Développements VHDL et SOFT

Sommaire

[1 VHDL 2](#_Toc341679111)

[1.1 Patcher la GRLIB 2](#_Toc341679112)

[1.2 Tester le projet DIGILENT LCD 2](#_Toc341679113)

[1.3 Constitution d’un projet GRLIB 2](#_Toc341679114)

[1.4 Génération de la doc 3](#_Toc341679115)

[1.5 Projet MINIAMBA 3](#_Toc341679116)

[1.6 SpaceWire 3](#_Toc341679117)

[2 VHDL XILINX 4](#_Toc341679118)

[2.1 A partir du 24 OCT 2011 4](#_Toc341679119)

[2.1.1 Projet de base Leon3 Xilinx 4](#_Toc341679120)

[2.1.2 Projet Leon3 avec SpaceWire Xilinx 4](#_Toc341679121)

[2.1.3 Bilan des projets compilables dans la grlib-ft 5](#_Toc341679122)

[2.2 LFR-142200-DM-LEON3-BASE 5](#_Toc341679123)

[2.2.1 SSRAM\_PLUGIN.vhd 5](#_Toc341679124)

[2.3 MINI AMBA SpaceWire XILINX 6](#_Toc341679125)

[2.4 20120910\_digilent-Leon3-spacewire 8](#_Toc341679126)

[3 VHDL ACTEL 8](#_Toc341679127)

[3.1 Projet pour carte M7A3P1000 8](#_Toc341679128)

[3.1.1 20120723 9](#_Toc341679129)

[3.1.2 20120730 9](#_Toc341679130)

[3.1.3 Projet-Blanc-M7A3PK-LPP 9](#_Toc341679131)

[3.1.4 SPW-Projet-Blanc-M7A3PK-LPP 9](#_Toc341679132)

[3.1.5 20120913-SPW-Projet-Blanc-M7A3PK-LPP 10](#_Toc341679133)

[4 Développement soft 10](#_Toc341679134)

[4.1 Premiers projets 10](#_Toc341679135)

[4.1.1 spwpacketmanager.cbp 10](#_Toc341679136)

[4.1.2 time\_LPP.cbp 11](#_Toc341679137)

[4.1.3 irq\_LPP.cbp 11](#_Toc341679138)

[4.2 Gestion du temps SpaceWire 11](#_Toc341679139)

[4.2.1 Présentation 11](#_Toc341679140)

[4.2.2 M7A3P1000 11](#_Toc341679141)

[4.2.3 Carte Digilent 12](#_Toc341679142)

[4.3 Simulation de l’arrivée des matrices spectrales 12](#_Toc341679143)

[4.4 Budget d’occupation du CPU 13](#_Toc341679144)

[4.5 Composition d’une matrice spectrale 13](#_Toc341679145)

[4.5.1 Degree of polarization 13](#_Toc341679146)

[4.5.2 z-component of the normalized Poynting vector 13](#_Toc341679147)

[4.5.3 Gestion de version 13](#_Toc341679148)

[5 Xilinx VIRTEX-5 XUPV5-LX110T 14](#_Toc341679149)

# VHDL

## Patcher la GRLIB

Installer TortoiseHg, Doxygen et Graphviz.

Cloner la VHD-Lib sur le pc avec TortoiseHg

Créer la variable GRLIB qui pointe sur la GRLIB qu’on veut patcher

Aller dans le dossier VHD\_lib

make help pour avoir l’aide, permet aussi de verifier que la GRLIB est bien détectée

make Patch-GRLIB pour patcher la GRLIB

## Tester le projet DIGILENT LCD

Aller dans le répertoire

E:\GAISLER\VHD\_Lib\LPP\_drivers\exemples\APB\_lcd\_ctrlr

Puis lancer

make load PORT=COM2

E:\GAISLER\VHD\_Lib\LPP\_drivers\rules.mk contient les données de débit pour le port COM utilisé par grmon-eval (38400 fonctionne bien)

E:\GAISLER\VHD\_Lib\LPP\_drivers\exemples\APB\_lcd\_ctrlr\bin\APB\_lcd\_ctrlr.bin programme de test pour l’afficheur LCD de la carte SPARTAN-3E

## Constitution d’un projet GRLIB

|  |  |
| --- | --- |
| INPUT FILES | |
| .config |  |
| config.in |  |
| config.vhd.h |  |
| config.vhd.in |  |
| lconfig.tk |  |
| PROJECT\_DIR\  Makefile | makefile local permettant de générer le projet, contient entre autres le nom du top du projet VHDL |
| GRLIB\_DIR\boards\digilent-xc3s1600e  Makefile.inc | Renseignement additionnels pour le make scripts, notamment la cible (FPGA, package…) |
| tkconfig.h |  |
| OUTPUT FILES | |
| config.h | généré par make xconfig |
| PROJECT\_DIR\  config.vhd | options de configuration du projet leon3, peut être édité à la main où avec un utilitaire graphique  généré par make xconfig |
| PROJECT\_DIR\  leon3mp.vhd | top du projet, lié au fichier de config config.vhd |

make scripts marche mal avec cygwin, l’utiliser de préférence avec MinGW. Par contre, cygwin apporte les fonctions tcltk, notamment wish84.exe dont a besoin MinGW pour la configuration graphique.

