post_FilterState *st, enum Mode mode, Word32 *syn,
      Word32 *Az_4 )
{
   Word32 h[22], Ap3[MP1], Ap4[MP1];   /* bandwidth expanded LP parameters */
   Word32 tmp, i_subfr, i, temp1, temp2, overflow = 0;
   Word32 *Az, *p1, *p2, *syn_work = &st->synth_buf[M];
   const Word32 *pgamma3 = &gamma3[0];
   const Word32 *pgamma4 = &gamma4[0];


   /*
    * Post filtering
    */
   memcpy( syn_work, syn, L_FRAME <<2 );
   Az = Az_4;

   if ( ( mode == MR122 ) || ( mode == MR102 ) ) {
      pgamma3 = &gamma3_MR122[0];
      pgamma4 = &gamma4_MR122[0];
   }
```

file sp_dec.c after the change

```
static void Post_Filter( Post_FilterState *st, enum Mode mode, Word32 *syn,
      Word32 *Az_4 )
{
   Word32 h[22], Ap3[MP1], Ap4[MP1];   /* bandwidth expanded LP parameters */
   Word32 tmp, i_subfr, i, temp1, temp2, overflow = 0;
   Word32 *Az, *p1, *p2, *syn_work = &st->synth_buf[M];
   const Word32 *pgamma3 = &gamma3[0];
   const Word32 *pgamma4 = &gamma4_gamma3_MR122[0];


   /*
    * Post filtering
    */
   memcpy( syn_work, syn, L_FRAME <<2 );
   Az = Az_4;

   if ( ( mode == MR122 ) || ( mode == MR102 ) ) {
      pgamma3 = &gamma4_gamma3_MR122[0];
      pgamma4 = &gamma4_MR122[0];
   }
```

3. floating-point operations -> fixed-point operations

file sp_dec.c before the change (lines 55 to 62)

```
typedef struct
{
   Float32 hangCount;   /* counter; */
   /* history vector of past synthesis speech energy */
   Word32 cbGainHistory[L_CBGAINHIST];
   Word16 hangVar;   /* counter; */

}Cb_gain_averageState;
```

file sp_dec.c after the change

```
typedef struct
{
   Word32 hangCount;   /* counter; */
   /* history vector of past synthesis speech energy */
   Word32 cbGainHistory[L_CBGAINHIST];
   Word16 hangVar;   /* counter; */

}Cb_gain_averageState;
```

file sp_dec.c before the change (lines 974 to 986)

```
   /* backward Levinson recursion */
   for ( i = M - 1; i >= 0; i-- ) {
      if ( labs( aState[i] ) >= 4096 ) {
         goto ExitRefl;
      }
      refl[i] = aState[i] << 3;
      temp = ( refl[i] * refl[i] ) << 1;
      acc = ( MAX_32 - temp );
      frexp( ( Float64 )acc, &normShift );
      normShift = 31 - normShift;
      scale = 15 - normShift;
      acc = ( acc << normShift );
      temp = ( acc + ( Word32 )0x00008000L );
```

file sp_dec.c after the change

```
   /* backward Levinson recursion */
   for ( i = M - 1; i >= 0; i-- ) {
      if ( labs( aState[i] ) >= 4096 ) {
         goto ExitRefl;
      }
      refl[i] = aState[i] << 3;
      temp = ( refl[i] * refl[i] ) << 1;
      acc = ( MAX_32 - temp );
      normShift=0;
      if (acc != 0){
         temp = acc;
         while (!(temp & 0x40000000))
         {
            normShift++;
            temp = temp << 1;
         }
      }
      else{
         normShift = 0;
      }
      scale = 15 - normShift;
      acc = ( acc << normShift );
      temp = ( acc + ( Word32 )0x00008000L );
```

file sp_dec.c before the change (lines 1097 to 1105)

```
static void Log2( Word32 x, Word32 *exponent, Word32 *fraction )
{
   int exp;


   frexp( ( Float64 )x, &exp );
   exp = 31 - exp;
   Log2_norm( x <<exp, exp, exponent, fraction );
}
```

file sp_dec.c after the change

```
static void Log2( Word32 x, Word32 *exponent, Word32 *fraction )
{
   int tmp, exp=0;

   if (x != 0){
         tmp = x;
         while (!((tmp & 0x80000000) ^ ((tmp & 0x40000000) << 1)))
         {
            exp++;
            tmp = tmp << 1;
         }
      }
   Log2_norm( x <<exp, exp, exponent, fraction );
}
```

file sp_dec.c before the change (lines 3097 to 3103)

```
        /*
         * Compute: meansEner - 10log10(ener_code/ LSufr)
         */
      frexp( ( Float64 )ener_code, &exp_code );
      exp_code = 31 - exp_code;
      ener_code <<= exp_code;
```

file sp_dec.c after the change

```
        /*
         * Compute: meansEner - 10log10(ener_code/ LSufr)
         */
      exp_code=0;
      if (ener_code != 0){
         while (!(ener_code & 0x40000000))
         {
            exp_code++;
            ener_code = ener_code << 1;
         }
      }
```

file sp_dec.c before the change (lines 3674 to 3685)

```
   /* compute lsp difference */
   for ( i = 0; i < M; i++ ) {
      tmp1 = labs( lspAver[i]- lsp[i] );
      frexp( ( Float64 )tmp1, &shift1 );
      frexp( ( Float64 )lspAver[i], &shift2 );
      tmp2 = lspAver[i];
      shift2 = 15 - shift2;
      tmp2 <<= shift2;
      shift1 = 14 - shift1;
      tmp1 = tmp1 << shift1;
      tmp[i] = ( tmp1 << 15 ) / tmp2;
      shift = 2 + shift1 - shift2;
```

file sp_dec.c after the change

```
   /* compute lsp difference */
   for ( i = 0; i < M; i++ ) {
      tmp1 = labs( lspAver[i]- lsp[i] );
      shift1 = 0;
      if (tmp1 != 0){
         while (!(tmp1 & 0x2000))
         {
            shift1++;
            tmp1 = tmp1 << 1;
         }
      }
      tmp2 = lspAver[i];
      shift2 = 0;
      if (tmp2 != 0){
         while (!(tmp2 & 0x4000))
         {
            shift2++;
            tmp2 = tmp2 << 1;
         }
      }
      tmp[i] = ( tmp1 << 15 ) / tmp2;
      shift = 2 + shift1 - shift2;
```

file sp_dec.c before the change (lines 3789 to 3790)

```
   st->hangCount += 1;
   return cbGainMix;
```

file sp_dec.c after the change

```
   st->hangCount += 1;
   if (st->hangCount & 0x80000000)
      st->hangCount = 40;
   return cbGainMix;
```

file sp_dec.c before the change (lines 4008 to 4031)

```
static Word32 sqrt_l_exp( Word32 x, Word32 *exp )
{
   Word32 y, a, i, tmp;
   int e;


   if ( x <= ( Word32 )0 ) {
      *exp = 0;
      return( Word32 )0;
   }
   frexp( ( Float64 )x, &e );
   e = ( 31 - e ) & 0xFFFE;
   x = ( x << e );
   *exp = ( Word16 )e;
   x = ( x >> 9 );
   i = ( Word16 )( x >> 16 );
   x = ( x >> 1 );
   a = x & ( Word16 )0x7fff;
   i = ( i - 16 );
   y = ( sqrt_table[i] << 16 );
   tmp = ( sqrt_table[i] - sqrt_table[i + 1] );
   y -= ( tmp * a ) << 1;
   return( y );
}
```

file sp_dec.c after the change

```
static Word32 sqrt_l_exp( Word32 x, Word32 *exp )
{
   Word32 y, a, i, tmp;
   int e;


   if ( x <= ( Word32 )0 ) {
      *exp = 0;
      return( Word32 )0;
   }
   e=0;
   if (x != 0){
      tmp = x;
      while (!(tmp & 0x40000000))
      {
         e++;
         tmp = tmp << 1;
      }
   }
   e = e & 0xFFFE;
   x = ( x << e );
   *exp = ( Word16 )e;
   x = ( x >> 9 );
   i = ( Word16 )( x >> 16 );
   x = ( x >> 1 );
   a = x & ( Word16 )0x7fff;
   i = ( i - 16 );
   y = ( sqrt_table[i] << 16 );
   tmp = ( sqrt_table[i] - sqrt_table[i + 1] );
   y -= ( tmp * a ) << 1;
   return( y );
}
```

file sp_dec.c before the change (lines 4082 to 4088)

