

Source: TSG-SA WG4

Title:CRs TS 26.410 & TS 26.411 on Corrections to Enhanced aacPlus codec specifications (Release 6)

Document for: Approval

Agenda Item: 7.4.3

The following CRs, agreed at the TSG-SA WG4 meeting #35, are presented to TSG SA #28 for approval.

Spec	CR	Rev	Phase	Subject	Cat	Vers	WG	Meeting	S4 doc
26.410	023		Rel-6	Correction to C-code: 10 kbit/s mono encoding with stereo input files failed	F	6.2.0	S4	TSG-SA WG4#35	S4-050299
26.411	001		Rel-6	Correction to C-code: Corrections to ETSI operator usage	F	6.0.0	S4	TSG-SA WG4#35	S4-050300
26.411	002		Rel-6	Correction to C-code: Avoid reading of non-initialized variable	F	6.0.0	S4	TSG-SA WG4#35	S4-050355
26.411	003		Rel-6	Correction to C-code: fixes for error concealment	F	6.0.0	S4	TSG-SA WG4#35	S4-050386

CR-Form-v7.1

CHANGE REQUEST

№ 26.410 CR 023 № rev - № Current version: 6.2.0 №

For **HELP** on using this form, see bottom of this page or look at the pop-up text over the № symbols.

Proposed change affects: UICC apps№ ☐ ME ☒ Radio Access Network ☐ Core Network ☒

Title:	№ Correction to C-code: 10 kbit/s mono encoding with stereo input files failed		
Source:	№ TSG SA WG4 Codec		
Work item code:	№ PSSrel6-Stage3	Date:	№ 06/06/2005
Category:	№ F		Release: № Rel-6
Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:	
F (correction)		Ph2 (GSM Phase 2)	
A (corresponds to a correction in an earlier release)		R96 (Release 1996)	
B (addition of feature),		R97 (Release 1997)	
C (functional modification of feature)		R98 (Release 1998)	
D (editorial modification)		R99 (Release 1999)	
Detailed explanations of the above categories can be found in 3GPP TR 21.900 .		Rel-4 (Release 4)	
		Rel-5 (Release 5)	
		Rel-6 (Release 6)	
		Rel-7 (Release 7)	

Reason for change:	№ When feeding the encoder with stereo files and operating at mono at bitrates that require usage of the resampler tool, the encoded result contained dropouts due to false input buffer handling.
Summary of change:	№ Buffer handling for stereo to mono downmix in combination with the input resampler is corrected.
Consequences if not approved:	№ Feeding the encoder with stereo and operating at low mono bitrates results in suboptimal audio performance.

Clauses affected:	№ C-Code appendix				
Other specs affected:	Y	N	Other core specifications		
	№	X		Test specifications	
	№	X			O&M Specifications
	№	X			
Other comments:	№ The problem did not manifest itself when the encoder was fed with mono input, which was done during characterization testing.				

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <http://www.3gpp.org/specs/CR.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.

In the source code file main.c in the encoder:

line 528, add:

```
“  
    int stride = inputInfo.nChannels;  
    #if (MAX_CHANNELS==2)  
        if (inputInfo.nChannels==1)  
            stride = 2;  
    #endif  
“
```

line 572, replace:

```
“  
    &inputBuffer[writeOffset + r*nDSOutBlockSize*inputInfo.nChannels],  
“  
with  
“  
    &inputBuffer[writeOffset + r*nDSOutBlockSize*stride],  
“
```

line 575, add:

```
“  
    if (inputInfo.nChannels==1 && stride==2) {  
        /* reordering necessary since the encoder takes interleaved data */  
        for (i=nDSOutBlockSize-1; i>=0; i--) {  
            inputBuffer[writeOffset+r*nDSOutBlockSize*2+2*i] =  
                inputBuffer[writeOffset+r*nDSOutBlockSize*2+i];  
        }  
    }  
“
```

CHANGE REQUEST

26.411 CR 001 rev - Current version: 6.0.0

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

Proposed change affects: UICC apps ☐ ME ☒ Radio Access Network ☐ Core Network ☒

Title:	Correction to C-code: Corrections to ETSI operator usage	
Source:	TSG SA WG4 Codec	
Work item code:	PSSrel6-Stage3	Date: 06/06/2005
Category:	F	Release: Rel-6
Use <u>one</u> of the following categories:		Use <u>one</u> of the following releases:
F (correction)		Ph2 (GSM Phase 2)
A (corresponds to a correction in an earlier release)		R96 (Release 1996)
B (addition of feature),		R97 (Release 1997)
C (functional modification of feature)		R98 (Release 1998)
D (editorial modification)		R99 (Release 1999)
Detailed explanations of the above categories can be found in 3GPP TR 21.900 .		Rel-4 (Release 4)
		Rel-5 (Release 5)
		Rel-6 (Release 6)
		Rel-7 (Release 7)