Renommage de la fonction wish.exe de cygwin pour éviter les interférences avec wish de la grlib qui fait appel à wish84.exe

## Génération de la doc

E:\GAISLER\VHD\_Lib\lib\lpp

Doxygen Doxyfile

Le fichier contenant les répertoires à scanner pour doxygen est

E:\GAISLER\VHD\_Lib\lib\lpp\dirs.txt

## Projet MINIAMBA

Tests avec grmon:

grmon-eval –uart COM2 –baud 38400

mem 0x80000200

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | hindex | haddr | pindex | paddr | pmask | pirq |
| ahbuart | 1 (master) |  | 4 | 7 |  |  |
| apbctrl | 1 (slave) | 16#800# |  |  |  |  |
| apbuart |  |  | 1 | 1 |  | 2 |
| apb\_rotarypushbuttonswitch |  |  | 2 | 2 | 16#fff# | 3 |

## SpaceWire

Dans la doc GRESB, on lit “The spacewire signal conform to the 2.5V LVDS signalling scheme”.

# VHDL XILINX

## A partir du 24 OCT 2011

### Projet de base Leon3 Xilinx

design: E:\GAISLER\grlib-gpl-1.1.0-b4108\designs\PLE\_leon3-digilent-xc3s1600e

Compilation correcte avec Leon3

grmon-eval –u –uart COM1 –baud 38400

Exécution de HELLO dans E:\LFR\Eclipse\leon3\HELLO\Release

### Projet Leon3 avec SpaceWire Xilinx

copie du projet

grlib-gpl-1.1.0-b4108\designs\leon3-digilent-xc3s1600e

sous

grlib-ft-fpga-1.0.21-b4003\designs\PLE\_leon3-digilent-xc3s1600e

Copie du Makefile dans le fichier Makefile.old

Modification du Makefile

* retrait de spw de LIBSKIP
* retrait de spacewire de DIRSKIP

Lancement de make scripts OK

Synthetize => problème avec le fichier ahbrom.vhd qui fait appel à une fonction qui n’est pas implémentée dans le amba.vhd de la grlib ft (hbdrivedata)

Dans l’exemple d’un design pour la ft (Actel ProAsic), la fonction précédente n’est pas utilisée.

|  |  |  |  |
| --- | --- | --- | --- |
| led(7) | errorn |  | errorn |
| led(6) |  |  | gpio.dout(2) |
| led(5) | <= lock ; |  | <= lock ; |
| led(4) | <= dsuo.active ; |  | <= dsuo.active ; |
| led(3) | <= not duo.txd ; |  | <= not duo.txd ; |
| led(2) | <= not rxd2 ; |  | <= not rxd2 ; |
| led(1) | <= not txd1 ; |  | gpio.dout(1) |
| led(0) | <= not rxd1 ; |  | gpio.dout(0) |

Modification du fichier de contrainte du projet, la led(0) n’est pas câblée sur la bonne pin, ne pas tenir compte de la sérigraphie. led(0) => D4, avec le IOSTANDARD défini dans la doc de la carte Digilent.

Suppression du projet de tous les modules inutiles, Ethernet, SVGA, PS2. La fréquence de fonctionnement MAX passe à 40 MHz.

Enregistrement du projet vierge avec Leon3, compilable dans la grlib ft. Liste des fichiers modifiés :

* ahbrom.vhd (suppression de la fonction hbdrivedata non implémentée dans la grlib ft)
* Makefile (ajout des bibliothèques SpaceWire)
* leon3mp.vhd (suppression des modules inutiles, ethernet, svga, ps2)
* leon3mp.ucf (mise à jour de la netlist et correction de la pin de la led(6))
* leon3mp.xcf (suppression des contraintes liées à Ethernet)

### Bilan des projets compilables dans la grlib-ft

PLE\_GRLIB-FT\_leon3-digilent-xc3s1600e => projet compilé simple avec un leon3 et des GPIO câblée sur les LED 0, 1 et 6.

PLE\_ leon3-digilent-xc3s1600e => projet vierge (make scripts non exécuté) identique au précédent.

## LFR-142200-DM-LEON3-BASE

Utilisation du nouveau fichier makefile de Gaisler (copié dans bin)

Suppression des fonctions ahbdrivedata dans le fichier ahbrom.vhd

Suppression des variable SYNFILES et SIMTOP dans le fichier Makefile

L’ajout du fichier ssram\_plugin.vhd dans la liste de la variable VHDLSYNFILES permet à Synplify de détecter l’entité ssram\_plugin.

Le numéro ahbslave du contrôleur de mémoire doit être supérieur ou égal à 1 d’après la doc GRIP.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | ID |
| 00.01 :003 | LEON3 | 0 |  | AHB master |  |
| 01.01 :007 | AHB Debug UART | 1 | 7 | AHB master / APB slave | 01 :007 |
| 00.04 :00f | LEON2 Memory Controller | 0 | 0 | AHB slave / APB slave | 04 :00f |
| 01.01 :006 | AHB/APB Bridge | 1 |  | AHB slave | 01 :006 |
| 02.01 :004 | LEON3 DSU | 2 |  | AHB slave |  |
| 01.01 :00c | APB UART |  | 1 | APB slave |  |
| 02.01 :00d | Interrupt Ctrl |  | 2 | APB slave |  |
| 03.01 :011 | Modular Timer |  | 3 | APB slave |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | ID |
| 00.01 :003 | LEON3 | M 0 |  | AHB master |  |
| 01.01 :007 | AHB Debug UART | M 1 | 4 | AHB master / APB slave | 01 :007 |
| 00.04 :00f | LEON2 Memory Controller | S 0 | 0 | AHB slave / APB slave | 04 :00f |
| 01.01 :006 | AHB/APB Bridge | S 1 |  | AHB slave | 01 :006 |
| 02.01 :004 | LEON3 DSU | S 2 |  | AHB slave |  |
| 03.01 :00e | AHB RAM | S 3 |  | AHB slave | 01 :00e |
| 04.01 :00e | AHB RAM | S 4 |  | AHB slave | 01 :00e |
| 01.01 :00c | APB UART |  | 1 | APB slave |  |
| 02.01 :00d | Interrupt Ctrl |  | 2 | APB slave |  |
| 03.01 :011 | Modular Timer |  | 3 | APB slave |  |