```
      /* scaleFactor=avgEnergy/excEnergy in Q0 */
      frexp( ( Float64 )excEnergy, &exp );
      exp = 15 - exp;
      excEnergy = excEnergy << exp;
      excEnergy = 536838144 / excEnergy;
      T0 = ( avgEnergy * excEnergy ) << 1;
      T0 = ( T0 >> ( 20 - exp ) );
```

file sp_dec.c after the change

```
      /* scaleFactor=avgEnergy/excEnergy in Q0 */
      exp=0;
      if (excEnergy != 0){
         while (!(excEnergy & 0x4000))
         {
            exp++;
            excEnergy = excEnergy << 1;
         }
      }
      excEnergy = 536838144 / excEnergy;
      T0 = ( avgEnergy * excEnergy ) << 1;
      T0 = ( T0 >> ( 20 - exp ) );
```

file sp_dec.c before the change (lines 4125 to 4138)

```
static Word32 Inv_sqrt( Word32 x )
{
   int i, a, tmp, exp;
   Word32 y;


   if ( x <= ( Word32 )0 )
      return( ( Word32 )0x3fffffffL );
   frexp( ( Float64 )x, &exp );
   exp = 31 - exp;

   /* x is normalized */
   x = ( x << exp );
   exp = ( 30 - exp );
```

file sp_dec.c after the change

```
static Word32 Inv_sqrt( Word32 x )
{
   int i, a, tmp, exp;
   Word32 y;


   if ( x <= ( Word32 )0 )
      return( ( Word32 )0x3fffffffL );
   exp=0;
   while (!(x & 0x40000000))
   {
      exp++;
      x = x << 1;
   }

   /* x is normalized */
   exp = ( 30 - exp );
```

file sp_dec.c before the change (lines 4270 to 4296)

```
static void agc2( Word32 *sig_in, Word32 *sig_out )
{
   Word32 s;
   int i, exp;
   Word16 gain_in, gain_out, g0;


   /* calculate gain_out with exponent */
   s = energy_new( sig_out );

   if ( s == 0 ) {
      return;
   }
   frexp( ( Float64 )s, &exp );
   exp = 30 - exp;
   gain_out = ( Word16 )( ( ( s << exp ) + 0x00008000L ) >> 16 );

   /* calculate gain_in with exponent */
   s = energy_new( sig_in );

   if ( s == 0 ) {
      g0 = 0;
   }
   else {
      frexp( ( Float64 )s, &i );
      i = 31 - i;
      s <<= i;

      if ( s < 0x7fff7fff )
         gain_in = ( Word16 )( ( s + 0x00008000L ) >> 16 );
      else
         gain_in = 32767;
      exp = ( exp - i );
```

file sp_dec.c after the change

```
static void agc2( Word32 *sig_in, Word32 *sig_out )
{
   Word32 s;
   int i, exp;
   Word16 gain_in, gain_out, g0;


   /* calculate gain_out with exponent */
   s = energy_new( sig_out );

   if ( s == 0 ) {
      return;
   }
   exp=0;
   while (!(s & 0x20000000))
   {
      exp++;
      s = s << 1;
   }

   gain_out = ( Word16 )( ( s + 0x00008000L ) >> 16 );

   /* calculate gain_in with exponent */
   s = energy_new( sig_in );

   if ( s == 0 ) {
      g0 = 0;
   }
   else {
      i = 0;
      while (!(s & 0x40000000))
      {
         i++;
         s = s << 1;
      }

      if ( s < 0x7fff7fff )
         gain_in = ( Word16 )( ( s + 0x00008000L ) >> 16 );
      else
         gain_in = 32767;
      exp = ( exp - i );
```

file sp_dec.c before the change (lines 4487 to 4511)

```
static void dtx_dec_activity_update( dtx_decState *st, Word32 lsf[], Word32
      frame[] )
{
   Float64 frame_en;
   Word32 log_en_e, log_en_m, log_en, i;


   /* update lsp history */
   st->lsf_hist_ptr += M;

   if ( st->lsf_hist_ptr == 80 ) {
      st->lsf_hist_ptr = 0;
   }
   memcpy( &st->lsf_hist[st->lsf_hist_ptr], lsf, M <<2 );

   /* compute log energy based on frame energy */
   frame_en = 0;   /* Q0 */

   for ( i = 0; i < L_FRAME; i ++ ) {
      frame_en += frame[i] * frame[i];
   }

   log_en = ( frame_en > 0x3fffffff ) ? 0x7FFFFFFE: (Word32)frame_en << 1;

   Log2( log_en , &log_en_e, &log_en_m );
```

file sp_dec.c after the change

```
static void dtx_dec_activity_update( dtx_decState *st, Word32 lsf[], Word32
      frame[] )
{
   Word32 frame_en;
   Word32 log_en_e, log_en_m, log_en, i;


   /* update lsp history */
   st->lsf_hist_ptr += M;

   if ( st->lsf_hist_ptr == 80 ) {
      st->lsf_hist_ptr = 0;
   }
   memcpy( &st->lsf_hist[st->lsf_hist_ptr], lsf, M <<2 );

   /* compute log energy based on frame energy */
   frame_en = 0;   /* Q0 */

   for ( i = 0; (i < L_FRAME); i ++ ) {
      frame_en += frame[i] * frame[i];
      if (frame_en & 0x80000000)
         break;
   }

   log_en = (frame_en & 0xC0000000) ? 0x7FFFFFFE: (Word32)frame_en << 1;

   Log2( log_en , &log_en_e, &log_en_m );
```

file sp_dec.c before the change (lines 5282 to 5312)

```
static void agc( agcState *st, Word32 *sig_in, Word32 *sig_out, Word16 agc_fac )
{
   Word32 s, gain_in, gain_out, g0, gain;
   int exp, i;


   /* calculate gain_out with exponent */
   s = energy_new( sig_out );

   if ( s == 0 ) {
      st->past_gain = 0;
      return;
   }
   frexp( ( Float32 )s, &exp );
   exp = 30 - exp;

   if ( exp >= 0 )
      gain_out = ( ( s << exp ) + 0x00008000L ) >> 16;
   else
      gain_out = ( ( s >> abs( exp ) ) + 0x00008000L ) >> 16;

   /* calculate gain_in with exponent */
   s = energy_new( sig_in );

   if ( s == 0 ) {
      g0 = 0;
   }
   else {
      frexp( ( Float32 )s, &i );
      i = 31 - i;
      s = ( s << i ) + 0x00008000L;
```

file sp_dec.c after the change

```
static void agc( agcState *st, Word32 *sig_in, Word32 *sig_out, Word16 agc_fac )
{
   Word32 s, gain_in, gain_out, g0, gain;
   int exp, i;


   /* calculate gain_out with exponent */
   s = energy_new( sig_out );

   if ( s == 0 ) {
      st->past_gain = 0;
      return;
   }
   exp=0;
   i = s;
   while (!(i & 0x40000000))
   {
      exp++;
      i = i << 1;
   }
   exp -=1;
   if (exp & 0x80000000) {
      s >>= 1;
```

```
}
else {
   s <<= exp;
}
gain_out = ( s + 0x00008000L ) >> 16;

/* calculate gain_in with exponent */
s = energy_new( sig_in );

if ( s == 0 ) {
   g0 = 0;
}
else {
   i=0;
while (!(s & 0x40000000))
{
   i++;
   s = s << 1;
}
   s = s + 0x00008000L;
```

4. unused variables removed (sp_enc.c):

file sp_enc.c before the change (lines 617 to 623)

```
static void Az_lsp( Float32 a[], Float32 lsp[], Float32 old_lsp[] )
{
   Word32 i, j, nf, ip;
   Float32 xlow, ylow, xhigh, yhigh, xmid, ymid, xint;
   Float32 x, y;
   Float32 *coef;
   Float32 f1[6], f2[6];
```

file sp_enc.c after the change

```
static void Az_lsp( Float32 a[], Float32 lsp[], Float32 old_lsp[] )
{
   Word32 i, j, nf, ip;
   Float32 xlow, ylow, xhigh, yhigh, xmid, ymid, xint;
   Float32 y;
   Float32 *coef;
   Float32 f1[6], f2[6];
```

file sp_enc.c before the change (lines 672 to 678)

```
         /*
          * Linear interpolation
          * xint = xlow - ylow*(xhigh-xlow)/(yhigh-ylow)
          */
         x = xhigh - xlow;
         y = yhigh - ylow;
```

file sp_enc.c after the change

```
         /*
          * Linear interpolation
          * xint = xlow - ylow*(xhigh-xlow)/(yhigh-ylow)
          */
         y = yhigh - ylow;
```

file sp_enc.c before the change (lines 4503 to 4558)

