GSM Audio Pocket Knife - Bug #2514

GSM HR encoding result does not match the reference

09/15/2017 02:39 PM - fixeria

Status:	Stalled	**Start date:**	09/15/2017
Priority:	Normal	**Due date:**	
Assignee:	fixeria	**% Done:**	10%
Category:			
Target version:			

Description

Implementing the automake transcoding tests, I found that GSM HR related encoding tests are always fail, while decoding from the reference files is ok.

Regression tests summary:

```
1: procqueue                                        ok
  2: io/pq_file                                       ok
  3: io/pq_rtp                                        ok
  4: conv/enc/amr_efr                                 ok
  5: conv/enc/gsm                                     ok
  6: conv/enc/racal_hr                                FAILED (testsuite.at:49)
  7: conv/enc/racal_fr                                ok
  8: conv/enc/racal_efr                               ok
  9: conv/enc/ti_hr                                   FAILED (testsuite.at:79)
 10: conv/enc/ti_fr                                   ok
 11: conv/enc/ti_efr                                  ok
 12: conv/enc/rtp_efr                                 ok
 13: conv/enc/rtp_hr_etsi                             FAILED (testsuite.at:119)
 14: conv/enc/rtp_hr_ietf                             FAILED (testsuite.at:129)
 15: conv/dec/amr_efr                                 ok
 16: conv/dec/gsm                                     ok
 17: conv/dec/racal_hr                                ok
 18: conv/dec/racal_fr                                ok
 19: conv/dec/racal_efr                               ok
 20: conv/dec/ti_hr                                   ok
 21: conv/dec/ti_fr                                   ok
 22: conv/dec/ti_efr                                  ok
 23: conv/dec/rtp_efr                                 ok
 24: conv/dec/rtp_hr_etsi                             ok
 25: conv/dec/rtp_hr_ietf                             ok
```

The same problem can be discovered using the built-in shell script named 'test_all_formats.sh'.

I have also attempted to compare the encoding results with the reference files and found, that the mismatching bytes are starting from 128th until 352 bytes.