### SSRAM\_PLUGIN.vhd

|  |  |  |
| --- | --- | --- |
| SSRAM\_CLK | clk |  |
| nBWa | nBWaint | mem\_ctrlr\_o.WRN(3) **or** mem\_ctrlr\_o.ramsn(0) |
| nBWb | nBWbint | mem\_ctrlr\_o.WRN(2) **or** mem\_ctrlr\_o.ramsn(0) |
| nBWc | nBWcint | mem\_ctrlr\_o.WRN(1) **or** mem\_ctrlr\_o.ramsn(0) |
| nBWd | nBWdint | mem\_ctrlr\_o.WRN(0) **or** mem\_ctrlr\_o.ramsn(0) |
| nBWE | nBWEint | mem\_ctrlr\_o.WRITEN **or** mem\_ctrlr\_o.ramsn(0) |
| nADSC | ‘1’ |  |
| nADSP | nADSPint | **not**( RAMSN\_reg **xor** mem\_ctrl\_o.RAMSN(0) ) **and** RAMSN\_reg ) |
|  |  | process(clk)  RAMSN\_reg <= mem\_ctrlr\_o.RAMSN(0) |
|  |  | Permet de réduire la durée du signal nADPS à un cycle |
| nADSP | nADSPint |  |
| nADV | ‘1’ |  |
| nGW | ‘1’ |  |
| nCE1 | nADSPint |  |
| CE2 | ‘1’ |  |
| nCE3 | ‘0’ |  |
| nOE | nOEint | nOEint <= OEreg **or** mem\_ctrlr\_o.RAMOEN(0) |
|  |  | process(clk)  OEreg <= mem\_ctrlr\_o.OEN |
|  |  | Permet de décaler le signal OEN **mais** réduit sa durée à 1 cycle |
| MODE | ‘0’ |  |
| ZZ | ‘0’ |  |

## MINI AMBA SpaceWire XILINX

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pin** | **Name** | **Description** | **xc3s16600e**  **FG320** | **FX2 connector** | **MINIAMBA**  **Designation** |
| **1** | DIN+ | Data In +ve | A4 | IO2 | dop |
| **6** | DIN- | Data In –ve | B4 | IO1 | don |
| **2** | SIN+ | Strobe In +ve | C5 | IO4 | sop |
| **7** | SIN- | Strobe In –ve | D5 | IO3 | son |
| **3** | SHIELD | Inner Shield (connects to DGND) |  |  |  |
| **8** | SOUT+ | Strobe Out +ve | F7 | IO8 | sip |
| **4** | SOUT- | Strobe Out –ve | E7 | IO7 | sin |
| **9** | DOUT+ | Data Out +ve | B6 | IO6 | dip |
| **5** | DOUT- | Data Out -ve | A6 | IO5 | din |

30 JAN 2012

Copie du projet MINIAMBA-spacewire dans 20120130\_MINIAMBA\_spacewire pour essayer d’implémenter le module spacewire comme ce qui est fait dans le design designs/leon3-gr-xc3s-1500/leon3mp.vhd (recommandation suite à mail Gaisler).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | ID |
| 01.01 :007 | AHB Debug UART | M 0 | 0 | AHB master / APB slave | 01 :007 |
| 02.01 :01f | GRSPW SpaceWire | M1 | 1 | AHB master / APB slave | 01 :01f |
| 01.01 :006 | AHB/APB Bridge | S 0 | em | AHB slave | 01 :006 |
| 01.01 :00c | APB UART |  | 2 | APB slave |  |

02 FEB 2012

Projet compilé et fonctionnel

**20120130\_MINIAMBA-spacewire**

Permet la connexion de grmon via le module gresb au design.

Copie du projet vierge PLE\_leon3-digilent-xc3s1600e (projet gaisler modifié, présenté plus haut). Objectif => implanter le module spacewire avec le leon3. Projet créé

**20120202\_digilent-Leon3-spacewire**

12 :40 lancement d’une compilation du projet sans modification

13 :20 compilation réussie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | IRQ |
| 00.01 :003 | Leon3 | M 0 |  | AHB master |  |
| 01.01:01c | GRSPW SpaceWire | M1 | 5 | AHB Master | 10 |
| 02.01:0007 | AHB Debug UART | M2 | 4 | AHB Master |  |
| 01.01 :006 | AHB/APB Bridge | S1 |  | AHB Slave |  |
| 02.01 :004 | Leon3 Debug Support Unit | S2 |  | AHB Slave |  |
| 04.01 :025 | DDR266 Controller | S4 |  | AHB Slave |  |
| 05.04 :00f | ESA LEON2 memory controller | S5 | 0 | AHB Slave |  |
| 01.01 :00c | APB UART |  | 1 |  | 2 |
| 02.01 :00d | Interrupt Controller |  | 2 |  |  |
| 03.01 :011 | Timer |  | 3 |  | 8 |
| 0b :01 :01a | General Purpose I/O port |  | b |  |  |

Compilation réussie avec un Leon3 et le module SpaceWire => leon3mp\_20120206.bit

Compilation réussie avec un Leon3 et le module SpaceWire (numérotation différente des masters) => leon3mp\_20120206\_v2.bit

28 JUIN 2012

Reprise du projet précédent, renommé 20120628\_digilent-Leon3-spacewire

Objectif = attribuer l’IRQ 10 au module GRSPW (ancienne IRQ 0 interdite d’après les registres du contrôleur d’interruption).

