

LFR-FSW - Bug #657

HK_LFR_xE_CNT doesn't manage the wrap of 8bits counter error

09/03/2016 03:28 PM - Veronique bouzid

Status:	Closed	Start date:	09/03/2016
Priority:	Immediate	Due date:	
Assignee:	Veronique bouzid	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	0.00 hour
revision:	r0		
Description			
<p>En Analysant le passage à zero des compteurs d'erreurs suivants (HK_LFR_TIMECODE_MISSING et HK_LFR_TIME_NOT_SYNCHRO, j observe bien le passage de 255 vers 0 pour les compteurs d'erreur codés sur 8 bits mais l'effet non attendu est que les compteurs HK_LFR_xE_CNT qui eux sont codés sur 16 bits ne sont plus bons car ils representent la somme des erreurs liées à une criticité et donc le fait qu'un compteur passe à 0 diminue la valeur de ce compteur HK_LFR_xE_CNT</p> <p>Voici un extrait du fichier Detail.txt</p> <pre>15:54:46.122923, TM_LFR_HK, TIME=0x000000ff81da, HK_LFR_UPDATE_TIME_TC_CNT=255, HK_LFR_LE_CNT=509, HK_LFR_ME_CNT=0, HK_LFR_LAST_ER_RID: LE_LFR_TIME = 42119, HK_LFR_LAST_ER_CODE: NOT_SYNCHRO = 25, HK_LFR_LAST_ER_TIME=0x8000013bd189, HK_LFR_DPU_SPW_TICK_OUT_CNT=255, HK_LFR_DPU_SPW_LAST_TIMC=63, HK_LFR_DPU_SPW_PARITY=0, HK_LFR_DPU_SPW_DISCONNECT=0, HK_LFR_DPU_SPW_ESCAPE=0, HK_LFR_DPU_SPW_CREDIT=0, HK_LFR_DPU_SPW_WRITE_SYNC=0, HK_LFR_DPU_SPW_RX_AHB=0, HK_LFR_DPU_SPW_TX_AHB=0, HK_LFR_DPU_SPW_EARLY_EOP=0, HK_LFR_DPU_SPW_INVALID_ADDR=0, HK_LFR_DPU_SPW_EEP=0, HK_LFR_DPU_SPW_RX_TOO_BIG=0, HK_LFR_TIMECODE_ERRONEOUS=0, HK_LFR_TIMECODE_MISSING=255, HK_LFR_TIMECODE_INVALID=0, HK_LFR_TIME_TIMECODE_IT=0, HK_LFR_TIME_NOT_SYNCHRO=254, HK_LFR_TIME_TIMECODE_CTR=0</pre> <pre>15:54:47.122944, TM_LFR_HK, TIME=0x0000010081da, HK_LFR_UPDATE_TIME_TC_CNT=255, HK_LFR_LE_CNT=254, HK_LFR_ME_CNT=0, HK_LFR_LAST_ER_RID: LE_LFR_TIMEC = 42129, HK_LFR_LAST_ER_CODE: MISSING = 21, HK_LFR_LAST_ER_TIME=0x000001003270, HK_LFR_DPU_SPW_TICK_OUT_CNT=255, HK_LFR_DPU_SPW_LAST_TIMC=63, HK_LFR_DPU_SPW_PARITY=0, HK_LFR_DPU_SPW_DISCONNECT=0, HK_LFR_DPU_SPW_ESCAPE=0, HK_LFR_DPU_SPW_CREDIT=0, HK_LFR_DPU_SPW_WRITE_SYNC=0, HK_LFR_DPU_SPW_RX_AHB=0, HK_LFR_DPU_SPW_TX_AHB=0, HK_LFR_DPU_SPW_EARLY_EOP=0, HK_LFR_DPU_SPW_INVALID_ADDR=0, HK_LFR_DPU_SPW_EEP=0, HK_LFR_DPU_SPW_RX_TOO_BIG=0, HK_LFR_TIMECODE_ERRONEOUS=0, HK_LFR_TIMECODE_MISSING=0, HK_LFR_TIMECODE_INVALID=0, HK_LFR_TIME_TIMECODE_IT=0, HK_LFR_TIME_NOT_SYNCHRO=254, HK_LFR_TIME_TIMECODE_CTR=0</pre> <p>Ceci est donc valable pour n'importe quel compteur codé sur 8 bits et utilisé dans une somme.</p> <p>Les fichiers (2016_02_25-16_00_04-) sont dans /home/validation/data/R3/3.0.0.22/1.1.89/SVS-0078/not_synchro. Probleme remonté en lançant verif_fields sur le fichier Detail.txt. Le fichier sss_cp_eqs_050.txt met en evidence les observations et permet de valider les passages à zero des compteurs :</p> <pre>15:54:47.122944, TM_LFR_HK HK_LFR_LE_CNT: res=254; exp>=509 HK_LFR_TIMECODE_MISSING: res=0; exp>=255</pre> <pre>15:55:49.123074, TM_LFR_HK HK_LFR_DPU_SPW_TICK_OUT_CNT: res=0; exp>=255 HK_LFR_DPU_SPW_LAST_TIMC: res=0; exp>=63</pre> <pre>15:56:50.123381, TM_LFR_HK HK_LFR_LE_CNT: res=1; exp>=256 HK_LFR_TIME_NOT_SYNCHRO: res=0; exp>=255</pre> <p>Le script utilisé se nomme /opt/VALIDATION_R3/lfrverif/LFR_SVS/SVS-0078/update_time_not_synchro_cnt_wrap.py.</p> <p>Contexte du test</p> <p>-----</p> <p>FSW 3.0.0.22 VHDL 1.1.89 EM sans Timegen SocExplorerEngine.getSocExplorer: Version = 0.6.2, Branch = default, Changeset = 819d0376d481</p>			

History

#1 - 09/03/2016 05:49 PM - Veronique bouzid

- Description updated

#2 - 24/05/2016 10:43 PM - paul leroy

- Status changed from New to Feedback

- Assignee changed from paul leroy to Veronique bouzid

J'ai modifié la façon de gérer les compteurs dans fsw >= 3.1.0.1, il y avait une erreur et je ne tenais pas compte de la valeur précédente du compteur pour recalculer la nouvelle valeur. A suivre...

