LFR-FSW - Support #627

Commutation immediate

14/02/2016 10:10 AM - Veronique bouzid

Status:	Closed	**Start date:**	14/02/2016
Priority:	Urgent	**Due date:**	
Assignee:	Veronique bouzid	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:		**Spent time:**	0.00 hour
revision:	r0		

Description

_
Sur le test joué en 3.0.0.22, je regarde la commutation
NORMAL vers SBM1 car plus facile à valider vu que les produits CWF_F1 sont tout de suite disponibles.

La commutation demandée est immediate, ce qui signifie qu'elle sera effective sur le tick-out d'apres = la seconde suivante.
2016 2 13 11:58:53:343 TM_LFR_HK time = 0x 1e51d6fc 6174
2016 2 13 11:58:53:512 TM_LFR_TC_EXE_SUCCESS time =* 0x 1e51d6fc 8cf1*
2016 2 13 11:58:53:770 TM_LFR_SCIENCE_SBM1_CWF_F1 time =* 0x 1e51d6fc 259d*

Je m'attends donc à voir arriver un produit CWF_F1 proche de 0x 1e51d6fd voire meme au mieux 1e51d6fc eaff (ffff-1500) mais comme on doit avoir 8 buffers, le calcul
doit etre FFFF-A800=557F.
Ce n'est pas le cas, on a l'impression même que la commutation a été immédiate.

1er groupe
2016 2 13 11:58:53:770 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc 259d
2016 2 13 11:58:53:782 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc 3a9d
2016 2 13 11:58:53:782 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc 4f9d
2016 2 13 11:58:53:794 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc 649d
2016 2 13 11:58:53:794 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc 799d
2016 2 13 11:58:53:804 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc 8e9d
2016 2 13 11:58:53:808 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc a39d
2016 2 13 11:58:53:808 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc b89d

2eme groupe (A800 fine-time plus loin)
2016 2 13 11:58:54:426 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc cd9d /A800) soit 43008 fine-time
2016 2 13 11:58:54:438 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc e29d
2016 2 13 11:58:54:438 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fc f79d
2016 2 13 11:58:54:450 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fd c9d -> la on a franchi la seconde / commutation demandée
2016 2 13 11:58:54:450 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fd 219d
2016 2 13 11:58:54:454 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fd 369d
2016 2 13 11:58:54:467 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fd 4b9d
2016 2 13 11:58:54:467 TM_LFR_SCIENCE_SBM1_CWF_F1 time = 0x 1e51d6fd 609d
2016 2 13 11:58:54:503 TM_LFR_SCIENCE_SBM1_BP1_F0 time = 0x 1e51d6fd 3da1
2016 2 13 11:58:54:735 TM_LFR_SCIENCE_SBM1_BP1_F0 time = 0x 1e51d6fd 7da1

donc on aurait du commencer à voir seulement les produits du 2eme groupe

--> EST CE QUE MON ANALYSE EST CORRECTE?

Autre explication:
Tu prends le premier groupe car tu as deja le buffer qui est disponible et tu estimes compatible avec le temps de commutation.

Le second buffer arrive trop loin de la seconde soit 0x 1e51d6fc cd9d + A800 = 0x 1e51d6fd bda1.

QUELLE EST LA BONNE ANALYSE?

Il faut vraiment documenter cette partie pour renseigner la SRS et eventuellement le SUM._

Related issues:

Related to Task #624: Long test with SBM1 and SBM2 FSW 3.0.0.22	**Closed**	**12/02/2016**
Related to Task #632: Mise à jour document SRS / commutation immediate	**Closed**	**19/02/2016**

History

#1 - 14/02/2016 10:15 AM - Veronique bouzid

- Related to Task #620: 3.0.0.22 added

#2 - 14/02/2016 10:16 AM - Veronique bouzid

- Related to Task #624: Long test with SBM1 and SBM2 FSW 3.0.0.22 added

#3 - 14/02/2016 10:16 AM - Veronique bouzid

- Related to deleted (Task #620: 3.0.0.22)

#4 - 15/02/2016 04:33 PM - bruno katra

Je rejoins l'analyse de Véronique, d'après ce que nous avons expliqué dans le document fourni à Milan le mois dernier et le nouveau fonctionnement du ENTER_MODE(0), on s'attend à ce que le premier echantillon CWF_F1 soit proche de 1e51d6fd 0000 à 0.2ms près or là le premier échantillon est antérieur de plus de 800ms.

#5 - 19/02/2016 08:24 AM - Veronique bouzid

- Related to Task #632: Mise à jour document SRS / commutation immediate added

#6 - 23/02/2016 11:09 AM - paul leroy

- Assignee changed from paul leroy to Veronique bouzid

Effectivement, il y a des choses à préciser. Actuellement, dans le code, les CWF sont envoyées dès qu'elles sont prêtes, indépendemment de la date de commutation. C'est dû au fait que la variable locale lfrCurrentMode est mise à jour dès la fin de l'exécution de la TC de changement de mode, ce qui provient de la façon historique de commuter (on considérait qu'on avait effectivement changé de mode à la fin de la TC même si les acquisitions commençaient plus tard, à la date demandée).