Reason for change:	During the verification task for checking the consistent application of ETSI operators, some few files have not been checked. For those files the remaining cleanup to make the application of ETSI operators consistent is necessary.
Summary of change:	In three source code and files and one header file some data types and the instrumentation of ETSI operators is corrected to be consistent. The changes are all bit-exact.
Consequences if not approved:	Some inconsistencies in applying the ETSI operators for the affected files will remain

Clauses affected:	C-Code appendix, written specification									
Other specs affected:	<table><tr><td>Y</td><td>N</td></tr><tr><td>☒</td><td>☒</td></tr><tr><td>☒</td><td>☒</td></tr><tr><td>☒</td><td>☒</td></tr></table>	Y	N	☒	☒	☒	☒	☒	☒	Other core specifications
	Y	N								
	☒	☒								
	☒	☒								
☒	☒									
		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/specs/CR.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.

Changes related to the C-Code Appendix:

In the decoder, file **conceal.h** apply the following patch:

```

46c46
< CConcealment_Init (Word8 ch);
---
> CConcealment_Init (Word16 ch);

```

In the decoder, file **conceal.c** apply the following patch:

```

63c63
<                                     Word32          sfbCnt,
---
>                                     Word16          sfbCnt,
77c77
< CConcealment_Init (Word8 maxChan          /*!< max channels */
---
> CConcealment_Init (Word16 maxChan          /*!< max channels */
81c81
<   Word8  ch;
---
>   Word16 ch;
165c165
<   /* if previous frame was not ok */
logic32(); test();
---
>   /* if previous frame was not ok */
test();
171c171
<   frame (f_(n-2) or f_(n-3) or ...). */
logic32(); test();
---
>   frame (f_(n-2) or f_(n-3) or ...). */
test(); test();
175c175
<   if (pIcsInfo->WindowSequence == EightShortSequence) { /* f_(n-2) ==
EightShortSequence */
---
>   if (sub(pIcsInfo->WindowSequence, EightShortSequence) == 0) { /* f_(n-2)
== EightShortSequence */
183c183
<       Word32 scaleFactorBandsTotal = SamplingRateInfoTable[pIcsInfo-
>SamplingRateIndex].NumberOfScaleFactorBands_Short;
---
>       Word16 scaleFactorBandsTotal = SamplingRateInfoTable[pIcsInfo-
>SamplingRateIndex].NumberOfScaleFactorBands_Short;
185c185
<
move32(); move16();
---
>
move16(); move16();
217c217
<       Word32 scaleFactorBandsTotal = SamplingRateInfoTable[pIcsInfo-
>SamplingRateIndex].NumberOfScaleFactorBands_Long;
---

```

```

>          Word16 scaleFactorBandsTotal = SamplingRateInfoTable[pIcsInfo-
>SamplingRateIndex].NumberOfScaleFactorBands_Long;
219c219
<          Word16 specScaleOut;
move32();move16();
---
>          Word16 specScaleOut;
move16();move16();
263c263
<          Word32 scaleFactorBandsTotal = SamplingRateInfoTable[pIcsInfo-
>SamplingRateIndex].NumberOfScaleFactorBands_Long;
---
>          Word16 scaleFactorBandsTotal = SamplingRateInfoTable[pIcsInfo-
>SamplingRateIndex].NumberOfScaleFactorBands_Long;
265c265
<          Word16 specScaleAct          = pConcealmentInfo->SpecScale[ch][0];
move32();move16();move16();
---
>          Word16 specScaleAct          = pConcealmentInfo->SpecScale[ch][0];
move16();move16();move16();
368c368
<  int i;
---
>  Word16 i;
394c394
<  Word32 line;
---
>  Word16 line;
397c397
<  Word32 scaleFactorBandsTotal = 0;
---
>  Word16 scaleFactorBandsTotal = 0;
402c402,405
<  line = 0;
move32();
---
>  move32(); /* init enAccu */
>  move16(); /* init scaleFactorBandsTotal */
>
>  line = 0;
move16();
413c416
<  scaleFactorBandsTotal =
SamplingRateInfoTable[samplingRateIndex].NumberOfScaleFactorBands_Long;
move32();
---
>  scaleFactorBandsTotal =
SamplingRateInfoTable[samplingRateIndex].NumberOfScaleFactorBands_Long;
move16();
416a420
>  Word16 line_stop;
419c423,424
<  for( ; line < pSfbOffset[sfb+1]; line++) {
---
>  line_stop = pSfbOffset[sfb+1]; move16();
>  for( ; line < line_stop; line++) {
428c433
<  scaleFactorBandsTotal =
SamplingRateInfoTable[samplingRateIndex].NumberOfScaleFactorBands_Short;
move32();
---