#3 - 22/12/2016 03:12 PM - Veronique bouzid

- Subject changed from *HK_LFR_xE_CNT doesn't manage the wrap of 8bits counter error* to *HK_LFR_xE_CNT doesn't manage the wrap of 8bits counter error*

#4 - 25/01/2017 02:33 PM - Veronique bouzid

- Assignee changed from Veronique bouzid to paul leroy

Bug non corrigé mais il y a du mieux.

Le script commence par generer des erreurs alimentant les 2 compteurs HK_LFR_TIMECODE_MISSING, HK_LFR_TIME_NOT_SYNCHRO.

Voici un extrait de la valeur des 3 compteurs HK_LFR_LE_CNT, HK_LFR_TIMECODE_MISSING, HK_LFR_TIME_NOT_SYNCHRO.

```
16:04:17.8784, TM_LFR_HK, TIME=0x800000bc8554, HK_LFR_LE_CNT=255, HK_LFR_TIMECODE_MISSING=128,  
HK_LFR_TIME_NOT_SYNCHRO=127
```

--

```
18:17:17.897872, TM_LFR_HK, TIME=0x000000ff53fb, HK_LFR_LE_CNT=508, HK_LFR_TIMECODE_MISSING=255,  
HK_LFR_TIME_NOT_SYNCHRO=253
```

--

```
18:18:20.898004, TM_LFR_HK, TIME=0x00000100047f, HK_LFR_LE_CNT=509, HK_LFR_TIMECODE_MISSING=255,  
HK_LFR_TIME_NOT_SYNCHRO=254
```

```
18:18:21.898016, TM_LFR_HK, TIME=0x00000101047e,+ HK_LFR_LE_CNT=509, HK_LFR_TIMECODE_MISSING=0,  
HK_LFR_TIME_NOT_SYNCHRO=254+
```

Cette dernière ligne montre le bug. Quand le compteur HK_LFR_TIMECODE_MISSING passe de 255 à 0, le compteur HK_LFR_LE_CNT aurait du s'incrémenter de 1.

--> Pas de prise en compte sur le wrap à 255 du compteur.

Le meme scénario est observé quand cette fois HK_LFR_TIME_NOT_SYNCHRO passe à 0

```
18:20:23.898146, TM_LFR_HK, TIME=0x0000013cb50e, HK_LFR_LE_CNT=511, HK_LFR_TIMECODE_MISSING=1,  
HK_LFR_TIME_NOT_SYNCHRO=255
```

```
18:20:24.898168, TM_LFR_HK, TIME=0x8000013db50e, HK_LFR_LE_CNT=511, HK_LFR_TIMECODE_MISSING=1,  
HK_LFR_TIME_NOT_SYNCHRO=0
```

--> au final on a observé 257 erreurs pour HK_LFR_TIMECODE_MISSING + 256 erreurs sur HK_LFR_TIME_NOT_SYNCHRO soit pour HK_LFR_LE_CNT = 513 erreurs. Il nous en manque 2 qui correspondent au passage à zéro des 2 compteurs.

Le script utilisé est /opt/VALIDATION_R3plus/lfrverif/LFR_SVS/SVS-0026/update_time_not_synchro_cnt_wrap.py

Les fichiers (2017_01_18-18_23_39*) sont rangés dans le répertoire /home/validation/data/R3+/3.1.0.6/3.1.91/SVS-0026 (avant on était dans SVS-0078).

Contexte du test

FSW 3.1.0.6

VHDL 3.1.91

PFM sans Timegen

SocExplorerEngine.getSocExplorer: Version = 0.7.0, Branch = 0.6, Changeset = c459540a6dbd+

StarDundee

#5 - 25/01/2017 02:53 PM - Veronique bouzid

- Priority changed from Normal to Immediate

#6 - 26/01/2017 08:00 AM - paul leroy

- Assignee changed from paul leroy to Veronique bouzid

J'ai l'impression que pour corriger ce bug, il faut juste que je change 255 en 256, c'est de là que vient mon décalage. Je mets ça dans la prochaine révision, fswf >= 3.1.0.7

```
void increment_hk_counter( unsigned char newValue, unsigned char oldValue, unsigned int *counter )
{
    int delta;

    delta = 0;

    if (newValue >= oldValue)
    {
        delta = newValue - oldValue;
    }
    else
    {
        delta = (255 - oldValue) + newValue;
    }

    *counter = *counter + delta;
}
```

#7 - 06/02/2017 10:49 AM - Veronique bouzid

- Status changed from Feedback to Closed

Bug corrigé.

les fichiers (2017_02_03-15_42_47*) sont rangés dans le répertoire /home/validation/data/R3+/3.1.0.7/1.1.91/SVS-0026.

La dernière HK indique comme valeur

HK_LFR_LE_CNT=519,

on vérifie

HK_LFR_TIMECODE_MISSING=4 soit 256+!=260

HK_LFR_TIME_NOT_SYNCHRO=3 soit 256+!=259

260+259=519 --> C est donc bon.

Contexte du test

FSW 3.1.0.7

VHDL 1.1.91

EM1 sans Timegen

SocExplorerEngine.getSocExplorer: Version = 0.7.0, Branch = 0.6, Changeset = c459540a6dbd+

StarDundee