

LFR-FSW - Bug #606

Gestion des erreurs: champ HK_LFR_LAST_ER_RID, HK_LFR_LAST_ER_CODE, HK_LFR_LAST_ER_TIME

03/02/2016 04:29 PM - Veronique bouzid

Status:	Closed	Start date:	03/02/2016
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

J'ai joué le script /opt/VALIDATION_R3/lfrverif/LFR_SVS/SVS-0064/spw_failure_standby.py

Ce test consiste à débrancher le connecteur du link1 du spacewire en mode standby.

On va s'intéresser à la chronologie des détections des erreurs, mixant erreurs de time et spw.

On démarre par une hk où seule l'erreur de time-code missing est décrite: c'est normal. On voit que la détection du time-code est à la fin de la séquence de boot.

```
14:57:30.929213, TM_LFR_HK, SEQUENCE_CNT=1, TIME=0x8000000348b9, HK_LFR_LE_CNT=1, HK_LFR_ME_CNT=0,
HK_LFR_HE_CNT=0, HK_LFR_LAST_ER_RID: LE_LFR_TIMEC = 42129, HK_LFR_LAST_ER_CODE: MISSING = 21,
HK_LFR_LAST_ER_TIME=0x800000022f11*, 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=1, HK_LFR_TIMECODE_INVALID=0,
HK_LFR_TIME_TIMECODE_IT=0, HK_LFR_TIME_NOT_SYNCHRO=0, HK_LFR_TIME_TIMECODE_CTR=0,
HK_LFR_BUFFER_DPU_TC_FIFO=0, HK_LFR_BUFFER_DPU_TM_FIFO=0, HK_LFR_AHB_CORRECTABLE=0,
HK_LFR_AHB_UNCORRECTABLE=0, SPARE=0x0
```

--
dernier hk avant d'enlever le link

```
14:57:36.527587, TM_LFR_HK, SEQUENCE_CNT=7, TIME=0x8000000948ba, HK_LFR_LE_CNT=1, HK_LFR_ME_CNT=0,
HK_LFR_HE_CNT=0, HK_LFR_LAST_ER_RID: LE_LFR_TIMEC = 42129, HK_LFR_LAST_ER_CODE: MISSING = 21,
HK_LFR_LAST_ER_TIME=0x800000022f11, 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=1, HK_LFR_TIMECODE_INVALID=0,
HK_LFR_TIME_TIMECODE_IT=0, HK_LFR_TIME_NOT_SYNCHRO=0, HK_LFR_TIME_TIMECODE_CTR=0,
HK_LFR_BUFFER_DPU_TC_FIFO=0, HK_LFR_BUFFER_DPU_TM_FIFO=0, HK_LFR_AHB_CORRECTABLE=0,
HK_LFR_AHB_UNCORRECTABLE=0, SPARE=0x0
```

reprise après la remise en service du câble

```
14:57:43.056605, TM_LFR_HK, SEQUENCE_CNT=9, TIME=0x8000000b48b8, HK_LFR_LE_CNT=3, HK_LFR_ME_CNT=0,
HK_LFR_HE_CNT=0, HK_LFR_LAST_ER_RID: LE_LFR_TIMEC = 42129, HK_LFR_LAST_ER_CODE: MISSING = 21,
HK_LFR_LAST_ER_TIME=0x800000022f11, HK_LFR_DPU_SPW_PARITY=1, HK_LFR_DPU_SPW_DISCONNECT=1,
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=1, HK_LFR_TIMECODE_INVALID=0,
HK_LFR_TIME_TIMECODE_IT=0, HK_LFR_TIME_NOT_SYNCHRO=0, HK_LFR_TIME_TIMECODE_CTR=0,
HK_LFR_BUFFER_DPU_TC_FIFO=0, HK_LFR_BUFFER_DPU_TM_FIFO=0, HK_LFR_AHB_CORRECTABLE=0,
HK_LFR_AHB_UNCORRECTABLE=0, SPARE=0x0
```

La on voit que l'on a perdu une HK sequence_cnt=8 avec le TIME=0x8000000axxxx.