```



```

>         scaleFactorBandsTotal =
SamplingRateInfoTable[samplingRateIndex].NumberOfScaleFactorBands_Short;
move16();
431a437
>         Word16 line_stop;
434c440,441
<         for(; line < shl(pSfbOffset[sfb+1],3); line++) {
---
>         line_stop = shl(pSfbOffset[sfb+1],3);
>         for(; line < line_stop; line++) {
452,453c459,460
<         /* standart short calculation */
<         scaleFactorBandsTotal =
SamplingRateInfoTable[samplingRateIndex].NumberOfScaleFactorBands_Short;
move32();
---
>         /* standard short calculation */
>         scaleFactorBandsTotal =
SamplingRateInfoTable[samplingRateIndex].NumberOfScaleFactorBands_Short;
move16();
456a464
>         Word16 line_stop;
459c467,468
<         for( ; line < pSfbOffset[sfb+1]; line++) {
---
>         line_stop = pSfbOffset[sfb+1]; move16();
>         for( ; line < line_stop; line++) {
470c479
<         scaleFactorBandsTotal =
SamplingRateInfoTable[samplingRateIndex].NumberOfScaleFactorBands_Long;
move32();
---
>         scaleFactorBandsTotal =
SamplingRateInfoTable[samplingRateIndex].NumberOfScaleFactorBands_Long;
move16();
473a483
>         Word16 line_stop;
476,477c486,488
<         for ( ; line < pSfbOffset[sfb+1]; line++) {
<         Word16 tmp = extract_h(spectrum[L_shr(line, 3)]);
---
>         line_stop = pSfbOffset[sfb+1]; move16();
>         for ( ; line < line_stop; line++) {
>         Word16 tmp = extract_h(spectrum[shr(line, 3)]);
502c513
<                                     Word32          sfbCnt,
---
>                                     Word16          sfbCnt,
544a556
>         /* CConcealment_0k is 0x0000, therefore, no need to instrument */
546,547c558
<
logic16();logic16();
<
logic16();logic16();test();
---
>
test(); test(); test(); test();
565,566c576
<
logic16();logic16();
<
logic16();logic16();test();

```

```

---
>
test(); test(); test(); test();
573a584
>         add(pConcealmentInfo->cntConcealFrame[ch], 1); move16();
598a610
>         add(pConcealmentInfo->cntConcealFrame[ch], 1); move16();
614a627
>         add(pConcealmentInfo->cntConcealFrame[ch], 1); move16();

```

In the decoder, file **imdct.c** apply the following patch:

```

20c20
< static UWord16 permutationTable [480] =
---
> static const UWord16 permutationTable [480] =
76c76
< static UWord16 permutationTable [56] =
---
> static const UWord16 permutationTable [56] =
104c104
< static void preTwiddle (Word32 x [], Word32 m, const Word16 sineWindow [])
---
> static void preTwiddle (Word32 x [], Word16 m, const Word16 sineWindow [])
107d106
< Word16 im2_hi, im2_lo;
109,111c108
< Word16 count = shr((Word16)m, 1);
< Word16 m2 = shl((Word16)m, 1);
< Word16 *px_16Bit = (Word16 *)x;
move16();
---
> Word16 count = shr(m, 1);
115c112
< if(m == 128) {
---
> if(sub(m,128) == 0) {
122,123c119
< im2_lo = logical_shr(px_16Bit[j+2], 1);
< im2_hi = px_16Bit[j+3];
move16();
---
> Word32 xtmp = x[i+1];
move32();
125,128c121,129
< cplxmul_scale (x + i, px_16Bit [m2-j-1], logical_shr(px_16Bit [m2-j-2], 1),
sineWindow [i],
< px_16Bit [j +1], logical_shr(px_16Bit [j], 1), sineWindow
[m-1-i], shift) ;
< cplxmul_scale (x + m-2 - i, im2_hi, im2_lo, sineWindow [m-2 - i], px_16Bit
[m2-j-3],
< logical_shr(px_16Bit [m2-j-4], 1), sineWindow [i+1], shift)
;
---
> x[i+1] = L_sub(L_shr(fixmul_32x16b(x[m-i-1], sineWindow[m-1-i]), shift),
> L_shr(fixmul_32x16b(x[i], sineWindow[i]), shift));
> x[i] = L_add(L_shr(fixmul_32x16b(x[m-i-1], sineWindow[i]), shift),
> L_shr(fixmul_32x16b(x[i], sineWindow[m-1-i]), shift));
>
> x[m-1-i] = L_sub(L_shr(fixmul_32x16b(xtmp, sineWindow[i+1]), shift),
> L_shr(fixmul_32x16b(x[m-i-2], sineWindow[m-i-2]),
shift));