29 JUIN 2012

Compilation réussie (leon3mp\_20120628.bit)

Modification des options CFG\_MAC (0=>1) et CFG\_FPU (0 + 16\*0=>1 + 16\*0)

La compilation échoue, il faut utiliser la netlist grlfpw.ngo pour implémenter la grfpu lite

CFG\_FPU (0 + 16\*0=>1 + 16\*1)

Compilation russie

02 JUILLET 2012

Conception d’un module **lfr\_time\_management** contenant un compteur **fine\_time\_counter** comptant de 0 à 2^16-1 (2^16 Hz)

* le compteur est remis à zéro à réception du signal tick-out émis par le module GRSPW après chaque réception d’un time-code valide
* un registre **coarse\_time\_r** contenant 32 bits est connecté au bus APB
* un registre **coarse\_time\_load\_value\_r** contenant 32 bits est connecté au bus APB
* la valeur du registre **coarse\_time\_load\_value\_r** est mise à jour par le Leon3 lors de la réception d’une télécommande appropriée.
* la valeur du registre **coarse\_time\_r** est mise à jour lors de la réception du signal tick-out (**grspw\_tick**) avec la valeur courante du registre **coarse\_time\_load\_value\_r**
* le registre **fine\_time\_r** contenant 32 bits est connecté au bus APB, il contient la valeur courante du compteur

|  |  |
| --- | --- |
| **APB address offset** | **Register** |
| 0x00 | coarse time load value |
| 0x04 | coarse time current value |
| 0x08 | fine time value |

03 JUILLET

Compilation réussie du projet AMBA\_rotarypushbuttonswitch-digilent-xc3s1660e

Contient une instanciation de apb\_lfr\_time\_management fonctionnelle

Essai de l’intégration de apb\_lfr\_time\_management dans un projet avec grspw et leon3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | IRQ |
| 00.01 :003 | Leon3 | M 0 |  | AHB master |  |
| 01.01:01c | GRSPW SpaceWire | M1 | 5 | AHB Master | 10 |
| 02.01:0007 | AHB Debug UART | M2 | 4 | AHB Master |  |
| 01.01 :006 | AHB/APB Bridge | S1 |  | AHB Slave |  |
| 02.01 :004 | Leon3 Debug Support Unit | S2 |  | AHB Slave |  |
| 04.01 :025 | DDR266 Controller | S4 |  | AHB Slave |  |
| 05.04 :00f | ESA LEON2 memory controller | S5 | 0 | AHB Slave |  |
| 01.01 :00c | APB UART |  | 1 |  | 2 |
| 02.01 :00d | Interrupt Controller |  | 2 |  |  |
| 03.01 :011 | Timer |  | 3 |  | 8 |
| 0b :01 :01a | General Purpose I/O port |  | b |  |  |
|  | apb\_lfr\_time\_management |  | 6 |  |  |

Projet compilé et testé (20120703\_digilent-Leon3-spacewire). Le module apb\_lfr\_time\_management fonctionne correctement. L’entrée grspw\_tick est câblée sur le signal swno.tickout du module grspw

## 20120910\_digilent-Leon3-spacewire

10 sept.-12

Modification du module lfr\_time\_management pour inclure une fonction de forçage logiciel de la remise à 0 du compteur fine time.

|  |  |
| --- | --- |
| **register** | **offset** |
| ctrl | 0x00 |
| coarse\_time\_load | 0x04 |
| coarse\_time | 0x08 |
| fine\_time | 0x0c |

Compilation réussie, fichier de programmation stocké en PROM.

## Configuration port COM carte Digilent

Câblage sur le SubD mâle vers l’éclateur SubD, utilisation d’un câble gris série

* 2 => 3
* 3 => 2
* 5 => 5

# VHDL ACTEL

## Projet pour carte M7A3P1000

Reprise du projet livré dans la grlib-ft-fpga-1.0.22-b4098, leon3-actel-proasic3, renommé 20120705 leon3-actel-proasic3.

Modification du numéto de master pour grspw2 => 2 au lieu de 1 (incompatibilité avec master 0 Leon3 et master 1 ahbuart).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | IRQ |
| 00.01 :003 | Leon3 | M 0 |  | AHB master |  |
| 01.01:01c | GRSPW SpaceWire | M1 | 5 | AHB Master | 10 |
| 02.01:0007 | AHB Debug UART | M2 | 7 | AHB Master |  |
| 01.01 :006 | AHB/APB Bridge | S1 |  | AHB Slave |  |
| 02.01 :004 | Leon3 Debug Support Unit | S2 |  | AHB Slave |  |
| 04.01 :025 | DDR266 Controller | S4 |  | AHB Slave |  |
| 05.04 :00f | ESA LEON2 memory controller | S5 | 0 | AHB Slave |  |
| 01.01 :00c | APB UART |  | 1 |  | 2 |
| 02.01 :00d | Interrupt Controller |  | 2 |  |  |
| 03.01 :011 | Timer |  | 3 |  | 8 |
| 0b :01 :01a | General Purpose I/O port |  | b |  |  |
|  | apb\_lfr\_time\_management |  | 6 |  |  |

