

## LFR-FSW - Bug #658

### HK\_LFR\_TIMECODE\_INVALID is set after the boot sequence

09/03/2016 05:47 PM - Veronique bouzid

<b>Status:</b>	Closed	<b>Start date:</b>	09/03/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Veronique bouzid	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	0.00 hour
<b>revision:</b>	r0		

#### Description

Voici une configuration qui apres la sequence de boot positionne le champ HK\_LFR\_TIMECODE\_INVALID à 1 au lieu du champ HK\_LFR\_TIMECODE\_MISSING.

Voici le scénario

- je lance le script /home/validation/SCRIPT/R3/genere\_invalid\_timecode.py qui envoie des timecodes.

- je sors de soc explorer

--> les fichiers de log sont (2016\_03\_09-17\_32\_11-\*)

+ fichier 2016\_03\_09-17\_32\_11-Nb.txt+ :

17:31:47.318532,ensure FSW has been reseted before this TEST CASE. Check 0x80000514

17:31:50.31973,PROXY LOAD is True

17:31:55.326809,Conditions initiales TIMECNT = 0

fichier 2016\_03\_09-17\_32\_11-Detail.txt :

```
on voit que la premiere HK contient HK_LFR_TIMECODE_MISSING=1, HK_LFR_TIMECODE_INVALID=0
```

```
HK_LFR_DPU_SPW_TICK_OUT_CNT=0, HK_LFR_DPU_SPW_LAST_TIMC=0
```

- je relance soc explorer sans avoir resetter la carte . ce reset met à zero le registre 0x80000514 (TIMECNT = dernier timecode valid recu

- je lance à nouveau le script script /home/validation/SCRIPT/R3/genere\_invalid\_timecode.py

--> les fichiers de log sont (2016\_03\_09-17\_41\_15-\*)

+ fichier 2016\_03\_09-17\_41\_15-Nb.txt+ :

17:40:50.878657,ensure FSW has been reseted before this TEST CASE. Check 0x80000514

17:40:53.879778,PROXY LOAD is True

17:40:58.894728,Conditions initiales TIMECNT = 11

17:40:58.895438,Script will be failed 11

+ fichier 2016\_03\_09-17\_41\_15-Detail.txt+ :

```
on voit que la premiere HK contient HK_LFR_TIMECODE_MISSING=0, HK_LFR_TIMECODE_INVALID=1
```

```
HK_LFR_DPU_SPW_TICK_OUT_CNT=0, HK_LFR_DPU_SPW_LAST_TIMC=0
```

As tu une explication à ce comportement? Je l'ai mis en Bug pour l'instant, à declasser si besoin en support pour renseigner ce comportement dans la SRS ou le SUM.

Contexte du test

-----

FSW 3.0.0.22

VHDL 1.1.89

EM sans Timegen

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

StarDundee

#### History

#1 - 09/03/2016 05:48 PM - Veronique bouzid

- Description updated

#2 - 29/04/2016 11:03 AM - paul leroy

On peut expliquer ce comportement. Au redémarrage de SocExplorer et du soft, l'IP SpaceWire n'est pas redémarrée et donc la dernière valeur

valide est toujours dans les registres du FPGA. A la fin de la tâche INIT, le timer qui permet de détecter un `timecode_missing` démarre et arrive à échéance 2s après. La routine compare le dernier `timecode` valide avec une valeur courante interne au soft de vol et identifie un `timecode_invalid`.

Dans `fsw_init.c`:

```
rtms_task Init( rtms_task_argument ignored )
{
    [...]

    // start the timer to detect a missing spacewire timecode
    // the timeout is larger because the spw IP needs to receive several valid timecodes before generating a t
    ickout
    // if a tickout is generated, the timer is restarted
    status = rtms_timer_fire_after( timecode_timer_id, TIMECODE_TIMER_TIMEOUT_INIT, timecode_timer_routine, N
    ULL );

    grspw_timecode_callback = &timecode_irq_handler;

    status = rtms_task_delete(RTEMS_SELF);
}

```

Dans `fsw_spacewire.c`:

```
rtms_timer_service_routine timecode_timer_routine( rtms_id timer_id, void *user_data )
{
    static unsigned char initStep = 1;

    unsigned char currentTimecodeCtr;

    currentTimecodeCtr = (unsigned char) (grspwPtr[0] & TIMECODE_MASK);

    if (initStep == 1)
    {
        if (currentTimecodeCtr == previousTimecodeCtr)
        {
            //*****
            // HK_LFR_TIMECODE_MISSING
            // the timecode value has not changed, no valid timecode has been received, the timecode is MISSIN
            G
            increase_unsigned_char_counter( &housekeeping_packet.hk_lfr_timecode_missing );
            update_hk_lfr_last_er_fields( RID_LE_LFR_TIMEC, CODE_MISSING );
        }
        else if (currentTimecodeCtr == (previousTimecodeCtr+1))
        {
            // the timecode value has changed and the value is valid, this is unexpected because
            // the timer should not have fired, the timecode_irq_handler should have been raised
        }
        else
        {
            //*****
            // HK_LFR_TIMECODE_INVALID
            // the timecode value has changed and the value is not valid, no tickout has been generated
            // this is why the timer has fired
            increase_unsigned_char_counter( &housekeeping_packet.hk_lfr_timecode_invalid );
            update_hk_lfr_last_er_fields( RID_LE_LFR_TIMEC, CODE_INVALID );
        }
    }
    else
    {
        initStep = 1;
        //*****
        // HK_LFR_TIMECODE_MISSING
        increase_unsigned_char_counter( &housekeeping_packet.hk_lfr_timecode_missing );
        update_hk_lfr_last_er_fields( RID_LE_LFR_TIMEC, CODE_MISSING );
    }

    rtms_event_send( Task_id[TASKID_DUMB], RTEMS_EVENT_13 );
}

```

#3 - 29/04/2016 11:15 AM - paul leroy

- Status changed from New to Resolved

- Assignee changed from paul leroy to Veronique bouzid

**#4 - 02/06/2016 09:02 AM - Veronique bouzid**

- Status changed from Resolved to Closed

Ce comportement etant lié au banc d'essai, je clos ce point.

Dans mes tests, j'ai identifié les tests qui ont besoin d'un reset de la carte pour que l'enchainement des time-codes soit nominal.