History

#1 - 06/30/2018 03:18 PM - fixeria

- Status changed from New to In Progress

- Assignee set to fixeria

Looks like this is related to the problem:

```
$ cd gapk/tests/
$ valgrind --track-origins=yes ../src/.libs/lt-osmo-gapk -i ref-files/hhgttg_part1_5.s16 -f rawpcm-s16le -g racal-hr -o /dev/null -v

==29901== Memcheck, a memory error detector
```

```
==29901== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==29901== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright info
==29901== Command: ../src/.libs/lt-osmo-gapk -i ref-files/hhgttg_part1_5.s16 -f rawpcm-s16le -g racal-hr -o /d
ev/null -v
==29901==
<0000> app_osmo_gapk.c:405 Opening I/O streams
<0000> app_osmo_gapk.c:408 Using a file as source
<0000> app_osmo_gapk.c:435 Using a file as sink
<0000> app_osmo_gapk.c:526 Creating a processing queue
<0000> pq_file.c:111 PQ 'main': Adding file input (blk_len=320)
<0000> pq_format.c:66 PQ 'main': Adding conversion from rawpcm-s16le to canon (for codec pcm)
<0000> pq_codec.c:83 PQ 'main': Adding codec hr, encoding to format hr-ref-enc
<0000> pq_format.c:66 PQ 'main': Adding conversion from hr-ref-enc to canon (for codec hr)
<0000> pq_format.c:60 PQ 'main': Adding conversion from canon to racal-hr (for codec hr)
<0000> pq_file.c:125 PQ 'main': Adding file output (blk_len=14)
<0000> app_osmo_gapk.c:777 Init complete, starting processing queue...
==29901== Conditional jump or move depends on uninitialised value(s)
==29901==    at 0x637C655: saturate (mathhalf.c:92)
==29901==    by 0x637C655: sub (mathhalf.c:1979)
==29901==    by 0x637C655: findBestInQuantList (sp_frm.c:2227)
==29901==    by 0x638083E: aflat (sp_frm.c:416)
==29901==    by 0x6372E26: speechEncoder (sp_enc.c:212)
==29901==    by 0x6384D49: gsmhr_encode (libgsmhr.c:98)
==29901==    by 0x5273688: codec_hr_encode (codec_hr.c:53)
==29901==    by 0x5270CB9: osmo_gapk_pq_execute (procqueue.c:202)
==29901==    by 0x402122: run (app_osmo_gapk.c:650)
==29901==    by 0x402122: main (app_osmo_gapk.c:778)
==29901==  Uninitialised value was created by a stack allocation
==29901==    at 0x63804A0: aflat (sp_frm.c:236)
==29901==
==29901== Conditional jump or move depends on uninitialised value(s)
==29901==    at 0x637C65E: saturate (mathhalf.c:96)
==29901==    by 0x637C65E: sub (mathhalf.c:1979)
==29901==    by 0x637C65E: findBestInQuantList (sp_frm.c:2227)
==29901==    by 0x638083E: aflat (sp_frm.c:416)
==29901==    by 0x6372E26: speechEncoder (sp_enc.c:212)
==29901==    by 0x6384D49: gsmhr_encode (libgsmhr.c:98)
==29901==    by 0x5273688: codec_hr_encode (codec_hr.c:53)
==29901==    by 0x5270CB9: osmo_gapk_pq_execute (procqueue.c:202)
==29901==    by 0x402122: run (app_osmo_gapk.c:650)
==29901==    by 0x402122: main (app_osmo_gapk.c:778)
==29901==  Uninitialised value was created by a stack allocation
==29901==    at 0x63804A0: aflat (sp_frm.c:236)
==29901==
==29901== Conditional jump or move depends on uninitialised value(s)
==29901==    at 0x637C634: findBestInQuantList (sp_frm.c:2227)
==29901==    by 0x638083E: aflat (sp_frm.c:416)
==29901==    by 0x6372E26: speechEncoder (sp_enc.c:212)
==29901==    by 0x6384D49: gsmhr_encode (libgsmhr.c:98)
==29901==    by 0x5273688: codec_hr_encode (codec_hr.c:53)
==29901==    by 0x5270CB9: osmo_gapk_pq_execute (procqueue.c:202)
==29901==    by 0x402122: run (app_osmo_gapk.c:650)
==29901==    by 0x402122: main (app_osmo_gapk.c:778)
==29901==  Uninitialised value was created by a stack allocation
==29901==    at 0x63804A0: aflat (sp_frm.c:236)
==29901==
==29901== Conditional jump or move depends on uninitialised value(s)