### 20120723

Modification du paramètre PADTECH, inferred => apa3 dans le fichier config.h

Problème lors de la lecture du fichier pdc, erreur sur les bit ramben(0 :3), address(0 :18), data(0 :1)

port name doesn’t exist in the netlist or is not connected to an IoCell macro at PDC Line : set\_io address(0) –pinname F12 –fixed yes

### 20120730

Reprise du projet

Suppression de lpp dans le Makefile (LIBSKIP, DIRSKIP)

### Projet-Blanc-M7A3PK-LPP

Essai d’obtention d’une grlib fonctionnelle :

1. téléchargement grlib ft 1.1.0-b4113
2. copie du repertoire lib/tech/proasic3 à partir de la grlib-ft
3. copie du fichier lib/techmap/proasic3 à partir de la grlib-ft
4. copie du fichier projet blanc dans design
5. copie du fichier dans le répertoire boards
6. modification du fichier leon3mp.vhd pour supprimer la référence à la bibliothèque lpp puisque la grlib n’est pas encore patchée.
7. Compilation complète du projet sans problème

### SPW-Projet-Blanc-M7A3PK-LPP

1. copie du répertoire netlists (contient les netlists xilinx, inutile pour le projet Actel)
2. les netlists pour le SpaceWire sont dans le répertoire déjà copié lib/techmap/proasic3

Problème pendant la synthèse avec le signal timerrstval, qui ne devrait pas être utilisé par grspw2. Valeur fixée à 0 avec l’instruction : (others => ‘0’)

1. Ajout du fichier grspw2\_phy.vhd dans le répertoire gaisler/spacewire (requis pour le design avec SpaceWire)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | IRQ |
| 00.01 :003 | Leon3 | M 0 |  | AHB master |  |
| 01.01:007 | AHB Debug UART | M1 | 4 | AHB Master |  |
| 02.01:029 | GRSPW | M2 | 5 | AHB Master | 10 |
|  | | | | | |
| 00.04 :00f | ESA LEON2 memory controller | S0 | 0 | AHB Slave |  |
| 01.01 :006 | AHB/APB Bridge | S1 |  | AHB Slave |  |
| 02.01 :004 | Leon3 Debug Support Unit | S2 |  | AHB Slave |  |
|  | | | | | |
| 01.01 :00c | APB UART |  | 1 |  | 2 |
| 02.01 :00d | Interrupt Controller |  | 2 |  |  |
| 03.01 :011 | Timer |  | 3 |  | 8 |
| 0b :01 :01a | General Purpose I/O port |  | b |  |  |

1. Le fichier grspwc2\_net n’est pas à jour dans la grlib-gpl. Copie du fichier correspondant de la grlib-ft (lib/techmap/maps/grspwc2\_net.vhd)
2. Ne pas déclarer les pins des buffers différentiels dans le fichier pdc du répertoire boards. Libero requière de les configurer à la main dans Designer
3. Les résistances des buffers LVDS ne sont pas intégrées dans le FPGA. Les résistances sont ajoutées directement dans le connecteur spécifique

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Côté Actel** |  | **Côté GRESB** |
| G2 | dop | 1 | Din+ |
| G5 | sop | 2 | Sin+ |
|  |  | 3 | Inner shield |
| F3 | sin | 4 | Sout- |
| F4 | din | 5 | Dout- |
| G1 | don | 6 | Din- |
| G4 | son | 7 | Sin- |
| F2 | sip | 8 | Sout+ |
| E4 | dip | 9 | Dout+ |

1. modification de l’adresse SpaceWire par défaut de la carte M7A3P1000 (modification impossible dans l’IP GRSPW2, il faut le faire à la main à partir du Leon3 par exemple)

### 20120913-SPW-Projet-Blanc-M7A3PK-LPP

Modification du projet pour avoir une configuration proche du projet Xilinx, pour lequel les programmes d’exemple de rtems fonctionnent. Modification de l’ordre des masters ahb

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | IRQ |
| 00.01 :003 | Leon3 | M 0 |  | AHB master |  |
| 02.01:029 | GRSPW | M1 | 5 | AHB Master | 10 |
| 01.01:007 | AHB Debug UART | M2 | 4 | AHB Master |  |
|  | | | | | |
| 00.04 :00f | ESA LEON2 memory controller | S0 | 0 | AHB Slave |  |
| 01.01 :006 | AHB/APB Bridge | S1 |  | AHB Slave |  |
| 02.01 :004 | Leon3 Debug Support Unit | S2 |  | AHB Slave |  |
|  | | | | | |
| 01.01 :00c | APB UART |  | 1 |  | 2 |
| 02.01 :00d | Interrupt Controller |  | 2 |  |  |
| 03.01 :011 | Timer |  | 3 |  | 8 |
| 0b :01 :01a | General Purpose I/O port |  | b |  |  |

**Il faut configurer le nombre de register windows du Leon3 à 8 pour être compatible avec RCC et BCC (cf doc Leon3). => paramètre nwin**

# Développement soft

## Premiers projets

### spwpacketmanager.cbp

Projet permettant l’émission d’un packet vers grmon toutes les secondes. Le plugin rmap est capable de recevoir le packet et de le lire.

La période est déterminée par un timer.