Prenons l'exemple d'une commutation en SBM1: une fois que lfrCurrentMode = SBM1, dès qu'un buffer CWF1 est plein, il est envoyé. Précision: pour les BP, je fais un test sur la date de transition. Si la date locale est supérieure à la date de transition demandée, j'expédie les données (indépendamment de l'acquisition time), sinon, je les oublie. Je peux faire la même chose pour les CWF mais même en faisant ça, il y a de forte chance pour que j'expédie des données qui ont commencé à être enregistrée avant la date de commutation. Ca permettra de s'assurer néanmoins que dans le cas limite où un buffer serait prêt entre l'exécution de la TC de commutation et la date de commutation demandée, il soit oublié.

Toutes ces mises au point viennent du fait que la stratégie de commutation de mode a complètement changé entre l'époque où nous avons mis au point la-dite commutation, après avoir interrogé précisément Philippe, et maintenant. Le VHDL était adapté à l'ancienne version (redémarrage de tous les traitement pour chaque commutation à une date précise). J'ai bricolé le soft pour m'adapter à la commutation actuelle, mais c'était pas rigoureusement prévu pour et donc il y a des effets collatéraux.

#7 - 04/03/2016 11:00 AM - bruno katra

- Status changed from New to In Progress

J'ai bien lu l'explication de Paul. En effet, c'est plus compliqué que ce qu'on avait compris.
J'ai plusieurs questions et remarques :
- ce comportement sera t-il décrit dans le SDD ?
- cela concerne t'il tous les CWF ?
- Je suis un peu embêté car cela invalide en partie le document de caractérisation envoyé à Milan fin janvier sur la datation qui est cosigné par Thomas, Paul, Bruno et qui doit servir de base de discussions pour le system level RPW...

#8 - 04/03/2016 11:04 AM - bruno katra

- Assignee changed from Veronique bouzid to paul leroy

#9 - 24/05/2016 10:47 PM - paul leroy

Il faudrait que le comportement soit décrit quelquepart. Je vais essayer de mettre une note dans le SDD pour expliquer comment j'ai codé le changement de mode.

#10 - 24/05/2016 11:35 PM - bruno katra

2 solutions :

- tu avais parlé de ça il y a quelques mois :
pour les BP, je fais un test sur la date de transition. Si la date locale est supérieure à la date de transition demandée, j'expédie les données (indépendamment de l'acquisition time), sinon, je les oublie. Je peux faire la même chose pour les CWF mais même en faisant ça, il y a de forte

chance pour que j'expédie des données qui ont commencé à être enregistrée avant la date de commutation. Ça permettra de s'assurer néanmoins que dans le cas limite où un buffer serait prêt entre l'exécution de la TC de commutation et la date de commutation demandée, il soit oublié.
Si tu implémentes ça, alors on est cohérent avec le document fourni à Milan et avec la RFD émise et acceptée concernant la commutation immédiate. Le SUM contient déjà un paragraphe expliquant l'acquisition et la commutation immédiate. Il restera juste à expliquer un peu ça dans le SDD

- Si tu laisses tel que c'est : il faut que l'on émette une RFD + décrire ce comportement dans le SUM (pour des raisons/contraintes opérationnelles) + SDD à mettre à jour avec ce mécanisme et la relation avec le VHDL.

#11 - 25/05/2016 11:19 AM - paul leroy

fsw >= 3.1.0.1
J'ai ajouté un test sur la date de transition du mode. Les CWF1, CWF2 et les BP ne sont envoyé que si la date de transition est respectée:

```
// data are sent depending on the transition time
                if ( time_management_regs->coarse_time >= lastValidEnterModeTime)
                {
                    status =  rtems_message_queue_send( queue_id, &ring_node_to_send, sizeof( ring_node* ) );
                }
```

#12 - 25/05/2016 11:20 AM - paul leroy

- Assignee changed from paul leroy to Veronique bouzid

#13 - 25/05/2016 12:18 PM - bruno katra

paul leroy wrote:

> fsw >= 3.1.0.1
> J'ai ajouté un test sur la date de transition du mode. Les CWF1, CWF2 et les BP ne sont envoyé que si la date de transition est respectée:
>
> [...]

Super ! On lancera des tests pour voir si le fonctionnement attendu est nominal. Quid des CWF_F3 (par exemple lors d'une transition BURST vers NORMAL) ?

#14 - 25/05/2016 12:52 PM - paul leroy

BURST => NORM est une transition dans laquelle tout redémarre. Les transitions particulières affectées par la nouvelle commutation sont les transition SBMx <=> NORM.

#15 - 25/05/2016 12:54 PM - bruno katra

paul leroy wrote:

> BURST => NORM est une transition dans laquelle tout redémarre. Les transitions particulières affectées par la nouvelle commutation sont les transition SBMx <=> NORM.

Exact ! Au temps pour moi.

#16 - 27/01/2017 02:36 PM - bruno katra

- Status changed from In Progress to Closed

SUM mis à jour pour R3 updated DP concernant les changements de mode
+ RFD émises pour R3 updated DP:
RFD 226 (commutation immédiate)
RFD 227 (redémarrage de certians produits seulement)