Ensuite il y a apparition de 2 erreurs HK_LFR_DPU_SPW_PARITY=1, HK_LFR_DPU_SPW_DISCONNECT=1, détection cohérente avec le compteur HK_LFR_LE_CNT=3 mais pas avec la section qui décrit la dernière erreur détectée:

```
HK_LFR_LAST_ER_RID: LE_LFR_TIMEC = 42129, HK_LFR_LAST_ER_CODE: MISSING = 21,
HK_LFR_LAST_ER_TIME=0x800000022f11
```

Celle-ci correspond à l'absence de time-out détectée après la séquence de boot.

--> POURQUOI

Les fichiers de test (2016_02_03-14_57_55) se trouvent dans /home/validation/data/R3/3.0.0.16/1.1.89/SVS-0064.

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

History

#1 - 03/02/2016 04:30 PM - Veronique bouzid

- Description updated

#2 - 03/02/2016 04:33 PM - Veronique bouzid

- Description updated

#3 - 03/02/2016 04:43 PM - paul leroy

Oups, je viens de regarder mon code. J'ai écrit une fonction pour détecter une modif dans les stats SpaceWire et changer le champ hk_lfr_last_er, mais j'ai oublié de l'utiliser... Je l'ajoute et je te fais une 3.0.0.18 dans la foulée.

#4 - 04/02/2016 08:56 AM - Veronique bouzid

- Status changed from New to Feedback

Bug corrigé en 3.0.0.18.

Le script rejoué est /opt/VALIDATION_R3/lfrverif/LFR_SVS/SVS-0064/spw_failure_standby.py

Après l'opération débrancher le câble du link 1 et après 5s remettre en place le câble voici ce que j'observe dans les fichiers de test

la dernière HK avant l'enlèvement du câble

```
08:22:41.553099, TM_LFR_HK, SEQUENCE_CNT=10, TIME=0x800000c48da, HK_LFR_LE_CNT=1, HK_LFR_ME_CNT=0,
HK_LFR_HE_CNT=0, HK_LFR_LAST_ER_RID: LE_LFR_TIMEC = 42129, HK_LFR_LAST_ER_CODE: MISSING = 21,
HK_LFR_LAST_ER_TIME=0x800000022f33, 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=1,
HK_LFR_TIMECODE_INVALID=0, HK_LFR_TIME_TIMECODE_IT=0, HK_LFR_TIME_NOT_SYNCHRO=0, HK_LFR_TIME_TIMECODE_CTR=0,
HK_LFR_BUFFER_DPU_TC_FIFO=0, HK_LFR_BUFFER_DPU_TM_FIFO=0, HK_LFR_AHB_CORRECTABLE=0,
HK_LFR_AHB_UNCORRECTABLE=0, SPARE=0x0
```

Après la remise en service du câble:

```
08:22:49.022261, TM_LFR_HK, SEQUENCE_CNT=12, TIME=0x800000e48d9, HK_LFR_LE_CNT=2, HK_LFR_ME_CNT=1,
HK_LFR_HE_CNT=0, HK_LFR_LAST_ER_RID: ME_LFR_DPU_SPW = 42338, HK_LFR_LAST_ER_CODE: EARLY_EOP_EEP = 6,
HK_LFR_LAST_ER_TIME=0x8000000dc136*, HK_LFR_DPU_SPW_PARITY=0, HK_LFR_DPU_SPW_DISCONNECT=1,
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=1, 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=1,
HK_LFR_TIMECODE_INVALID=0, HK_LFR_TIME_TIMECODE_IT=0, HK_LFR_TIME_NOT_SYNCHRO=0, HK_LFR_TIME_TIMECODE_CTR=0,
HK_LFR_BUFFER_DPU_TC_FIFO=0, HK_LFR_BUFFER_DPU_TM_FIFO=0, HK_LFR_AHB_CORRECTABLE=0,
HK_LFR_AHB_UNCORRECTABLE=0, SPARE=0x0
```

Observations

1- On a perdu une HK (SEQUENCE_CNT=11 avec un TIME=0x800000d????