```
static void search_2i40_9bits( Word16 subNr, Float32 dn[], Float32 rr[][L_CODE],
      Word32 codvec[] )
{
   Float32 ps, ps0, ps1, psk, alp, alp0, alp1, alpk, sq, sq1;
   Word32 i0, i1, ix, i;
   Word16 ipos[2];
   Word16 track1;


   psk = -1;
   alpk = 1;

   for ( i = 0; i < 2; i++ ) {
      codvec[i] = i;
   }

   /* main loop: try 2x4  tracks */
   for ( track1 = 0; track1 < 2; track1++ ) {
      ipos[0] = startPos[( subNr <<1 )+( track1 << 3 )];
      ipos[1] = startPos[( subNr <<1 )+1 + ( track1 << 3 )];

      /* i0 loop: try 8 positions   */
      for ( i0 = ipos[0]; i0 < L_CODE; i0 += STEP ) {
         ps0 = dn[i0];
         alp0 = rr[i0][i0];

         /* i1 loop: 8 positions */
         sq = -1;
         alp = 1;
         ps = 0;
         ix = ipos[1];

         for ( i1 = ipos[1]; i1 < L_CODE; i1 += STEP ) {
            ps1 = ps0 + dn[i1];
            alp1 = alp0 + rr[i1][i1] + 2.0F * rr[i0][i1];
            sq1 = ps1 * ps1;

            if ( ( alp * sq1 ) > ( sq * alp1 ) ) {
               sq = sq1;
               ps = ps1;
               alp = alp1;
               ix = i1;
            }
         }

         /* memorise codevector if this one is better than the last one   */
         if ( ( alpk * sq ) > ( psk * alp ) ) {
            psk = sq;
            alpk = alp;
            codvec[0] = i0;
            codvec[1] = ix;
         }
      }
   }
   return;
}
```

file sp_enc.c after the change

```
static void search_2i40_9bits( Word16 subNr, Float32 dn[], Float32 rr[][L_CODE],
      Word32 codvec[] )
{
   Float32 ps0, ps1, psk, alp, alp0, alp1, alpk, sq, sq1;
   Word32 i0, i1, ix, i;
   Word16 ipos[2];
   Word16 track1;


   psk = -1;
   alpk = 1;

   for ( i = 0; i < 2; i++ ) {
      codvec[i] = i;
   }

   /* main loop: try 2x4  tracks */
   for ( track1 = 0; track1 < 2; track1++ ) {
      ipos[0] = startPos[( subNr << 1 ) + ( track1 << 3 )];
      ipos[1] = startPos[( subNr << 1 ) + 1 + ( track1 << 3 )];

      /* i0 loop: try 8 positions   */
      for ( i0 = ipos[0]; i0 < L_CODE; i0 += STEP ) {
         ps0 = dn[i0];
         alp0 = rr[i0][i0];

         /* i1 loop: 8 positions */
         sq = -1;
         alp = 1;
         ix = ipos[1];

         for ( i1 = ipos[1]; i1 < L_CODE; i1 += STEP ) {
            ps1 = ps0 + dn[i1];
            alp1 = alp0 + rr[i1][i1] + 2.0F * rr[i0][i1];
            sq1 = ps1 * ps1;

            if ( ( alp * sq1 ) > ( sq * alp1 ) ) {
               sq = sq1;
               alp = alp1;
               ix = i1;
            }
         }

         /* memorise codevector if this one is better than the last one   */
         if ( ( alpk * sq ) > ( psk * alp ) ) {
            psk = sq;
            alpk = alp;
            codvec[0] = i0;
            codvec[1] = ix;
         }
      }
   }
   return;
}
```

file sp_enc.c before the change (lines 4732 to 4802)

```
static void search_2i40_11bits( Float32 dn[], Float32 rr[][L_CODE], Word32
      codvec[] )
{
   Float64 alpk, alp, alp0, alp1;
   Float32 ps, psk, ps0, ps1, sq, sq1;
   Word32 i, i0, i1, ix = 0;
   Word16 ipos[2];
   Word16 track1, track2;
   psk = -1;
   alpk = 1;

   for ( i = 0; i < 2; i++ ) {
      codvec[i] = i;
   }

   /*
    * main loop: try 2x4  tracks.
    */
   for ( track1 = 0; track1 < 2; track1++ ) {
      for ( track2 = 0; track2 < 4; track2++ ) {
         /* fix starting position */
         ipos[0] = startPos1[track1];
         ipos[1] = startPos2[track2];

         /*
          * i0 loop: try 8 positions.
          */
         for ( i0 = ipos[0]; i0 < L_CODE; i0 += STEP ) {
            ps0 = dn[i0];
            alp0 = rr[i0][i0] * 0.25F;

            /*
             * i1 loop: 8 positions.
             */
            sq = -1;
            alp = 1;
            ps = 0;
            ix = ipos[1];

            for ( i1 = ipos[1]; i1 < L_CODE; i1 += STEP ) {
               ps1 = ps0 + dn[i1];

               /* alp1 = alp0 + rr[i0][i1] + 1/2*rr[i1][i1]; */
               alp1 = alp0 + rr[i1][i1] * 0.25F;
               alp1 += rr[i0][i1] * 0.5F;
               sq1 = ps1 * ps1;

               if ( ( alp * sq1 ) > ( sq * alp1 ) ) {
                  sq = sq1;
                  ps = ps1;
                  alp = alp1;
                  ix = i1;
               }
            }

            /*
             * memorise codevector if this one is better than the last one.
             */
            if ( ( alpk * sq ) > ( psk * alp ) ) {
               psk = sq;
               alpk = alp;
               codvec[0] = i0;
               codvec[1] = ix;
            }
         }
      }
   }
   return;
}
```

file sp_enc.c after the change

```
static void search_2i40_11bits( Float32 dn[], Float32 rr[][L_CODE], Word32
      codvec[] )
{
   Float64 alpk, alp, alp0, alp1;
   Float32 psk, ps0, ps1, sq, sq1;
   Word32 i, i0, i1, ix = 0;
   Word16 ipos[2];
   Word16 track1, track2;


   psk = -1;
   alpk = 1;

   for ( i = 0; i < 2; i++ ) {
      codvec[i] = i;
   }

   /*
    * main loop: try 2x4  tracks.
    */
   for ( track1 = 0; track1 < 2; track1++ ) {
      for ( track2 = 0; track2 < 4; track2++ ) {
         /* fix starting position */
         ipos[0] = startPos1[track1];
         ipos[1] = startPos2[track2];

         /*
          * i0 loop: try 8 positions.
          */
         for ( i0 = ipos[0]; i0 < L_CODE; i0 += STEP ) {
            ps0 = dn[i0];
            alp0 = rr[i0][i0] * 0.25F;

            /*
             * i1 loop: 8 positions.
             */
            sq = -1;
            alp = 1;
            ix = ipos[1];

            for ( i1 = ipos[1]; i1 < L_CODE; i1 += STEP ) {
               ps1 = ps0 + dn[i1];

               /* alp1 = alp0 + rr[i0][i1] + 1/2*rr[i1][i1]; */
               alp1 = alp0 + rr[i1][i1] * 0.25F;
               alp1 += rr[i0][i1] * 0.5F;
               sq1 = ps1 * ps1;

               if ( ( alp * sq1 ) > ( sq * alp1 ) ) {
                  sq = sq1;
                  alp = alp1;
                  ix = i1;
               }
            }

            /*
             * memorise codevector if this one is better than the last one.
             */
            if ( ( alpk * sq ) > ( psk * alp ) ) {
               psk = sq;
               alpk = alp;
               codvec[0] = i0;
               codvec[1] = ix;
            }
         }
      }
   }
   return;
}
```

file sp_enc.c before the change (lines 5670 to 5681)

```
static void search_8i40( Float32 dn[], Float32 rr[][L_CODE], Word32 ipos[],
      Word32 pos_max[], Word32 codvec[] )
{
   Float32 rrv[L_CODE];
   Float32 psk, ps, ps0, ps1, ps2, sq, sq2, alpk, alp, alp0, alp1, alp2;
   Float32 *p_r, *p_r0, *p_r1, *p_r2, *p_r3, *p_r4, *p_r5, *p_r6, *p_r7, *p_r8;
   Float32 *p_rrv, *p_rrv0, *p_rrv_max, *p_dn, *p_dn0, *p_dn1, *p_dn_max;
   Word32 i0, i1, i2, i3, i4, i5, i6, i7, j, k, ia, ib, i, pos;


   p_rrv_max = &rrv[L_CODE];
   p_dn_max = &dn[39];
```

file sp_enc.c after the change