L’envoie d’un packet vers la cible déclenche une interruption sur la ligne 10, le packet est lu et un message est envoyé sur la sortie standard pour lecture dans GRMON.

|  |  |
| --- | --- |
| spw\_send\_ccsds.c |  |
| spwAPI\_LPP.h |  |
| spwAPI\_LPP.c |  |

### time\_LPP.cbp

Projet permettant le test d’un timer dans le leon3. Affichage de messages périodique.

|  |  |
| --- | --- |
| time\_LPP.c |  |

### irq\_LPP.cbp

Projet permettant de tester les interruptions en forçant l’émission d’une interruption (irq 10).

|  |  |
| --- | --- |
| main.c |  |

## Gestion du temps SpaceWire

### Présentation

#### Objectifs

Le projet a pour objectif la gestion du temps SpaceWire telle que décrite dans la spécification LFR (SRS). La gestion se fait en deux temps i) envoi d’un paquet CCSDS contenant le temps système, ii) mise à jour du temps système à la réception du timecode SpaceWire suivant la réception du paquet CCSDS.

#### Développement STEP 1

1. envoi par system\_time\_generator des paquets CCSDS contenant le temps système toutes les secondes
2. réception par la cible des paquets et envoi vers LPPMON d’un accusé réception contenant le temps reçu ainsi que l’adresse d’origine
3. écriture du temps système dans le registre 0x00 de l’IP lpp\_time\_manager

### M7A3P1000

#### Objectifs

Le Leon3 instancié dans la carte M7A3P1000 a pour objectifs :

1. envoyer un timecode SpaceWire toutes les secondes
2. envoyer un paquet contenant la valeur du temps système 300ms avant chaque timecode SpaceWire (le temps vaut 0 au démarrage puis est incrémenté de 1 toutes les secondes)
3. démarrage sur la ROM et émission continue des paquets et des timecodes dès le reset

#### Déroulement du programme

1. Initialisation :
   1. modification de l’adresse par défaut 0xfe => 0xfd (modifier la table de routage de la gresb en conséquence : Address 253, Enable yes, Link type SPW, Link ID 1)
   2. initialisation SpaceWire
2. Mise en place d’un timer pour l’envoi des paquets contenant le temps système
3. Mise en place d’un timer pour l’envoi des timecodes SpaceWire

#### Projet Code ::Blocks

system\_time\_generator

#### Notes

Les registres de l’IP grspw sont situés à l’adresse GRSPW\_APB\_ADDR qui est stockées en dur dans le fichier spwAPI\_LPP.h

### Carte Digilent

#### Objectifs

Le Leon3 instancié dans la carte Digilent a pour objectifs :

1. réception des paquets CCSDS contenant le temps système
2. copie de la valeur du temps système dans le registre approprié
3. mise à jour du temps système lors de la réception du timecode SpaceWire qui suit le paquet CCSDS (signal émis par le module grspw2, swno.tickout)
4. envoi du temps système vers le GSE

#### Déroulement du programme

1. initialisation :
   1. mise en place du gestionnaire d’interruption pour la réception des paquets SpaceWire
2. boucle infinie en attente des interruptions

## Simulation de l’arrivée des matrices spectrales

Création d’un module VHDL avec des registres permettant de stocker les adresses allouées par le Leon3 pour le stockage des matrices spectrales (deux buffers de 3840 mots permettant de stocker deux matrices spectrales complètes).

Le module contient un registre ctrl permettant de signaler au Leon3 qu’une matrice est prête à être traitée.

|  |  |
| --- | --- |
| **register** | **offset** |
| ctrl | 0x00 |
| spectral\_matrix\_1 | 0x04 |
| spectral\_matrix\_2 | 0x08 |

Déroulement du programme :

1. Leon3 alloue deux espaces pour le stockage des matrices spectrales
2. Leon3 écrit l’adresse des buffers de stockage dans les registres spectral\_matrix\_1 et spectral\_matrix\_2
3. LPPMON lit les adresses des buffers de stockage
4. LPPMON écrit alternativement dans chaque buffer, toutes les 62.5 ms (256 échantillons \* 1/4096s)
5. LPPMON met à jour le buffer ctrl pour signaler la disponibilité des matrices spectrales
6. Leon3 lit les matrices spectrales et fait les moyennes
7. Une fois la moyenne effectuée, il envoi la matrice à LPPMON
8. LPPMON reçoit le paquet et le traite

20120920 => projet intégrant le module lfr\_spectral\_matrices\_DMA.

## Budget d’occupation du CPU

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | % |  |
| Nombre total de counts | 65535 | 100.00 % | 1 000 ms |
| Moyenne d’une matrice spectrale | 170 |  |  |
|  |  |  |  |
|  |  |  |  |

## Composition d’une matrice spectrale

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S11**  0 1 | **S12**  2 3 | **S13**  4 5 | **S14**  6 7 | **S15**  8 9 |  | **S11**  0 1 | **S12**  2 3 | **S13**  4 5 | **S14**  6 7 | **S15**  8 9 |
| S12 | **S22**  10 11 | **S23**  12 13 | **S24**  14 15 | **S25**  16 17 |  | S12 | **S22**  10 11 | **S23**  12 13 | **S24**  14 15 | **S25**  16 17 |
| S13 | S23 | **S33**  18 19 | **S34**  20 21 | **S35**  22 23 |  | S13 | S23 | **S33**  18 19 | **S34**  20 21 | **S35**  22 23 |
| S14 | S24 | S34 | **S44**  24 25 | **S45**  26 27 |  | S14 | S24 | S34 | **S44**  24 25 | **S45**  26 27 |
| S15 | S25 | S35 | S45 | **S55**  28 29 |  | S15 | S25 | S35 | S45 | **S55**  28 29 |