==29901==    at 0x637C655: saturate (mathhalf.c:92)
==29901==    by 0x637C655: sub (mathhalf.c:1979)
==29901==    by 0x637C655: findBestInQuantList (sp_frm.c:2227)
==29901==    by 0x63809B8: aflat (sp_frm.c:468)
==29901==    by 0x6372E26: speechEncoder (sp_enc.c:212)
==29901==    by 0x6384D49: gsmhr_encode (libgsmhr.c:98)
==29901==    by 0x5273688: codec_hr_encode (codec_hr.c:53)
==29901==    by 0x5270CB9: osmo_gapk_pq_execute (procqueue.c:202)
==29901==    by 0x402122: run (app_osmo_gapk.c:650)
==29901==    by 0x402122: main (app_osmo_gapk.c:778)
==29901==  Uninitialised value was created by a stack allocation
==29901==    at 0x63804A0: aflat (sp_frm.c:236)
==29901==
==29901== Conditional jump or move depends on uninitialised value(s)
==29901==    at 0x637C65E: saturate (mathhalf.c:96)
==29901==    by 0x637C65E: sub (mathhalf.c:1979)
==29901==    by 0x637C65E: findBestInQuantList (sp_frm.c:2227)
==29901==    by 0x63809B8: aflat (sp_frm.c:468)
```

```
==29901==    by 0x6372E26: speechEncoder (sp_enc.c:212)
==29901==    by 0x6384D49: gsmhr_encode (libgsmhr.c:98)
==29901==    by 0x5273688: codec_hr_encode (codec_hr.c:53)
==29901==    by 0x5270CB9: osmo_gapk_pq_execute (procqueue.c:202)
==29901==    by 0x402122: run (app_osmo_gapk.c:650)
==29901==    by 0x402122: main (app_osmo_gapk.c:778)
==29901==  Uninitialised value was created by a stack allocation
==29901==    at 0x63804A0: aflat (sp_frm.c:236)
==29901==
==29901== Conditional jump or move depends on uninitialised value(s)
==29901==    at 0x637C634: findBestInQuantList (sp_frm.c:2227)
==29901==    by 0x63809B8: aflat (sp_frm.c:468)
==29901==    by 0x6372E26: speechEncoder (sp_enc.c:212)
==29901==    by 0x6384D49: gsmhr_encode (libgsmhr.c:98)
==29901==    by 0x5273688: codec_hr_encode (codec_hr.c:53)
==29901==    by 0x5270CB9: osmo_gapk_pq_execute (procqueue.c:202)
==29901==    by 0x402122: run (app_osmo_gapk.c:650)
==29901==    by 0x402122: main (app_osmo_gapk.c:778)
==29901==  Uninitialised value was created by a stack allocation
==29901==    at 0x63804A0: aflat (sp_frm.c:236)
==29901==
<0000> procqueue.c:206 PQ 'main': execution aborted: item 'source/file' returned -1
<0000> app_osmo_gapk.c:655 Processed 250 frames
<0000> app_osmo_gapk.c:467 Closing I/O streams
full talloc report on 'osmo-gapk root context' (total    870 bytes in   7 blocks)
    logging                        contains    847 bytes in   5 blocks (ref 0) 0x6cba200
        struct log_target              contains    198 bytes in   2 blocks (ref 0) 0x6cba680
            struct log_category            contains     38 bytes in   1 blocks (ref 0) 0x6cba7c0
        struct log_info                contains    648 bytes in   2 blocks (ref 0) 0x6cba2b0
            struct log_info_cat            contains    608 bytes in   1 blocks (ref 0) 0x6cba380
    .name                          contains     23 bytes in   1 blocks (ref 0) 0x6cba140
==29901==
==29901== HEAP SUMMARY:
==29901==     in use at exit: 1,574 bytes in 8 blocks
==29901==   total heap usage: 29 allocs, 21 frees, 10,518 bytes allocated
==29901==
==29901== LEAK SUMMARY:
==29901==    definitely lost: 0 bytes in 0 blocks
==29901==    indirectly lost: 0 bytes in 0 blocks
==29901==      possibly lost: 1,542 bytes in 7 blocks
==29901==    still reachable: 32 bytes in 1 blocks
==29901==         suppressed: 0 bytes in 0 blocks
==29901== Rerun with --leak-check=full to see details of leaked memory
==29901==
==29901== For counts of detected and suppressed errors, rerun with: -v
==29901== ERROR SUMMARY: 53386 errors from 6 contexts (suppressed: 0 from 0)
```