```
static void search_8i40( Float32 dn[], Float32 rr[][L_CODE], Word32 ipos[],
      Word32 pos_max[], Word32 codvec[] )
{
   Float32 rrv[L_CODE];
   Float32 psk, ps, ps0, ps1, ps2, sq, sq2, alpk, alp, alp0, alp1, alp2;
   Float32 *p_r, *p_r0, *p_r1, *p_r2, *p_r3, *p_r4, *p_r5, *p_r6, *p_r7, *p_r8;
   Float32 *p_rrv, *p_rrv0, *p_dn, *p_dn0, *p_dn1, *p_dn_max;
   Word32 i0, i1, i2, i3, i4, i5, i6, i7, j, k, ia, ib, i, pos;

   p_dn_max = &dn[39];
```

file sp_enc.c before the change (lines 6312 to 6324)

```
static void search_10i40( Float32 dn[], Float32 rr[][L_CODE], Word32 ipos[],
      Word32 pos_max[], Word32 codvec[] )
{
   Float32 rrv[L_CODE];
   Float32 psk, ps, ps0, ps1, ps2, sq, sq2, alpk, alp, alp0, alp1, alp2;
   Float32 *p_r, *p_r0, *p_r1, *p_r2, *p_r3, *p_r4, *p_r5, *p_r6, *p_r7, *p_r8,
         *p_r9, *p_r10;
   Float32 *p_rrv, *p_rrv0, *p_rrv_max, *p_dn, *p_dn0, *p_dn1, *p_dn_max;
   Word32 i0, i1, i2, i3, i4, i5, i6, i7, i8, i9, j, k, ia, ib, i, pos;


   p_rrv_max = &rrv[L_CODE];
   p_dn_max = &dn[39];
```

file sp_enc.c after the change

```
static void search_10i40( Float32 dn[], Float32 rr[][L_CODE], Word32 ipos[],
      Word32 pos_max[], Word32 codvec[] )
{
   Float32 rrv[L_CODE];
   Float32 psk, ps, ps0, ps1, ps2, sq, sq2, alpk, alp, alp0, alp1, alp2;
   Float32 *p_r, *p_r0, *p_r1, *p_r2, *p_r3, *p_r4, *p_r5, *p_r6, *p_r7, *p_r8,
         *p_r9, *p_r10;
   Float32 *p_rrv, *p_rrv0, *p_dn, *p_dn0, *p_dn1, *p_dn_max;
   Word32 i0, i1, i2, i3, i4, i5, i6, i7, i8, i9, j, k, ia, ib, i, pos;

   p_dn_max = &dn[39];
```

3GPP TSG-SA WG4 Meeting #18
Erlangen, Germany, 3 Sep – 7 Sep 2001

Tdoc S4 (01)0410R

CR-Form-v4

CHANGE REQUEST

⌘	**26.104**	**CR**	**010**	⌘	ev	**1**	⌘	Current version:	**4.1.1**	⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘	(U)SIM	☐	ME/UE	X	Radio Access Network	☐	Core Network	X	

Title:	⌘	Correction to make encoder and decoder memories independent	
Source:	⌘	TSG SA WG4	
Work item code:	⌘	Low bit rate codec for Multimedia Telephony	***Date:*** ⌘ 24.9.2001
Category:	⌘	**A**	***Release:*** ⌘ Rel-4

Use one of the following categories:
- ***F*** *(correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(addition of feature),*
- ***C*** *(functional modification of feature)*
- ***D*** *(editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘ Encoder module is dependent on decoder constant tables, some of the variables are unused

Summary of change: ⌘

1. Parameter sizes are moved from file rom_dec.h to interf_rom.h
   interf_rom.h is #included into rom_dec.h
2. gamma3_MR122 and gamma4 tables are combined into a single table in rom_dec.h
3. Remaining floating-point operations are converted to fixed-point operations in the decoder (sp_dec.c)
   struct: Cb_gain_averageState
   function: A_Refl, Log2, gc_pred, Cb_gain_average, sqrt_l_exp, Ex_ctrl, Inv_sqrt, agc2, agc, dtx_dec_activity_update

4. unused variables removed (sp_enc.c):
   Az_lsp
   search_2i40_9bits
   search_2i40_11bits
   search_8i40
   search_10i40

Consequences if not approved: ⌘ Encoder cannot be compiled without decoder files.
Same information is copied to multiple memory locations
Decoder can not be imported to fixed-point platforms without modifications

Clauses affected: ⌘ interf_rom.h, rom_dec.h, sp_dec.c, sp_enc.c

Other specs affected:	⌘	☐	Other core specifications	⌘
		☐	Test specifications	
		☐	O&M Specifications	

Other comments: ⌘	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

How the code is changed

1. Parameter sizes are moved from file rom_dec.h to interf_rom.h interf_rom.h is #included into rom_dec.h.

file interf_rom.h before the change (lines 28 to 37) (Blue lines are to be deleted)

```
/*
 * include files
 */
#include"typedef.h"
#include"rom_dec.h"

/*
 * definition of constants
 */
```

file interf_rom.h after the change (lines 28 to 45) (Red lines are to be inserted)

```
/*
 * include files
 */
#include"typedef.h"

/*
 * definition of constants
 */

/* number of parameters */
#define PRMNO_MR475 17
#define PRMNO_MR515 19
#define PRMNO_MR59  19
#define PRMNO_MR67  19
#define PRMNO_MR74  19
#define PRMNO_MR795 23
#define PRMNO_MR102 39
#define PRMNO_MR122 57
#define PRMNO_MRDTX 5
```

file interf_rom.h after the change (end of file) (Red lines are to be inserted)

```
/* parameter sizes (# of bits), one table per mode */
static const Word16 bitno_MR475[PRMNO_MR475] =
{
   8, 8, 7,    /* LSP VQ          */
   8, 7, 2, 8, /* first subframe  */
   4, 7, 2,    /* second subframe */
   4, 7, 2, 8, /* third subframe  */
   4, 7, 2     /* fourth subframe */
};
static const Word16 bitno_MR515[PRMNO_MR515] =
{
   8, 8, 7,    /* LSP VQ          */
   8, 7, 2, 6, /* first subframe  */
   4, 7, 2, 6, /* second subframe */
   4, 7, 2, 6, /* third subframe  */
   4, 7, 2, 6  /* fourth subframe */
};
static const Word16 bitno_MR59[PRMNO_MR59] =
{
   8, 9, 9,    /* LSP VQ          */
   8, 9, 2, 6, /* first subframe  */
   4, 9, 2, 6, /* second subframe */
   8, 9, 2, 6, /* third subframe  */
   4, 9, 2, 6  /* fourth subframe */
};
static const Word16 bitno_MR67[PRMNO_MR67] =
{
   8, 9, 9,       /* LSP VQ          */
   8, 11, 3, 7,   /* first subframe  */
   4, 11, 3, 7,   /* second subframe */
   8, 11, 3, 7,   /* third subframe  */
   4, 11, 3, 7    /* fourth subframe */
};
static const Word16 bitno_MR74[PRMNO_MR74] =
{
   8, 9, 9,       /* LSP VQ          */
   8, 13, 4, 7,   /* first subframe  */
   5, 13, 4, 7,   /* second subframe */
   8, 13, 4, 7,   /* third subframe  */
   5, 13, 4, 7    /* fourth subframe */
};
static const Word16 bitno_MR795[PRMNO_MR795] =
{
   9, 9, 9,          /* LSP VQ          */
   8, 13, 4, 4, 5,   /* first subframe  */
   6, 13, 4, 4, 5,   /* second subframe */
   8, 13, 4, 4, 5,   /* third subframe  */
   6, 13, 4, 4, 5    /* fourth subframe */
};
static const Word16 bitno_MR102[PRMNO_MR102] =
{
   8, 9, 9,                         /* LSP VQ          */
   8, 1, 1, 1, 1, 10, 10, 7, 7,  /* first subframe  */
   5, 1, 1, 1, 1, 10, 10, 7, 7,  /* second subframe */
   8, 1, 1, 1, 1, 10, 10, 7, 7,  /* third subframe  */
   5, 1, 1, 1, 1, 10, 10, 7, 7   /* fourth subframe */
};
static const Word16 bitno_MR122[PRMNO_MR122] =
{
   7, 8, 9, 8, 6,                              /* LSP VQ          */
   9, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 5,    /* first subframe  */
   6, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 5,    /* second subframe */
   9, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 5,    /* third subframe  */
   6, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 5     /* fourth subframe */
};
static const Word16 bitno_MRDTX[PRMNO_MRDTX] =
{
   3, 8, 9, 9, 6
};
```

file rom_dec.h before the change (lines 26 to 30)

```
/*
 * include files
 */
#include"typedef.h"
```

file rom_dec.h after the change (Red lines are to be inserted)