```

```

>     x[m-2-i] = L_add(L_shr(fixmul_32x16b(xtmp, sineWindow[m-i-2]), shift),
>                      L_shr(fixmul_32x16b(x[m-i-2], sineWindow[i+1]), shift));
129a131
>
132c134
< static void postTwiddle (Word32 x [], Word32 m, Word32 step, const Word16
sineTable [], Word32 tableSize)
---
> static void postTwiddle (Word32 x [], Word16 m, Word16 step, const Word16
sineTable [], Word16 tableSize)
138,143c140,141
< Word16 lo1, lo2, hi3, lo3;
< Word16 m2 = shl((Word16)m, 1);
< Word16 *px_16Bit = (short *)x;
move16();
<
<
move32(); move32();
< count = shr((Word16)m, 1);
---
>
move16(); move16();
> count = shr(m, 1);
145c143
< if(step == 8)
---
> if(sub(step,8) == 0)
147c145,146
< for (i = 0, j = 0 ; i < count ; i+=2, j+= 4)
---
> j = 0; move16();
> for (i = 0; i < count ; i+=2, j+= 4)
149,152c148
< lo1 = logical_shr(px_16Bit[j], 1);
< lo2 = logical_shr(px_16Bit[j+2], 1);
< hi3 = px_16Bit[m2-j-1];
move16();
< lo3 = logical_shr(px_16Bit[m2-j-2],1);
---
> Word32 xtmp = x[m-1-i]; move32();
154,155c150,153
< x [m-1-i] = fixmsub_32x16b_Shift (px_16Bit[j + 1], lo1, *sinPtr,
px_16Bit[j+3], lo2, *cosPtr, 1); move32();
< x [i] = fixmadd_32x16b_Shift (px_16Bit[j + 1], lo1, *cosPtr,
px_16Bit[j+3], lo2, *sinPtr, 1); move32();
---
> x [m-1-i] = L_sub(L_shr(fixmul_32x16b(x[i], *sinPtr), 1),
>                  L_shr(fixmul_32x16b(x[i+1], *cosPtr), 1)); move32();
> x [i] = L_add(L_shr(fixmul_32x16b(x[i], *cosPtr), 1),
>              L_shr(fixmul_32x16b(x[i+1], *sinPtr), 1)); move32();
159,162c157,160
<
< lo1 = logical_shr(px_16Bit[m2-j-4], 1);
< x [i+1] = fixmsub_32x16b_Shift (px_16Bit[m2-j-3], lo1, *cosPtr, hi3,
lo3, *sinPtr, 1) ; move32();
< x [m-2-i] = fixmadd_32x16b_Shift (px_16Bit[m2-j-3], lo1, *sinPtr, hi3,
lo3, *cosPtr, 1) ; move32();
---
> x [i+1] = L_sub(L_shr(fixmul_32x16b(x[m-i-2], *cosPtr), 1),
>                L_shr(fixmul_32x16b(xtmp, *sinPtr), 1));
move32();
> x [m-2-i] = L_add(L_shr(fixmul_32x16b(x[m-i-2], *sinPtr), 1),