2- 2 erreurs sont apparues en plus

HK_LFR_DPU_SPW_DISCONNECT=1 --> comptabilisée dans HK_LFR_LE_CNT qui est devenu 2

HK_LFR_DPU_SPW_EARLY_EOP=1 --> comptabilisée dans HK_LFR_ME_CNT qui est devenu 1

C'est cette erreur qui est la dernière erreur tracée dans les champs

HK_LFR_LAST_ER_RID: ME_LFR_DPU_SPW = 42338,

HK_LFR_LAST_ER_CODE: EARLY_EOP_EEP = 6,

HK_LFR_LAST_ER_TIME=0x8000000dc136

Et si on regarde le temps de HK_LFR_LAST_ER_TIME, il correspond à la HK que nous avons perdu (0x8000000dc136)

--> Je n'ai pas retrouvé exactement le contexte du test effectué en 3.0.0.16 mais ce que j'observe au niveau de la gestion des erreurs est exacte.

Paul analyse les erreurs du spacewire dans l'ordre de la description des HK
(HK_LFR_DPU_SPW_PARITY, HK_LFR_DPU_SPW_DISCONNECT, HK_LFR_DPU_SPW_ESCAPE, HK_LFR_DPU_SPW_CREDIT,
HK_LFR_DPU_SPW_WRITE_SYNC, HK_LFR_DPU_SPW_RX_AHB, HK_LFR_DPU_SPW_TX_AHB, HK_LFR_DPU_SPW_EARLY_EOP,
HK_LFR_DPU_SPW_INVALID_ADDR, HK_LFR_DPU_SPW_EEP, HK_LFR_DPU_SPW_RX_TOO_BIG)

et donc le dernier champ en erreur trouvé est celui déclaré dans les champs dédiés.

--> Paul, je te laisse valider le test et répondre à la question

- POURQUOI PERD ON UNE HK?

Cela semble récurrent au vu des tests joués et donc il nous faudra l'identifier dans notre SRS.

Les fichiers de test (2016_02_04-08_23_01*) sont dans le répertoire /home/validation/data/R3/3.0.0.18/1.1.89/SVS-0064.

Contexte du test

FSW 3.0.0.18

VHDL 1.1.89

EM sans Timegen

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

StarDundee

#5 - 05/02/2016 07:56 AM - paul leroy

- Assignee changed from paul leroy to Veronique bouzid

Pour la perte du HK, je ne suis pas certain qu'il faille incriminer LFR, ça peut venir de la brique. Enlever le câble, c'est un peu violent, pour la brique, pour son driver sur le PC de test, pour LFR. Donc tu peux noter la perte d'un paquet dans le compte-rendu sur le test, mais pas forcément conclure sur l'endroit où celui-ci est perdu.

La perte d'un paquet pourrait venir d'un arrachage du câble pendant l'émission du paquet côté LFR (je ne suis pas sûr que le système essaie de relancer la transmission si elle n'a pas abouti), ou de la perte du paquet entre le moment où il est dans la brique et le moment où il va vers le PC de test.

#6 - 10/02/2016 09:33 AM - Veronique bouzid

L'utilisation du banc de test du LESIA simulateur mk2 pourra nous permet de mieux observer si on perd ou pas une HK car le matériel n'est pas intrusif.

#7 - 22/02/2016 04:25 PM - Veronique bouzid

Une information concernant la gestion des erreurs liées à la lecture de la structure spw_stats

Comme indiqué par Paul, seul le premier champ de la structure (tx_link_err) ne donne pas lieu à une remontée d'erreurs.

En effet les 2 champs suivants

```
unsigned int rx_rmap_header_crc_err; // NOT IN HK
```

```
unsigned int rx_rmap_data_crc_err; // NOT IN HK
```

ne sont pas comptabilisés car ne correspondent pas à un champ dans les HK mais par contre, ils donnent lieu à une mise à jour des 3 champs HK_LFR_LAST_ER_RID, HK_LFR_LAST_ER_CODE, HK_LFR_LAST_ER_TIME :

```
HK_LFR_LAST_ER_RID = RID_LE_LFR_DPU_SPW
```

```
HK_LFR_LAST_ER_CODE = CODE_HEADER_CRC
```

```
HK_LFR_LAST_ER_TIME = heure de la détection de l'erreur
```

De plus le compteur d'erreur HK_LFR_LE_CNT sera incrémenté.

#8 - 21/06/2018 01:34 PM - Veronique bouzid

- Status changed from Feedback to Closed