```
/*
 * include files
 */
#include"typedef.h"
#include"interf_rom.h"
```

file rom_dec.h before the change (lines 206 to 275) (Blue lines are to be deleted)

```
static const Word32 gamma4[M] =
{
   22938,
   16057,
   11240,
   7868,
   5508,
   3856,
   2699,
   1889,
   1322,
   925
};

/* parameter sizes (# of bits), one table per mode */
static const Word16 bitno_MR475[PRMNO_MR475] =
{
   8,
   8,
   7,
   /* LSP VQ          */ 8,
… … … …
… … … …
… … … …
};
static const Word16 bitno_MRDTX[PRMNO_MRDTX] =
{
   3,
   8,
   9,
   9,
   6
};

/* adaptive codebook gain quantization table (MR122, MR795) */
#define NB_QUA_PITCH 16
static const Word32 qua_gain_pitch[NB_QUA_PITCH] =
```

file rom_dec.h after the change

```
static const Word32 gamma4[M] =
{
   22938,
   16057,
   11240,
   7868,
   5508,
   3856,
   2699,
   1889,
   1322,
   925
};

/* adaptive codebook gain quantization table (MR122, MR795) */
#define NB_QUA_PITCH 16
static const Word32 qua_gain_pitch[NB_QUA_PITCH] =
```

2. gamma3_MR122 and gamma4 tables are combined into a single table in rom_dec.h

file rom_dec.h before the change (lines 167 to 179)

```
 static const Word32 gamma3_MR122[M] =
{
   22938,
   16057,
   11240,
   7868,
   5508,
   3856,
   2699,
   1889,
   1322,
   925
};
```

file rom_dec.h before the change (lines 206 to 218) (Blue lines are to be deleted)

```
static const Word32 gamma4[M] =
{
   22938,
   16057,
   11240,
   7868,
   5508,
   3856,
   2699,
   1889,
   1322,
   925
};
```

file rom_dec.h after the change

```
static const Word32 gamma4_gamma3_MR122[M] =
{
  22938,
  16057,
  11240,
  7868,
  5508,
  3856,
  2699,
  1889,
  1322,
  925
};
```

file sp_dec.c before the change (lines 5390 to 5409)

```
static void Post_Filter( Post_FilterState *st, enum Mode mode, Word32 *syn,
      Word32 *Az_4 )
{
   Word32 h[22], Ap3[MP1], Ap4[MP1];   /* bandwidth expanded LP parameters */
   Word32 tmp, i_subfr, i, temp1, temp2, overflow = 0;
   Word32 *Az, *p1, *p2, *syn_work = &st->synth_buf[M];
   const Word32 *pgamma3 = &gamma3[0];
   const Word32 *pgamma4 = &gamma4[0];


   /*
    * Post filtering
    */
   memcpy( syn_work, syn, L_FRAME <<2 );
   Az = Az_4;

   if ( ( mode == MR122 ) || ( mode == MR102 ) ) {
      pgamma3 = &gamma3_MR122[0];
      pgamma4 = &gamma4_MR122[0];
   }
```

file sp_dec.c after the change

```
static void Post_Filter( Post_FilterState *st, enum Mode mode, Word32 *syn,
      Word32 *Az_4 )
{
   Word32 h[22], Ap3[MP1], Ap4[MP1];   /* bandwidth expanded LP parameters */
   Word32 tmp, i_subfr, i, temp1, temp2, overflow = 0;
   Word32 *Az, *p1, *p2, *syn_work = &st->synth_buf[M];
   const Word32 *pgamma3 = &gamma3[0];
   const Word32 *pgamma4 = &gamma4_gamma3_MR122[0];


   /*
    * Post filtering
    */
   memcpy( syn_work, syn, L_FRAME <<2 );
   Az = Az_4;

   if ( ( mode == MR122 ) || ( mode == MR102 ) ) {
      pgamma3 = &gamma4_gamma3_MR122[0];
      pgamma4 = &gamma4_MR122[0];
   }
```

3. floating-point operations -> fixed-point operations

file sp_dec.c before the change (lines 55 to 62)

```
typedef struct
{
   Float32 hangCount;   /* counter; */
   /* history vector of past synthesis speech energy */
   Word32 cbGainHistory[L_CBGAINHIST];
   Word16 hangVar;   /* counter; */

}Cb_gain_averageState;
```

file sp_dec.c after the change

```
typedef struct
{
   Word32 hangCount;   /* counter; */
   /* history vector of past synthesis speech energy */
   Word32 cbGainHistory[L_CBGAINHIST];
   Word16 hangVar;   /* counter; */

}Cb_gain_averageState;
```

file sp_dec.c before the change (lines 974 to 986)

```
   /* backward Levinson recursion */
   for ( i = M - 1; i >= 0; i-- ) {
      if ( labs( aState[i] ) >= 4096 ) {
         goto ExitRefl;
      }
      refl[i] = aState[i] << 3;
      temp = ( refl[i] * refl[i] ) << 1;
      acc = ( MAX_32 - temp );
      frexp( ( Float64 )acc, &normShift );
      normShift = 31 - normShift;
      scale = 15 - normShift;
      acc = ( acc << normShift );
      temp = ( acc + ( Word32 )0x00008000L );
```

file sp_dec.c after the change

```
   /* backward Levinson recursion */
   for ( i = M - 1; i >= 0; i-- ) {
      if ( labs( aState[i] ) >= 4096 ) {
         goto ExitRefl;
      }
      refl[i] = aState[i] << 3;
      temp = ( refl[i] * refl[i] ) << 1;
      acc = ( MAX_32 - temp );
      normShift=0;
      if (acc != 0){
         temp = acc;
         while (!(temp & 0x40000000))
         {
            normShift++;
            temp = temp << 1;
         }
      }
      else{
         normShift = 0;
      }
      scale = 15 - normShift;
      acc = ( acc << normShift );
      temp = ( acc + ( Word32 )0x00008000L );
```

file sp_dec.c before the change (lines 1097 to 1105)

```
static void Log2( Word32 x, Word32 *exponent, Word32 *fraction )
{
   int exp;


   frexp( ( Float64 )x, &exp );
   exp = 31 - exp;
   Log2_norm( x <<exp, exp, exponent, fraction );
}
```

file sp_dec.c after the change

```
static void Log2( Word32 x, Word32 *exponent, Word32 *fraction )
{
   int tmp, exp=0;

   if (x != 0){
         tmp = x;
         while (!((tmp & 0x80000000) ^ ((tmp & 0x40000000) << 1)))
         {
            exp++;
            tmp = tmp << 1;
         }
      }
   Log2_norm( x <<exp, exp, exponent, fraction );
}
```

file sp_dec.c before the change (lines 3097 to 3103)

```
        /*
         * Compute: meansEner - 10log10(ener_code/ LSufr)
         */
      frexp( ( Float64 )ener_code, &exp_code );
      exp_code = 31 - exp_code;
      ener_code <<= exp_code;
```

file sp_dec.c after the change

```
        /*
         * Compute: meansEner - 10log10(ener_code/ LSufr)
         */
      exp_code=0;
      if (ener_code != 0){
         while (!(ener_code & 0x40000000))
         {
            exp_code++;
            ener_code = ener_code << 1;
         }
      }
```

file sp_dec.c before the change (lines 3674 to 3685)

```
   /* compute lsp difference */
   for ( i = 0; i < M; i++ ) {
      tmp1 = labs( lspAver[i]- lsp[i] );
      frexp( ( Float64 )tmp1, &shift1 );
      frexp( ( Float64 )lspAver[i], &shift2 );
      tmp2 = lspAver[i];
      shift2 = 15 - shift2;
      tmp2 <<= shift2;
      shift1 = 14 - shift1;
      tmp1 = tmp1 << shift1;
      tmp[i] = ( tmp1 << 15 ) / tmp2;
      shift = 2 + shift1 - shift2;
```

file sp_dec.c after the change

```
   /* compute lsp difference */
   for ( i = 0; i < M; i++ ) {
      tmp1 = labs( lspAver[i]- lsp[i] );
      shift1 = 0;
      if (tmp1 != 0){
         while (!(tmp1 & 0x2000))
         {
            shift1++;
            tmp1 = tmp1 << 1;
         }
      }
      tmp2 = lspAver[i];
      shift2 = 0;
      if (tmp2 != 0){
         while (!(tmp2 & 0x4000))
         {
            shift2++;
            tmp2 = tmp2 << 1;
         }
      }
      tmp[i] = ( tmp1 << 15 ) / tmp2;
      shift = 2 + shift1 - shift2;
```

file sp_dec.c before the change (lines 3789 to 3790)