```

```

>                                     L_shr(fixmul_32x16b(xtmp, *cosPtr), 1));
move32();
167c165,166
<     for (i = 0, j = 0 ; i < count ; i+=2, j+= 4)
---
>     j = 0; move16();
>     for (i = 0; i < count ; i+=2, j+= 4)
169,172c168
<         lo1 = logical_shr(px_16Bit[j], 1);
<         lo2 = logical_shr(px_16Bit[j+2], 1);
<         hi3 = px_16Bit[m2-j-1];
move16();
<         lo3 = logical_shr(px_16Bit[m2-j-2],1);
---
>         Word32 xtmp = x[m-1-i];
move32();
174,175c170,173
<         x [m-1-i] = fixmsub_32x16b_Shift (px_16Bit[j + 1], lo1, *sinPtr,
px_16Bit[j+3], lo2, *cosPtr, 1); move32();
<         x [i] = fixmadd_32x16b_Shift (px_16Bit[j + 1], lo1, *cosPtr,
px_16Bit[j+3], lo2, *sinPtr, 1); move32();
---
>         x [m-1-i] = L_sub(L_shr(fixmul_32x16b(x[i], *sinPtr), 1),
>                             L_shr(fixmul_32x16b(x[i+1], *cosPtr), 1)); move32();
>         x [i] = L_add(L_shr(fixmul_32x16b(x[i], *cosPtr), 1),
>                         L_shr(fixmul_32x16b(x[i+1], *sinPtr), 1)); move32();
180,182c178,181
<         lo1 = logical_shr(px_16Bit[m2-j-4], 1);
<         x [i+1] = fixmsub_32x16b_Shift (px_16Bit[m2-j-3], lo1, *cosPtr, hi3,
lo3, *sinPtr, 1) ; move32();
<         x [m-2-i] = fixmadd_32x16b_Shift (px_16Bit[m2-j-3], lo1, *sinPtr, hi3,
lo3, *cosPtr, 1) ; move32();
---
>         x [i+1] = L_sub(L_shr(fixmul_32x16b(x[m-i-2], *cosPtr), 1),
>                         L_shr(fixmul_32x16b(xtmp, *sinPtr), 1));
move32();
>         x [m-2-i] = L_add(L_shr(fixmul_32x16b(x[m-i-2], *sinPtr), 1),
>                         L_shr(fixmul_32x16b(xtmp, *cosPtr), 1));
move32();
187c186
< static void scaledFFT (Word32 x [], Word32 ldn, const Word16 sineTable [],
Word32 tableSize)
---
> static void scaledFFT (Word32 x [], Word16 ldn, const Word16 sineTable [],
Word16 tableSize)
196c195
<     const Word16 n = shl(1, (Word16)ldn) ;
---
>     const Word16 n = shl(1, ldn) ;
201d199
<     Word16 *px_16Bit = (Word16 *)x; move16();
285c283
<     for (ldm = 3 ; ldm <= ldn ; ++ldm)
---
>     for (ldm = 3 ; ldm <= ldn ; ldm++)
289a288
>         Word16 c_index, s_index;
322c321,322
<         angleStep = shr(shl((Word16)tableSize, 2), ldm) ;
---
>         /* tableSize is known in advance, so no shl(tableSize,2) is necessary */
>         angleStep = shr(tableSize<<2, ldm) ;
325,328c325,326

```

```

<     for (j = 1 ; j < count ; ++j)
<     {
<         Word16 s, c;
<         Word16 lo1, lo2, t3;
---
>         s_index = angleStep; move16();          /* init counter: j*angleStep */
>         c_index = sub(tableSize, angleStep); /* init counter: tableSize -
j*angleStep */
330,331c328,329
<         s = sineTable [j * angleStep] ;                      move16();
<         c = sineTable [tableSize - j * angleStep] ;          move16();
---
>         for (j = 1 ; j < count ; j++)
>         {
338,342c336,339
<             t3 = shl(t2, 1);
<             lo1 = logical_shr(px_16Bit[t3], 1);
<             lo2 = logical_shr(px_16Bit[t3+2], 1);
<             vr = fixmadd_32x16b (px_16Bit[t3+1], lo1, c, px_16Bit[t3+3], lo2, s) ;
<             vi = fixmsub_32x16b (px_16Bit[t3+3], lo2, c, px_16Bit[t3+1], lo1, s) ;
---
>             vr = L_add(fixmul_32x16b(x[t2], sineTable[c_index]),
>                         fixmul_32x16b(x[t2+1], sineTable[s_index]));
>             vi = L_sub(fixmul_32x16b(x[t2+1], sineTable[c_index]),
>                         fixmul_32x16b(x[t2], sineTable[s_index]));
354,358c351,354
<             t3 = shl(t2, 1);
<             lo1 = logical_shr(px_16Bit[t3], 1);
<             lo2 = logical_shr(px_16Bit[t3+2], 1);
<             vi = fixmadd_32x16b (px_16Bit[t3+1], lo1, c, px_16Bit[t3+3], lo2, s) ;
<             vr = fixmsub_32x16b (px_16Bit[t3+3], lo2, c, px_16Bit[t3+1], lo1, s) ;
---
>             vi = L_add(fixmul_32x16b(x[t2], sineTable[c_index]),
>                         fixmul_32x16b(x[t2+1], sineTable[s_index]));
>             vr = L_sub(fixmul_32x16b(x[t2+1], sineTable[c_index]),
>                         fixmul_32x16b(x[t2], sineTable[s_index]));
363a360
>
364a362,363
>         c_index -= angleStep;
>         s_index += angleStep;

