

LFR-FSW - Bug #601

Computation of HK_LFR_LE_CNT : sometimes erroneous

30/01/2016 05:53 PM - Veronique bouzid

Status:	Closed	Start date:	30/01/2016
Priority:	Normal	Due date:	
Assignee:	Veronique bouzid	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
revision:	r0		

Description

Lors de l'exécution du script /opt/VALIDATION_R3/lfrverif/LFR_SVS/SVS-0012/ccsds_time_code_format.py
Parfois le calcul du champ HK_LFR_LE_CNT n'est pas correct.

Ma question est donc
quand calcules tu ce champ HK_LFR_LE_CNT (au début ou à la fin de la génération de la TM_LFR_HK)?
est ce possible que quand tu generes une TM_LFR_HK , la tache soit interrompue et qu 'un des champs ait été incrementé

Voici l'extrait du fichier 2016_01_27-08_48_22-Extract.txt

```
08:47:14.123007, TM_LFR_HK, TIME=0x14250e78276b, HK_LFR_LE_CNT=121, HK_LFR_ME_CNT=0,  
HK_LFR_DPU_SPW_TICK_OUT_CNT=60, HK_LFR_DPU_SPW_LAST_TIMC=61, HK_LFR_TIMECODE_ERRONEOUS=0,  
HK_LFR_TIMECODE_MISSING=1, HK_LFR_TIMECODE_INVALID=0, HK_LFR_TIME_TIMECODE_IT=60,  
HK_LFR_TIME_NOT_SYNCHRO=0, HK_LFR_TIME_TIMECODE_CTR=60  
ici on calcule 1 + 60 + 60 = 121
```

```
08:47:15.123013, TM_LFR_HK, TIME=0x94250e79276b, HK_LFR_LE_CNT=121, HK_LFR_ME_CNT=0,  
HK_LFR_DPU_SPW_TICK_OUT_CNT=60, HK_LFR_DPU_SPW_LAST_TIMC=61, HK_LFR_TIMECODE_ERRONEOUS=0,  
HK_LFR_TIMECODE_MISSING=1, HK_LFR_TIMECODE_INVALID=0, HK_LFR_TIME_TIMECODE_IT=60,  
HK_LFR_TIME_NOT_SYNCHRO=1, HK_LFR_TIME_TIMECODE_CTR=60  
ici on calcule 1 + 60 + 1 + 60 = 122 au lieu de 121 dans le champ
```

```
08:47:16.122941, TM_LFR_HK, TIME=0x94250e7a94fe, HK_LFR_LE_CNT=124, HK_LFR_ME_CNT=0,  
HK_LFR_DPU_SPW_TICK_OUT_CNT=61, HK_LFR_DPU_SPW_LAST_TIMC=62, HK_LFR_TIMECODE_ERRONEOUS=0,  
HK_LFR_TIMECODE_MISSING=1, HK_LFR_TIMECODE_INVALID=0, HK_LFR_TIME_TIMECODE_IT=61,  
HK_LFR_TIME_NOT_SYNCHRO=1, HK_LFR_TIME_TIMECODE_CTR=61  
ici on calcule 1+61 +1 +61 = 124
```

```
08:47:17.122981, TM_LFR_HK, TIME=0x94250e7b9200, HK_LFR_LE_CNT=126, HK_LFR_ME_CNT=0,  
HK_LFR_DPU_SPW_TICK_OUT_CNT=62, HK_LFR_DPU_SPW_LAST_TIMC=63, HK_LFR_TIMECODE_ERRONEOUS=0,  
HK_LFR_TIMECODE_MISSING=1, HK_LFR_TIMECODE_INVALID=0, HK_LFR_TIME_TIMECODE_IT=62,  
HK_LFR_TIME_NOT_SYNCHRO=1, HK_LFR_TIME_TIMECODE_CTR=62  
ici on calcule 1+62 +1 +62 = 126
```

Les fichiers de test (2016_01_27-08_48_22*) se trouvent dans /home/validation/data/R3/3.0.0.14/1.1.89/SVS-0012

Contexte du test

FSW 3.0.0.14

VHDL 1.1.89

EM sans Timegen

SocExplorerEngine.getSocExplorer: Version = 0.6.2, Branch = default, Changeset = 819d0376d481

StarDundee

History

#1 - 01/02/2016 03:02 PM - paul leroy

- Assignee changed from paul leroy to Veronique bouzid

Good catch comme dirait l'autre. Dans la rev <= 3.0.0.15, j'ajoutais les compteurs avant de mettre à jour le champ hk_lfr_not_synchro.

Il est possible qu'un champ housekeeping change entre le moment où j'ajoute les compteurs et le moment où j'envoie le paquet HK mais c'est

peu probable. Je vais déplacer la fonction de calcul juste avant l'émission du paquet. Dans des cas limites, on pourrait retrouver le défaut, mais l'occurrence devrait être limitée.

See you in 3.0.0.16.

#2 - 03/02/2016 03:41 PM - Veronique bouzid

- *Status changed from New to Closed*

Le bug est corrigé en version 3.0.0.016.

Le script /opt/VALIDATION_R3/lfrverif/LFR_SVS/SVS-0012/ccsds_time_code_format.py a été rejoué.

L'analyse des hk a été modifiée pour détecter les erreurs de comptage et donc ce controle est effectué sur les tests de la campagne de validation.

Les fichiers de test (2016_02_03-14_56_26*) se trouvent dans /home/validation/data/R3/3.0.0.16/1.1.89/SVS-0012.

Contexte du test

FSW 3.0.0.16

VHDL 1.1.89

EM sans Timegen

SocExplorerEngine.getSocExplorer: Version = 0.6.2, Branch = default, Changeset = 819d0376d481

StarDundee