```
   st->hangCount += 1;
   return cbGainMix;
```

file sp_dec.c after the change

```
   st->hangCount += 1;
   if (st->hangCount & 0x80000000)
      st->hangCount = 40;
   return cbGainMix;
```

file sp_dec.c before the change (lines 4008 to 4031)

```
static Word32 sqrt_l_exp( Word32 x, Word32 *exp )
{
   Word32 y, a, i, tmp;
   int e;


   if ( x <= ( Word32 )0 ) {
      *exp = 0;
      return( Word32 )0;
   }
   frexp( ( Float64 )x, &e );
   e = ( 31 - e ) & 0xFFFE;
   x = ( x << e );
   *exp = ( Word16 )e;
   x = ( x >> 9 );
   i = ( Word16 )( x >> 16 );
   x = ( x >> 1 );
   a = x & ( Word16 )0x7fff;
   i = ( i - 16 );
   y = ( sqrt_table[i] << 16 );
   tmp = ( sqrt_table[i] - sqrt_table[i + 1] );
   y -= ( tmp * a ) << 1;
   return( y );
}
```

file sp_dec.c after the change

```
static Word32 sqrt_l_exp( Word32 x, Word32 *exp )
{
   Word32 y, a, i, tmp;
   int e;


   if ( x <= ( Word32 )0 ) {
      *exp = 0;
      return( Word32 )0;
   }
   e=0;
   if (x != 0){
      tmp = x;
      while (!(tmp & 0x40000000))
      {
         e++;
         tmp = tmp << 1;
      }
   }
   e = e & 0xFFFE;
   x = ( x << e );
   *exp = ( Word16 )e;
   x = ( x >> 9 );
   i = ( Word16 )( x >> 16 );
   x = ( x >> 1 );
   a = x & ( Word16 )0x7fff;
   i = ( i - 16 );
   y = ( sqrt_table[i] << 16 );
   tmp = ( sqrt_table[i] - sqrt_table[i + 1] );
   y -= ( tmp * a ) << 1;
   return( y );
}
```

file sp_dec.c before the change (lines 4082 to 4088)

```
      /* scaleFactor=avgEnergy/excEnergy in Q0 */
      frexp( ( Float64 )excEnergy, &exp );
      exp = 15 - exp;
      excEnergy = excEnergy << exp;
      excEnergy = 536838144 / excEnergy;
      T0 = ( avgEnergy * excEnergy ) << 1;
      T0 = ( T0 >> ( 20 - exp ) );
```

file sp_dec.c after the change

```
      /* scaleFactor=avgEnergy/excEnergy in Q0 */
      exp=0;
      if (excEnergy != 0){
         while (!(excEnergy & 0x4000))
         {
            exp++;
            excEnergy = excEnergy << 1;
         }
      }
      excEnergy = 536838144 / excEnergy;
      T0 = ( avgEnergy * excEnergy ) << 1;
      T0 = ( T0 >> ( 20 - exp ) );
```

file sp_dec.c before the change (lines 4125 to 4138)

```
static Word32 Inv_sqrt( Word32 x )
{
   int i, a, tmp, exp;
   Word32 y;


   if ( x <= ( Word32 )0 )
      return( ( Word32 )0x3fffffffL );
   frexp( ( Float64 )x, &exp );
   exp = 31 - exp;

   /* x is normalized */
   x = ( x << exp );
   exp = ( 30 - exp );
```

file sp_dec.c after the change

```
static Word32 Inv_sqrt( Word32 x )
{
   int i, a, tmp, exp;
   Word32 y;


   if ( x <= ( Word32 )0 )
      return( ( Word32 )0x3fffffffL );
   exp=0;
   while (!(x & 0x40000000))
   {
      exp++;
      x = x << 1;
   }

   /* x is normalized */
   exp = ( 30 - exp );
```

file sp_dec.c before the change (lines 4270 to 4296)

```
static void agc2( Word32 *sig_in, Word32 *sig_out )
{
   Word32 s;
   int i, exp;
   Word16 gain_in, gain_out, g0;


   /* calculate gain_out with exponent */
   s = energy_new( sig_out );

   if ( s == 0 ) {
      return;
   }
   frexp( ( Float64 )s, &exp );
   exp = 30 - exp;
   gain_out = ( Word16 )( ( ( s << exp ) + 0x00008000L ) >> 16 );

   /* calculate gain_in with exponent */
   s = energy_new( sig_in );

   if ( s == 0 ) {
      g0 = 0;
   }
   else {
      frexp( ( Float64 )s, &i );
      i = 31 - i;
      s <<= i;

      if ( s < 0x7fff7fff )
         gain_in = ( Word16 )( ( s + 0x00008000L ) >> 16 );
      else
         gain_in = 32767;
      exp = ( exp - i );
```

file sp_dec.c after the change

```
static void agc2( Word32 *sig_in, Word32 *sig_out )
{
   Word32 s;
   int i, exp;
   Word16 gain_in, gain_out, g0;


   /* calculate gain_out with exponent */
   s = energy_new( sig_out );

   if ( s == 0 ) {
      return;
   }
   exp=0;
   while (!(s & 0x20000000))
   {
      exp++;
      s = s << 1;
   }

   gain_out = ( Word16 )( ( s + 0x00008000L ) >> 16 );

   /* calculate gain_in with exponent */
   s = energy_new( sig_in );

   if ( s == 0 ) {
      g0 = 0;
   }
   else {
      i = 0;
      while (!(s & 0x40000000))
      {
         i++;
         s = s << 1;
      }

      if ( s < 0x7fff7fff )
         gain_in = ( Word16 )( ( s + 0x00008000L ) >> 16 );
      else
         gain_in = 32767;
      exp = ( exp - i );
```

file sp_dec.c before the change (lines 4487 to 4511)

```
static void dtx_dec_activity_update( dtx_decState *st, Word32 lsf[], Word32
      frame[] )
{
   Float64 frame_en;
   Word32 log_en_e, log_en_m, log_en, i;


   /* update lsp history */
   st->lsf_hist_ptr += M;

   if ( st->lsf_hist_ptr == 80 ) {
      st->lsf_hist_ptr = 0;
   }
   memcpy( &st->lsf_hist[st->lsf_hist_ptr], lsf, M <<2 );

   /* compute log energy based on frame energy */
   frame_en = 0;   /* Q0 */

   for ( i = 0; i < L_FRAME; i ++ ) {
      frame_en += frame[i] * frame[i];
   }

   log_en = ( frame_en > 0x3fffffff ) ? 0x7FFFFFFE: (Word32)frame_en << 1;

   Log2( log_en , &log_en_e, &log_en_m );
```

file sp_dec.c after the change

```
static void dtx_dec_activity_update( dtx_decState *st, Word32 lsf[], Word32
      frame[] )
{
   Word32 frame_en;
   Word32 log_en_e, log_en_m, log_en, i;


   /* update lsp history */
   st->lsf_hist_ptr += M;

   if ( st->lsf_hist_ptr == 80 ) {
      st->lsf_hist_ptr = 0;
   }
   memcpy( &st->lsf_hist[st->lsf_hist_ptr], lsf, M <<2 );

   /* compute log energy based on frame energy */
   frame_en = 0;   /* Q0 */

   for ( i = 0; (i < L_FRAME); i ++ ) {
      frame_en += frame[i] * frame[i];
      if (frame_en & 0x80000000)
         break;
   }

   log_en = (frame_en & 0xC0000000) ? 0x7FFFFFFE: (Word32)frame_en << 1;

   Log2( log_en , &log_en_e, &log_en_m );
```

file sp_dec.c before the change (lines 5282 to 5312)

```
static void agc( agcState *st, Word32 *sig_in, Word32 *sig_out, Word16 agc_fac )
{
   Word32 s, gain_in, gain_out, g0, gain;
   int exp, i;


   /* calculate gain_out with exponent */
   s = energy_new( sig_out );

   if ( s == 0 ) {
      st->past_gain = 0;
      return;
   }
   frexp( ( Float32 )s, &exp );
   exp = 30 - exp;

   if ( exp >= 0 )
      gain_out = ( ( s << exp ) + 0x00008000L ) >> 16;
   else
      gain_out = ( ( s >> abs( exp ) ) + 0x00008000L ) >> 16;

   /* calculate gain_in with exponent */
   s = energy_new( sig_in );

   if ( s == 0 ) {
      g0 = 0;
   }
   else {
      frexp( ( Float32 )s, &i );
      i = 31 - i;
      s = ( s << i ) + 0x00008000L;
```

file sp_dec.c after the change