#2 - 06/30/2018 08:23 PM - fixeria

- % Done changed from 0 to 10

When compiled with AddressSanitizer, the whole libgsmhr API is broken:

```
# Any HR-related format will produce same output
$ src/osmo-gapk -a default -f rawpcm-s16le -g ti-hr -o /dev/null
<0000> app_osmo_gapk.c:405 Opening I/O streams
<0000> app_osmo_gapk.c:428 Using ALSA as source
<0000> app_osmo_gapk.c:435 Using a file as sink
<0000> app_osmo_gapk.c:526 Creating a processing queue
=================================================================
==4955==ERROR: AddressSanitizer: global-buffer-overflow on address 0x7f030d0150a2 at pc 0x0000004c88bc bp 0x7f
fd1c4a8590 sp 0x7ffd1c4a7d40
READ of size 7080 at 0x7f030d0150a2 thread T0
    #0 0x4c88bb  (/home/wmn/osmocom/gapk/src/.libs/lt-osmo-gapk+0x4c88bb)
    #1 0x7f030cdf6143  (/home/wmn/osmocom/gapk/libgsmhr/.libs/libgsmhr.so.0+0x83143)
    #2 0x7f030e869444  (/home/wmn/osmocom/gapk/src/.libs/libosmogapk.so.0+0x16444)
    #3 0x516090  (/home/wmn/osmocom/gapk/src/.libs/lt-osmo-gapk+0x516090)
    #4 0x7f030d964f44  (/lib/x86_64-linux-gnu/libc.so.6+0x21f44)
    #5 0x41b81b  (/home/wmn/osmocom/gapk/src/.libs/lt-osmo-gapk+0x41b81b)

0x7f030d0150a2 is located 62 bytes to the left of global variable 'siUpdPointer' defined in 'refsrc/dtx.c:71:1
1' (0x7f030d0150e0) of size 2
0x7f030d0150a2 is located 0 bytes to the right of global variable 'swVadFrmCnt' defined in 'refsrc/dtx.c:68:11
' (0x7f030d0150a0) of size 2
SUMMARY: AddressSanitizer: global-buffer-overflow (/home/wmn/osmocom/gapk/src/.libs/lt-osmo-gapk+0x4c88bb)
```

```
Shadow bytes around the buggy address:
  0x0fe0e19fa9c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fe0e19fa9d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fe0e19fa9e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fe0e19fa9f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fe0e19faa00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0fe0e19faa10: 00 00 00 00[02]f9 f9 f9 f9 f9 f9 02 f9 f9 f9
  0x0fe0e19faa20: f9 f9 f9 f9 00 00 00 00 f9 f9 f9 f9 00 00 00 00
  0x0fe0e19faa30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fe0e19faa40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0fe0e19faa50: 00 00 00 00 00 00 00 00 f9 f9 f9 f9 f9 f9 f9 f9
  0x0fe0e19faa60: 02 f9 f9 f9 f9 f9 f9 f9 02 f9 f9 f9 f9 f9 f9 f9
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==4955==ABORTING
```

#3 - 06/30/2018 09:45 PM - fixeria

Ah, most likely the AddressSanitizer is not happy because we actually dlopen the libgsmhr...

#4 - 07/02/2018 09:48 AM - fixeria

The reference to uninitialized memory happens twice in function sp_fpm.c/aflat().
I am still trying to narrow the scope of potential bug/error...

According to Valgrind's output, the uninitialized value is created by stack
allocation for the sp_fpm.c/aflat(). Then it's passed to the sp_fpm.c/findBestInQuantList(),
where it causes Valgrind to complain. This function is called twice:

```
...
413: /* find 4 best prequantizer levels */
416: findBestInQuantList(quantList, 4, bestPql);
...
465: /* find best quantizer vector for this segment, and save it */
468: findBestInQuantList(quantList, 1, bestQl);
...
```

Adding a few memset() calls at the begining of sp_fpm.c/aflat():

```
memset(&quantList, 0x00, sizeof(quantList));
memset(bestPql, 0x00, sizeof(bestPql));
memset(bestQl, 0x00, sizeof(bestQl));
```

helps to avoid Valgrind warnings, actually the only one is required:

```
memset(bestQl, 0x00, sizeof(bestQl));
```

But this doesn't affect the test results. I've tried to use different
filler bytes: 0x00, 0x66, 0x99, 0xff - no changes, the output file
is always the same, and doesn't match the reference :/

#5 - 07/02/2018 10:33 AM - fixeria

I just also tried to downgrade GAPK to commit 2ba67e8c9e7f8925e144bd40050aac2afa266383,

where the reference files and a simple shell-script for testing were introduced. The Half Rate encoding tests also fail there...

laforge do you remember, was it working fine when you commited this?

#6 - 07/02/2018 10:50 AM - laforge

On Mon, Jul 02, 2018 at 10:33:06AM +0000, fixeria [REDMINE] wrote:

> laforge do you remember, was it working fine when you commited this?

I don't remember, sorry. Let me say I would be surprised if I merged something that's known-broken. But I wouldn't exclude it from having happened :/

#7 - 04/03/2019 08:37 AM - fixeria

- Status changed from In Progress to Stalled