```

In the encoder and decoder, file **bitbuffer.c** apply the following patch:

```

19c19
<
test();
---
>
test(); sub(1,1);
23c23
<
test();
---
>
test(); sub(1,1);
109c109
<     for ( i = 0; i < bytesToGoSrc + 1; i++ ) {
---
>     for ( i = 0; i <= bytesToGoSrc; i++ ) {
136c136

```

```

<
test();
---
>
139a140
>   logic16(); logic16(); /* 8-bit aligned read access: *pReadNext */
142d142
<   while (hBitBuf->rBitPos < 0) {
143a144
>   while (hBitBuf->rBitPos < 0) {
145c146
<       hBitBuf->pReadNext++;
add(1,1); /* counting operation */
---
>       hBitBuf->pReadNext++;
add(1,1); /* counting operation */ move16();
147c148
<
test();
---
>
test(); sub(1,1);
152a154,155
>
>   logic16(); logic16(); /* 8-bit aligned read access: *pReadNext */
153a157,158
>   /* end of while */
test();
>
174a180
>
test();
175a182
>   Word16 bits_to_shift;
177d183
<
test();
179c185
<   bitsToWrite = min (add(hBitBuf->wBitPos, 1), noBitsToWrite);
test(); move16();
---
>   bitsToWrite = S_min (add(hBitBuf->wBitPos, 1), noBitsToWrite);
180a187,188
>   bits_to_shift = sub(add(hBitBuf->wBitPos, 1), bitsToWrite);
>   logic32(); /* cast (Word8) is equivalent to a mask */
182c190
<                               << sub(add(hBitBuf->wBitPos, 1), bitsToWrite));
L_shl(1,1); L_shr(1,1); L_shl(1,1); /* counting operations */
---
>                               << bits_to_shift);                               L_shl(1,1);
L_shr(1,1); L_shl(1,1); /* counting operations */
184c192
<                               << sub(add(hBitBuf->wBitPos, 1), bitsToWrite) );
L_shl(1,1); L_shl(1,1); /* counting operations */
---
>                               << bits_to_shift );                               L_shl(1,1);
L_shl(1,1); /* counting operations */
197c205
<
test();
---
>
test(); sub(1,1);

```

```

201a210
>      /* end of while */
test();
216c225
<      hBitBuf->rBitPos = sub(hBitBuf->rBitPos, offset);
---
>      hBitBuf->rBitPos = sub(hBitBuf->rBitPos, offset);

```

In the written specification, replace the following table:

“

Table A.1: Weighted MOPS figures with ACCOUNT_ETSIOP_OVERHEAD_SPLITWORD32 set

	Test Case	Mono Encoder	Stereo Encoder	Decoder	Decoder, mono only
wMOPS [average / worst frame]	14m	26.50 / 28.87	26.50 / 31.59	18.34 / 20.37	13.93 / 15.82
	18s	---	61.38 / 65.25	34.36 / 37.20	14.24 / 16.35
	24m	29.51 / 34.28	29.51 / 34.26	20.15 / 22.99	15.10 / 17.88
	24s	---	63.46 / 68.17	36.52 / 40.12	15.02 / 17.81
	32s	---	64.59 / 71.01	37.54 / 41.40	15.57 / 18.60
	48s	---	64.16 / 77.61	30.99 / 36.75	20.31 / 25.14

Table A.2: Weighted MOPS figures with ACCOUNT_ETSIOP_OVERHEAD_SPLITWORD32 not set

	Test Case	Mono Encoder	Stereo Encoder	Decoder	Decoder, mono only
wMOPS [average / worst frame]	14m	23.79 / 25.41	23.79 / 29.03	16.00 / 17.87	12.52 / 14.35
	18s	---	51.13 / 53.92	29.48 / 32.19	12.83 / 14.89
	24m	26.52 / 29.60	26.52 / 29.66	17.65 / 20.24	13.64 / 16.31
	24s	---	53.06 / 56.05	31.29 / 34.75	13.56 / 16.28
	32s	---	54.09 / 58.38	32.18 / 35.83	14.09 / 17.04
	48s	---	57.45 / 65.87	28.11 / 33.65	18.77 / 23.45