```
static void agc( agcState *st, Word32 *sig_in, Word32 *sig_out, Word16 agc_fac )
{
   Word32 s, gain_in, gain_out, g0, gain;
   int exp, i;


   /* calculate gain_out with exponent */
   s = energy_new( sig_out );

   if ( s == 0 ) {
      st->past_gain = 0;
      return;
   }
   exp=0;
   i = s;
   while (!(i & 0x40000000))
   {
      exp++;
      i = i << 1;
   }
   exp -=1;
   if (exp & 0x80000000) {
      s >>= 1;
```

```
}
else {
   s <<= exp;
}
gain_out = ( s + 0x00008000L ) >> 16;

/* calculate gain_in with exponent */
s = energy_new( sig_in );

if ( s == 0 ) {
   g0 = 0;
}
else {
   i=0;
while (!(s & 0x40000000))
{
   i++;
   s = s << 1;
}
   s = s + 0x00008000L;
```

4. unused variables removed (sp_enc.c):

file sp_enc.c before the change (lines 617 to 623)

```
static void Az_lsp( Float32 a[], Float32 lsp[], Float32 old_lsp[] )
{
   Word32 i, j, nf, ip;
   Float32 xlow, ylow, xhigh, yhigh, xmid, ymid, xint;
   Float32 x, y;
   Float32 *coef;
   Float32 f1[6], f2[6];
```

file sp_enc.c after the change

```
static void Az_lsp( Float32 a[], Float32 lsp[], Float32 old_lsp[] )
{
   Word32 i, j, nf, ip;
   Float32 xlow, ylow, xhigh, yhigh, xmid, ymid, xint;
   Float32 y;
   Float32 *coef;
   Float32 f1[6], f2[6];
```

file sp_enc.c before the change (lines 672 to 678)

```
         /*
          * Linear interpolation
          * xint = xlow - ylow*(xhigh-xlow)/(yhigh-ylow)
          */
         x = xhigh - xlow;
         y = yhigh - ylow;
```

file sp_enc.c after the change

```
         /*
          * Linear interpolation
          * xint = xlow - ylow*(xhigh-xlow)/(yhigh-ylow)
          */
         y = yhigh - ylow;
```

file sp_enc.c before the change (lines 4503 to 4558)

```
static void search_2i40_9bits( Word16 subNr, Float32 dn[], Float32 rr[][L_CODE],
      Word32 codvec[] )
{
   Float32 ps, ps0, ps1, psk, alp, alp0, alp1, alpk, sq, sq1;
   Word32 i0, i1, ix, i;
   Word16 ipos[2];
   Word16 track1;


   psk = -1;
   alpk = 1;

   for ( i = 0; i < 2; i++ ) {
      codvec[i] = i;
   }

   /* main loop: try 2x4  tracks */
   for ( track1 = 0; track1 < 2; track1++ ) {
      ipos[0] = startPos[( subNr <<1 )+( track1 << 3 )];
      ipos[1] = startPos[( subNr <<1 )+1 + ( track1 << 3 )];

      /* i0 loop: try 8 positions   */
      for ( i0 = ipos[0]; i0 < L_CODE; i0 += STEP ) {
         ps0 = dn[i0];
         alp0 = rr[i0][i0];

         /* i1 loop: 8 positions */
         sq = -1;
         alp = 1;
         ps = 0;
         ix = ipos[1];

         for ( i1 = ipos[1]; i1 < L_CODE; i1 += STEP ) {
            ps1 = ps0 + dn[i1];
            alp1 = alp0 + rr[i1][i1] + 2.0F * rr[i0][i1];
            sq1 = ps1 * ps1;

            if ( ( alp * sq1 ) > ( sq * alp1 ) ) {
               sq = sq1;
               ps = ps1;
               alp = alp1;
               ix = i1;
            }
         }

         /* memorise codevector if this one is better than the last one   */
         if ( ( alpk * sq ) > ( psk * alp ) ) {
            psk = sq;
            alpk = alp;
            codvec[0] = i0;
            codvec[1] = ix;
         }
      }
   }
   return;
}
```

file sp_enc.c after the change

```
static void search_2i40_9bits( Word16 subNr, Float32 dn[], Float32 rr[][L_CODE],
      Word32 codvec[] )
{
   Float32 ps0, ps1, psk, alp, alp0, alp1, alpk, sq, sq1;
   Word32 i0, i1, ix, i;
   Word16 ipos[2];
   Word16 track1;


   psk = -1;
   alpk = 1;

   for ( i = 0; i < 2; i++ ) {
      codvec[i] = i;
   }

   /* main loop: try 2x4  tracks */
   for ( track1 = 0; track1 < 2; track1++ ) {
      ipos[0] = startPos[( subNr << 1 ) + ( track1 << 3 )];
      ipos[1] = startPos[( subNr << 1 ) + 1 + ( track1 << 3 )];

      /* i0 loop: try 8 positions   */
      for ( i0 = ipos[0]; i0 < L_CODE; i0 += STEP ) {
         ps0 = dn[i0];
         alp0 = rr[i0][i0];

         /* i1 loop: 8 positions */
         sq = -1;
         alp = 1;
         ix = ipos[1];

         for ( i1 = ipos[1]; i1 < L_CODE; i1 += STEP ) {
            ps1 = ps0 + dn[i1];
            alp1 = alp0 + rr[i1][i1] + 2.0F * rr[i0][i1];
            sq1 = ps1 * ps1;

            if ( ( alp * sq1 ) > ( sq * alp1 ) ) {
               sq = sq1;
               alp = alp1;
               ix = i1;
            }
         }

         /* memorise codevector if this one is better than the last one   */
         if ( ( alpk * sq ) > ( psk * alp ) ) {
            psk = sq;
            alpk = alp;
            codvec[0] = i0;
            codvec[1] = ix;
         }
      }
   }
   return;
}
```

file sp_enc.c before the change (lines 4732 to 4802)

```
static void search_2i40_11bits( Float32 dn[], Float32 rr[][L_CODE], Word32
      codvec[] )
{
   Float64 alpk, alp, alp0, alp1;
   Float32 ps, psk, ps0, ps1, sq, sq1;
   Word32 i, i0, i1, ix = 0;
   Word16 ipos[2];
   Word16 track1, track2;
   psk = -1;
   alpk = 1;

   for ( i = 0; i < 2; i++ ) {
      codvec[i] = i;
   }

   /*
    * main loop: try 2x4  tracks.
    */
   for ( track1 = 0; track1 < 2; track1++ ) {
      for ( track2 = 0; track2 < 4; track2++ ) {
         /* fix starting position */
         ipos[0] = startPos1[track1];
         ipos[1] = startPos2[track2];

         /*
          * i0 loop: try 8 positions.
          */
         for ( i0 = ipos[0]; i0 < L_CODE; i0 += STEP ) {
            ps0 = dn[i0];
            alp0 = rr[i0][i0] * 0.25F;

            /*
             * i1 loop: 8 positions.
             */
            sq = -1;
            alp = 1;
            ps = 0;
            ix = ipos[1];

            for ( i1 = ipos[1]; i1 < L_CODE; i1 += STEP ) {
               ps1 = ps0 + dn[i1];

               /* alp1 = alp0 + rr[i0][i1] + 1/2*rr[i1][i1]; */
               alp1 = alp0 + rr[i1][i1] * 0.25F;
               alp1 += rr[i0][i1] * 0.5F;
               sq1 = ps1 * ps1;

               if ( ( alp * sq1 ) > ( sq * alp1 ) ) {
                  sq = sq1;
                  ps = ps1;
                  alp = alp1;
                  ix = i1;
               }
            }

            /*
             * memorise codevector if this one is better than the last one.
             */
            if ( ( alpk * sq ) > ( psk * alp ) ) {
               psk = sq;
               alpk = alp;
               codvec[0] = i0;
               codvec[1] = ix;
            }
         }
      }
   }
   return;
}
```

file sp_enc.c after the change

```
static void search_2i40_11bits( Float32 dn[], Float32 rr[][L_CODE], Word32
      codvec[] )
{
   Float64 alpk, alp, alp0, alp1;
   Float32 psk, ps0, ps1, sq, sq1;
   Word32 i, i0, i1, ix = 0;
   Word16 ipos[2];
   Word16 track1, track2;


   psk = -1;
   alpk = 1;

   for ( i = 0; i < 2; i++ ) {
      codvec[i] = i;
   }

   /*
    * main loop: try 2x4  tracks.
    */
   for ( track1 = 0; track1 < 2; track1++ ) {
      for ( track2 = 0; track2 < 4; track2++ ) {
         /* fix starting position */
         ipos[0] = startPos1[track1];
         ipos[1] = startPos2[track2];