### Degree of polarization



a = S11 \* S11 + S12 \* S12 + S13 \* S13

b = S12 \* S12 + S22 \* S22 + S23 \* S23

c = S13 \* S13 + S23 \* S23 + S33 \* S33

Tr[ SB x SB] = a + b + c = S112 + S222 + S332 + 2 \* (S122 + S232 + S132)

### z-component of the normalized Poynting vector



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ex |  | Bx |  | Ey.Bz – By.Ez |
| Ey |  | By |  | Ex.Bz – Bx.Ez |
| Ez |  | Bz |  | Ex.By - Bx.Ey |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ex |  | Bx\* |  |  |
| Ey |  | By\* |  |  |
|  |  |  |  | Ex.By\* - Bx\*.Ey  E1.B2\* – B1\*.E2  S24\* – S15\* |

### Gestion de version

make distclean dans le /opt/PAULs\_LPPMON\_PLUGINS

Installation de tortoise (interface graphique de mercurial)

démarrer TortoiseHg

ouvrir le dépôt

ligne de commande

mkdir tmp

hg clone ssh://leroy@pc-instru://22//HDDS/VERSION/Repositories/LPPMON\_PLUGINs\_PAUL

smb://pc-instru/version/LPPMON\_PLUGINs\_PAUL

# Xilinx VIRTEX-5 XUPV5-LX110T

## Design grlib-gpl

de **grlib-gpl-1.1.0-b4113** vers **grlib-ft-fpga-1.0.22-b4098**

Nouveau projet **2012-11-22-leon3-xilinx-ml509**

Le répertoire boards contient déjà le répertoire **xilinx-ml-509-xc5vlx110t**

**make scripts** dans le répertoire du design avec MINGW

Migration du projet dans une version plus récente de ise

Le projet ne peut pas être compilé avec cette version de ise

Installation ISE 14.3, le projet ne compile pas tel quel dans la grlib-ft

Le projet ne compile pas dans la grlib-gpl-1.1.0-b4113

Le projet commence à compiler après modification du Makefile : suppression d’un espace avant l’option « -uc » de la variable XSTOPT

Suppression des appels à ahbdrivedata dans le fichier ahbrom.vhd (nécessire en simulation ?)

LIBSKIP i2c opencores

DIRSKIP greth opencores

Si on ne met opencores que dans LIBSKIP, la compilation echoue dans la grlib-ft

on enlève à la main

* gaisler\misc\i2cmst.vhd
* gaisler\misc\i2cmst\_gen.vhd
* gaisler\misc\i2cslv.vhd

Suppression des signaux liés à l’ethernet sinon des erreurs apparaissent à cause de l’absence de la Library correspondante.

Modification leon3mp.xcf

* suppression de la ligne phy\_tx\_data
* suppression de la ligne egtx\_clk

The serial port is connected to the console UART (UART 1) when dip switch 1 on the GPIO DIP switch is off. Otherwise it is connected to the DSU UART

Tableau 1SubD9 mâle sur la carte XUPV5-LX110T

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** |  | **2 Rx**  **AG 15**  **rxd1 in** |  | **3 Tx**  **AG 20**  **txd1 out** |  | **4** |  | **5 GND** |
|  | **6** |  | **7** |  | **8** |  | **9** |  |

Vérification à l’oscilloscope: Tx PC = pin 3 du connecteur Mâle PC => il faut croiser

Essai compilation sans l’AHBJTAG

Modification de la génération du signal lock modifiée par la suppression des horloges Ethernet et VGA

Compilation réussie, la connexion avec GRMON fonctionne.

## Design grlib-gpl avec SpaceWire

2012-11-26-leon3-xilinx-ml509

Modification du Makefile pour y ajouter le SpaceWire (variables DIRSKIP et LIBSKIP).

Tableau Configuration du design avant implantation du SpaceWire

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | IRQ |
| 00.01 :003 | Leon3 | M 0 |  | AHB master |  |
| 01.01 :007 | AHB Debug UART | M1 | 7 | AHB Master |  |
| 02.01:063 | SVGA controller | M2 | 6 | AHB Master |  |
|  | | | | | |
| 00.01 :02e | DDR2 controller | S0 |  |  |  |
| 01.01 :006 | AHB/APB Bridge | S1 |  | AHB Slave |  |
| 02.01 :004 | Leon3 Debug Support Unit | S2 |  | AHB Slave |  |
| 03.04 :00f | ESA LEON2 memory controller | S3 | 0 | AHB Slave |  |
| 04.01 :067 | System ACE I/F controller | S4 |  | AHB Slave |  |
|  | | | | | |
| 01.01 :00c | APB UART |  | 1 |  | 2 |
| 02.01 :00d | Interrupt Controller |  | 2 |  |  |
| 03.01 :011 | Timer |  | 3 |  | 8 |
| 04.01 :060 | PS/2 interface |  | 4 |  | 4 |
| 04.01 :060 | PS/2 interface |  | 5 |  | 5 |
| 0b :01 :01a | General Purpose I/O port |  | 8 |  |  |
| 0f.01 :052 | AHB status register |  | f |  | 7 |

