

libosmocode - Bug #4508

Viterbi Conv decoder performance

04/21/2020 09:32 AM - tnt

Status:	New	Start date:	04/21/2020
Priority:	High	Due date:	
Assignee:	Hoernchen	% Done:	0%
Category:	libosmocoding		
Target version:			
Spec Reference:			

Description

There seems to be some evidence that the viterbi decoder might not be performing as well as it should (vis-as-vis its ability to correct errors).

The tested code was a very short packet encoded with this code in tail-biting mode :

```
code = conv_gen.ConvolutionalCode(20,  
    [  
        ( conv_gen.poly(0,2,3,5,6), 1 ),  
        ( conv_gen.poly(0,1,2,3,6), 1 ),  
        ( conv_gen.poly(0,1,2,4,6), 1 ),  
    ],  
    name='lte'  
)
```

Decoding results with each time # of bit erasure, % failed packet, % of bit errors.

libosmocode:

```
0 0.00 0.00  
5 0.00 0.00  
10 0.00 0.00  
15 1.33 0.13  
20 1.33 0.13  
25 3.33 0.37  
30 14.00 1.83  
35 40.00 8.10  
40 89.33 40.30
```

HW decoder:

```
0 0.00 0.00  
5 0.00 0.00  
10 0.00 0.00  
15 0.00 0.00  
20 0.00 0.00  
25 0.00 0.00  
30 0.00 0.00  
35 2.67 1.43  
40 56.67 24.90
```

History

#1 - 11/14/2020 09:16 AM - laforge

- Category set to libosmocoding
- Assignee set to Hoernchen
- Priority changed from Normal to High

Let's increase priority of this issue and actually assign it to someone with related area of expertise.

[tnt](#): You compared the results of some FPGA hardware implementation with those of our software implementation? Do we know if this also happens in the convolutional decoding for GSM/GPRS/EGPRS? In the latter case, we should definitely investigate this ASAP.

Might also be a good idea to have unit tests that verify the error recovery matches the expected performance.

[tnt](#) if you have time/interest to work on this, sysmocom is happy to pay for it - but I somehow fear you have other priorities at the moment, hence I'm assigning it to [Hoernchen](#) for now.

#2 - 11/14/2020 06:47 PM - tnt

Yeah, I can look at it in ~ 1 month if Hoerchen hasn't gotten around to it by then.

I had a look back when I first reported the issue and spent about half a day and couldn't really see anything. Behavior was the same both with the "normal" and the "sse" version of the code despite them sharing almost nothing.

I don't know if the same thing is true for the normal GSM codes. Unfortunately trying a different code, using another termination mode and using different packet length isn't something trivial to change for the hardware decoder so it would take a good few hours to implement that to try.

#3 - 01/12/2021 05:11 PM - Hoernchen

[tnt](#) did you have time to look at this?