“

with

“

Table A.1: Weighted MOPS figures with ACCOUNT_ETSIOP_OVERHEAD_SPLITWORD32 set

	Test Case	Mono Encoder	Stereo Encoder	Decoder	Decoder, mono only
wMOPS [average / worst frame]	14m	26.51 / 28.87	26.50 / 31.61	19.15 / 21.20	14.73 / 16.80
	18s	---	61.38 / 65.25	35.18 / 38.04	15.14 / 17.39
	24m	29.51 / 34.28	29.51 / 34.26	20.98 / 23.84	15.93 / 18.74
	24s	---	63.47 / 68.17	37.35 / 40.98	15.93 / 18.72
	32s	---	64.61 / 71.02	38.39 / 42.28	16.47 / 19.60
	48s	---	64.17 / 77.63	32.65 / 38.46	21.96 / 26.83

Table A.2: Weighted MOPS figures with ACCOUNT ETSIOP OVERHEAD SPLITWORD32 not set

	<u>Test Case</u>	<u>Mono Encoder</u>	<u>Stereo Encoder</u>	<u>Decoder</u>	<u>Decoder, mono only</u>
<u>wMOPS</u> <u>[average /</u> <u>worst frame]</u>	<u>14m</u>	<u>23.80 / 25.41</u>	<u>23.79 / 29.04</u>	<u>15.86 / 17.73</u>	<u>12.38 / 14.38</u>
	<u>18s</u>	<u>---</u>	<u>51.14 / 53.92</u>	<u>29.35 / 32.07</u>	<u>12.77 / 14.95</u>
	<u>24m</u>	<u>26.52 / 29.60</u>	<u>26.52 / 29.66</u>	<u>17.53 / 20.13</u>	<u>13.51 / 16.21</u>
	<u>24s</u>	<u>---</u>	<u>53.08 / 56.06</u>	<u>31.17 / 34.64</u>	<u>13.57 / 16.22</u>
	<u>32s</u>	<u>---</u>	<u>54.11 / 58.39</u>	<u>32.08 / 35.75</u>	<u>14.05 / 17.09</u>
	<u>48s</u>	<u>---</u>	<u>57.46 / 65.89</u>	<u>27.88 / 33.43</u>	<u>18.53 / 23.22</u>

“

CR-Form-v7.1

CHANGE REQUEST

№ 26.411 CR 002 № rev - № Current version: 6.0.0 №

For **HELP** on using this form, see bottom of this page or look at the pop-up text over the № symbols.

Proposed change affects: UICC apps № ☐ ME ☒ Radio Access Network ☐ Core Network ☒

Title:	№ Correction to C-code: Avoid reading of non-initialized variable	
Source:	№ TSG SA WG4 Codec	
Work item code:	№ PSSrel6-Stage3	Date: № 06/06/2005
Category:	№ F 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 TR 21.900 .	Release: № Rel-6 Use <u>one</u> of the following releases: Ph2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) Rel-4 (Release 4) Rel-5 (Release 5) Rel-6 (Release 6) Rel-7 (Release 7)

Reason for change:	№ In rare cases a flag was set to a wrong value which lead to a non-initialised variable in a related code section.
Summary of change:	№ The initialisation of control flags for quantizer behaviour is corrected, such that reading of non-initialised variables later on is prevented.
Consequences if not approved:	№ Under certain circumstances and for rare cases the encoder will produce non predictable results.

Clauses affected:	№ C-Code appendix									
Other specs affected:	<table><tr><td>Y</td><td>N</td></tr><tr><td>№</td><td>X</td></tr><tr><td></td><td>X</td></tr><tr><td></td><td>X</td></tr></table>	Y	N	№	X		X		X	Other core specifications № Test specifications № O&M Specifications №
Y	N									
№	X									
	X									
	X									
Other comments:	№ The false behaviour was observed only for one clip out of a set of around 100 clips used for QAQC testing and only one configuration (14m) out of 11 tested configurations. All other configurations/clips remained bit-exact, i.e. the error did not manifest itself.									

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <http://www.3gpp.org/specs/CR.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.