         /*
          * i0 loop: try 8 positions.
          */
         for ( i0 = ipos[0]; i0 < L_CODE; i0 += STEP ) {
            ps0 = dn[i0];
            alp0 = rr[i0][i0] * 0.25F;

            /*
             * i1 loop: 8 positions.
             */
            sq = -1;
            alp = 1;
            ix = ipos[1];

            for ( i1 = ipos[1]; i1 < L_CODE; i1 += STEP ) {
               ps1 = ps0 + dn[i1];

               /* alp1 = alp0 + rr[i0][i1] + 1/2*rr[i1][i1]; */
               alp1 = alp0 + rr[i1][i1] * 0.25F;
               alp1 += rr[i0][i1] * 0.5F;
               sq1 = ps1 * ps1;

               if ( ( alp * sq1 ) > ( sq * alp1 ) ) {
                  sq = sq1;
                  alp = alp1;
                  ix = i1;
               }
            }

            /*
             * memorise codevector if this one is better than the last one.
             */
            if ( ( alpk * sq ) > ( psk * alp ) ) {
               psk = sq;
               alpk = alp;
               codvec[0] = i0;
               codvec[1] = ix;
            }
         }
      }
   }
   return;
}
```

file sp_enc.c before the change (lines 5670 to 5681)

```
static void search_8i40( Float32 dn[], Float32 rr[][L_CODE], Word32 ipos[],
      Word32 pos_max[], Word32 codvec[] )
{
   Float32 rrv[L_CODE];
   Float32 psk, ps, ps0, ps1, ps2, sq, sq2, alpk, alp, alp0, alp1, alp2;
   Float32 *p_r, *p_r0, *p_r1, *p_r2, *p_r3, *p_r4, *p_r5, *p_r6, *p_r7, *p_r8;
   Float32 *p_rrv, *p_rrv0, *p_rrv_max, *p_dn, *p_dn0, *p_dn1, *p_dn_max;
   Word32 i0, i1, i2, i3, i4, i5, i6, i7, j, k, ia, ib, i, pos;


   p_rrv_max = &rrv[L_CODE];
   p_dn_max = &dn[39];
```

file sp_enc.c after the change

```
static void search_8i40( Float32 dn[], Float32 rr[][L_CODE], Word32 ipos[],
      Word32 pos_max[], Word32 codvec[] )
{
   Float32 rrv[L_CODE];
   Float32 psk, ps, ps0, ps1, ps2, sq, sq2, alpk, alp, alp0, alp1, alp2;
   Float32 *p_r, *p_r0, *p_r1, *p_r2, *p_r3, *p_r4, *p_r5, *p_r6, *p_r7, *p_r8;
   Float32 *p_rrv, *p_rrv0, *p_dn, *p_dn0, *p_dn1, *p_dn_max;
   Word32 i0, i1, i2, i3, i4, i5, i6, i7, j, k, ia, ib, i, pos;

   p_dn_max = &dn[39];
```

file sp_enc.c before the change (lines 6312 to 6324)

```
static void search_10i40( Float32 dn[], Float32 rr[][L_CODE], Word32 ipos[],
      Word32 pos_max[], Word32 codvec[] )
{
   Float32 rrv[L_CODE];
   Float32 psk, ps, ps0, ps1, ps2, sq, sq2, alpk, alp, alp0, alp1, alp2;
   Float32 *p_r, *p_r0, *p_r1, *p_r2, *p_r3, *p_r4, *p_r5, *p_r6, *p_r7, *p_r8,
         *p_r9, *p_r10;
   Float32 *p_rrv, *p_rrv0, *p_rrv_max, *p_dn, *p_dn0, *p_dn1, *p_dn_max;
   Word32 i0, i1, i2, i3, i4, i5, i6, i7, i8, i9, j, k, ia, ib, i, pos;


   p_rrv_max = &rrv[L_CODE];
   p_dn_max = &dn[39];
```

file sp_enc.c after the change

```
static void search_10i40( Float32 dn[], Float32 rr[][L_CODE], Word32 ipos[],
      Word32 pos_max[], Word32 codvec[] )
{
   Float32 rrv[L_CODE];
   Float32 psk, ps, ps0, ps1, ps2, sq, sq2, alpk, alp, alp0, alp1, alp2;
   Float32 *p_r, *p_r0, *p_r1, *p_r2, *p_r3, *p_r4, *p_r5, *p_r6, *p_r7, *p_r8,
         *p_r9, *p_r10;
   Float32 *p_rrv, *p_rrv0, *p_dn, *p_dn0, *p_dn1, *p_dn_max;
   Word32 i0, i1, i2, i3, i4, i5, i6, i7, i8, i9, j, k, ia, ib, i, pos;

   p_dn_max = &dn[39];
```

3GPP TSG-SA4 Meeting #18 **Tdoc S4 (01)0502**
Erlangen, Germany, September 3-7, 2001

CR-Form-v4

CHANGE REQUEST

⌘	**26.104**	**CR**	**017**	⌘	ev	**-**	⌘	Current version:	**3.2.0**	⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE [✓ X] Radio Access Network ☐ Core Network [✓ X]

Title:	⌘	Correction of decoder operation in error concealment of lost frames		
Source:	⌘	TSG SA WG4		
Work item code:	⌘	Low bit rate codec for Multimedia Telephony	***Date:*** ⌘	24-Sep-2001
Category:	⌘	**F**	***Release:*** ⌘	R99
		Use one of the following categories: ***F*** *(correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*

Reason for change:	⌘	Inconsistency between 3G TS 26.104 and 3G TS 26.073
Summary of change:	⌘	Underflow of average frame energy in DTX history is prevented.
Consequences if not approved:	⌘	Inconsistency between 3G TS 26.104 and 3G TS 26.073. Approximately 6s of NO_DATA frames causes high-level noise.

Clauses affected:	⌘	sp_dec.c		
Other specs affected:	⌘	☐ Other core specifications ☐ Test specifications ☐ O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

How the code is changed

1. sp_dec.c

function dtx_dec before the change (lines 1825 to 1831)

```
    st->since_last_sid = 0;
    memcpy( st->lsp_old, st->lsp, M <<2 );
    st->old_log_en = st->log_en;

    /* subtract 1/8 in Q11 i.e -6/8 dB */
    st->log_en = st->log_en - 256;
  }
```

function dtx_dec after the change

```
    st->since_last_sid = 0;
    memcpy( st->lsp_old, st->lsp, M <<2 );
    st->old_log_en = st->log_en;

    /* subtract 1/8 in Q11 i.e -6/8 dB */
    st->log_en = st->log_en - 256;
    if (st->log_en < -32768) st->log_en = -32768;
  }
```

3GPP TSG-SA4 Meeting #18
Erlangen, Germany, September 3-7, 2001

Tdoc S4 (01)0502

CR-Form-v4

CHANGE REQUEST

⌘	**26.104**	**CR**	**018**	⌘	ev	**-**	⌘	Current version:	**4.1.1**	⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE [✓ X] Radio Access Network ☐ Core Network [✓ X]

Title: ⌘	Correction of decoder operation in error consealement of lost frames			
Source: ⌘	TSG SA WG4			
Work item code: ⌘	Low bit rate codec for Multimedia Telephony		***Date:*** ⌘	24-Sep-2001
Category: ⌘	**A**		***Release:*** ⌘	REL-4
	Use one of the following categories: ***F*** *(correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.			*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*

Reason for change: ⌘	Inconsistency between 3G TS 26.104 and 3G TS 26.073
Summary of change: ⌘	Underflow of average frame energy in DTX history is prevented.
Consequences if not approved: ⌘	Inconsistency between 3G TS 26.104 and 3G TS 26.073. Approximately 6s of NO_DATA frames causes high-level noise.

Clauses affected: ⌘	sp_dec.c	
Other specs affected: ⌘	☐ Other core specifications ☐ Test specifications ☐ O&M Specifications	⌘
Other comments: ⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

How the code is changed

1. sp_dec.c

function dtx_dec before the change (lines 1825 to 1831)

```
    st->since_last_sid = 0;
    memcpy( st->lsp_old, st->lsp, M <<2 );
    st->old_log_en = st->log_en;

    /* subtract 1/8 in Q11 i.e -6/8 dB */
    st->log_en = st->log_en - 256;
  }
```

function dtx_dec after the change

```
    st->since_last_sid = 0;
    memcpy( st->lsp_old, st->lsp, M <<2 );
    st->old_log_en = st->log_en;

    /* subtract 1/8 in Q11 i.e -6/8 dB */
    st->log_en = st->log_en - 256;
    if (st->log_en < -32768) st->log_en = -32768;
  }
```