Tableau Configuration du design après implantation du SpaceWire

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | IRQ |
| 00.01 :003 | Leon3 | M 0 |  | AHB master |  |
| **01.01:01c** | **GRSPW SpaceWire** | **M1** | **9** | **AHB Master** | **10** |
| **02**.01 :007 | AHB Debug UART | M2 | 7 | AHB Master |  |
| **03**.01:063 | SVGA controller | M3 | 6 | AHB Master |  |
|  | | | | | |
| 00.01 :02e | DDR2 controller | S0 |  |  |  |
| 01.01 :006 | AHB/APB Bridge | S1 |  | AHB Slave |  |
| 02.01 :004 | Leon3 Debug Support Unit | S2 |  | AHB Slave |  |
| 03.04 :00f | ESA LEON2 memory controller | S3 | 0 | AHB Slave |  |
| 04.01 :067 | System ACE I/F controller | S4 |  | AHB Slave |  |
|  | | | | | |
| 01.01 :00c | APB UART |  | 1 |  | 2 |
| 02.01 :00d | Interrupt Controller |  | 2 |  |  |
| 03.01 :011 | Timer |  | 3 |  | 8 |
| 04.01 :060 | PS/2 interface |  | 4 |  | 4 |
| 04.01 :060 | PS/2 interface |  | 5 |  | 5 |
| 0b :01 :01a | General Purpose I/O port |  | 8 |  |  |
| 0f.01 :052 | AHB status register |  | f |  | 7 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Header** | **FPGA** | **Côté AXilinx** |  | **Côté GRESB** |
| 4 | L34 | dop | 1 | Din+ |
| 8 | K33 | sop | 2 | Sin+ |
|  |  |  | 3 | Inner shield |
| 14 | R34 | sin | 4 | Sout- |
| 10 | N32 | din | 5 | Dout- |
| 2 | K34 | don | 6 | Din- |
| 6 | K32 | son | 7 | Sin- |
| 16 | T33 | sip | 8 | Sout+ |
| 12 | P32 | dip | 9 | Dout+ |

Compilation réussie, le module SpaceWire est bien instancié. Réglage de la valeur de la fréquence.

2013-01-10-leon3-xilinx-ml509

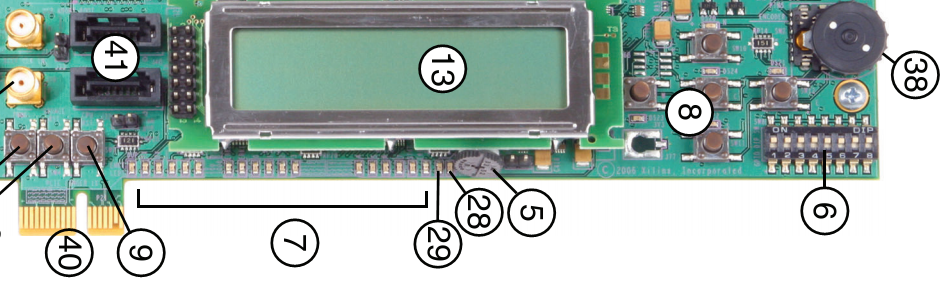
* modification de la position de grspw2 sur le bus apb pour avoir une compatibilité entre la carte spartan-3E et la carte virtex 5
* modification du câblage du port série pour inverser Rx et Tx et éviter l’utilisation de la carte éclatée

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ahb | apb |  | IRQ |
| 00.01 :003 | Leon3 | M 0 |  | AHB master |  |
| **01.01:01c** | **GRSPW SpaceWire** | **M1** | **~~9~~ 5** | **AHB Master** | **10** |
| **02**.01 :007 | AHB Debug UART | M2 | 7 | AHB Master |  |
| **03**.01:063 | SVGA controller | M3 | 6 | AHB Master |  |
|  | | | | | |
| 00.01 :02e | DDR2 controller | S0 |  |  |  |
| 01.01 :006 | AHB/APB Bridge | S1 |  | AHB Slave |  |
| 02.01 :004 | Leon3 Debug Support Unit | S2 |  | AHB Slave |  |
| 03.04 :00f | ESA LEON2 memory controller | S3 | 0 | AHB Slave |  |
| 04.01 :067 | System ACE I/F controller | S4 |  | AHB Slave |  |
|  | | | | | |
| 01.01 :00c | APB UART |  | 1 |  | 2 |
| 02.01 :00d | Interrupt Controller |  | 2 |  |  |
| 03.01 :011 | Timer |  | 3 |  | 8 |
| 04.01 :060 | PS/2 interface |  | 4 |  | 4 |
| 04.01 :060 | PS/2 interface |  | ~~5~~ 9 |  | 5 |
| 0b :01 :01a | General Purpose I/O port |  | 8 |  |  |
| 0f.01 :052 | AHB status register |  | f |  | 7 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** |  | **2 Rx**  **AG 15**  **txd1** |  | **3 Tx**  **AG 20**  **rxd1** |  | **4** |  | **5 GND** |
|  | **6** |  | **7** |  | **8** |  | **9** |  |

Le changement d’attribution des pins pour la liaison série ne fonctionne pas, rxd1 est déclarée comme input et rxd1 comme output

|  |  |  |
| --- | --- | --- |
| led(0) | gpio.val(0) |  |
| led(1) | not txd1 | txd1 comes from the pc |
| led(2) | not duo.txd when gpio.val(0) = ‘1’ else not u1o.txd |  |
| led(3) |  |  |
| led(4) | dsuo.active |  |
| led(5) | lock |  |
| led(6) | cgo.clklock |  |
| led(7) | not(cgo.clklock) |  |
| error 1 | bus\_error(0) <= notdbgo(0).error | red, F6 |
| error2 | bus\_error(1) ? | red, T10 |



Modification des ports connectés au port série pour revenir à la configuration initiale. Besoin d’un connecteur croisant 2 et 3 pour éviter d’utiliser la carte éclatée.