Changes related to the C-Code Appendix:

In the encoder, file **adj_thr.c** line 254, replace (the additional condition is marked in yellow):

```
“
for (sfb=0; sfb<psyOutChan->maxSfbPerGroup; sfb++) {
    test(); test(); move16();
    if (L_sub(psyOutChan->sfbSpreadEnergy[sfbGrp+sfb], psyOutChan-
>sfbEnergy[sfbGrp+sfb]) > 0 ||
        L_sub(psyOutChan->sfbMinSnr[sfbGrp+sfb], MAX_16) == 0) {
        ahFlag[ch][sfbGrp+sfb] = NO_AH;
    }
    else {
        ahFlag[ch][sfbGrp+sfb] = AH_INACTIVE;
    }
}
”
```

with

```
“
for (sfb=0; sfb<psyOutChan->maxSfbPerGroup; sfb++) {
    test(); test(); test(); move16();
    if (L_sub(psyOutChan->sfbSpreadEnergy[sfbGrp+sfb], psyOutChan-
>sfbEnergy[sfbGrp+sfb]) > 0 ||
        L_sub(psyOutChan->sfbEnergy[sfbGrp+sfb], psyOutChan-
>sfbThreshold[sfbGrp+sfb]) <= 0 ||
        L_sub(psyOutChan->sfbMinSnr[sfbGrp+sfb], MAX_16) == 0) {
        ahFlag[ch][sfbGrp+sfb] = NO_AH;
    }
    else {
        ahFlag[ch][sfbGrp+sfb] = AH_INACTIVE;
    }
}
”
```

3GPP TSG-SA4 Meeting #35
San Diego, USA, May 09-13 2005

Tdoc #S4-050386

CR-Form-v7.1

CHANGE REQUEST

⌘ 26.411 CR 003 ⌘ rev - ⌘ Current version: 6.0.0 ⌘

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

Proposed change affects: UICC apps ☐ ME ☒ Radio Access Network ☐ Core Network ☒

Title:	⌘ Correction to C-code: fixes for error concealment (Rel 6)	
Source:	⌘ TSG SA WG4 Codec	
Work item code:	⌘ PSSrel6-Stage3	Date: ⌘ 06/06/2005
Category:	⌘ F	Release: ⌘ Rel-6
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 TR 21.900 .		Use <u>one</u> of the following releases: Ph2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) Rel-4 (Release 4) Rel-5 (Release 5) Rel-6 (Release 6) Rel-7 (Release 7)

Reason for change:	⌘ The error concealment implementation contained bugs which resulted in degraded output in case of frame errors. A wrong frameErrorFlag was passed to the decoder, energies could be interpreted wrongly during concealment and concealment of parametric stereo erroneously switched to mono.
Summary of change:	⌘ The bugs for error concealment are corrected.
Consequences if not approved:	⌘ The error concealment algorithm would not perform as expected.

Clauses affected:	⌘ C-Code appendix									
Other specs affected:	<table border="1"> <tr> <td>Y</td> <td>N</td> </tr> <tr> <td>☐</td> <td>☒</td> </tr> <tr> <td>☐</td> <td>☒</td> </tr> <tr> <td>☐</td> <td>☒</td> </tr> </table>	Y	N	☐	☒	☐	☒	☐	☒	Other core specifications ⌘ Test specifications ⌘ O&M Specifications ⌘
	Y	N								
	☐	☒								
☐	☒									
☐	☒									
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/specs/CR.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.

The fixed point reference decoder applied the frameErrorFlag to the wrong SBR frame in time, thus in **main.c, line 260, add:**

```
“
    Flag lastFrameOk = 1;
“
```

and in **line 470, add:**

```
“
    {
        Flag tmp = frameOk;
        frameOk = lastFrameOk;
        lastFrameOk = tmp;
    }
“
```

Energies could be interpreted wrongly during concealment, thus in **env_dec.c, replace line 293**

```
“
    if (sub(hHeaderData->ampResolution, SBR_AMP_RES_1_5) == 0) {
”
```

with

```
“
    if (sub(h_sbr_data->ampResolutionCurrentFrame, SBR_AMP_RES_1_5) == 0)
    {
“
```

Concealment of parametric stereo erroneously switched to mono, thus in **sbrdecoder.c, replace line 380**

```
“
    hHeaderData->channelMode = stereo ? STEREO : MONO;    test();
    move16();
“
```

with

```
“
    if (sub(hHeaderData->channelMode, UNDEFINED_CHANNEL_MODE) == 0) {
        test();
        hHeaderData->channelMode = stereo ? STEREO : MONO;    test();
        move16();
    }
“
```