

組み込みソフトウェア実践プロジェクト演習講座

本演習は Xilinx ISE Design Suite 12.3 (Embedded Edition) を対象としています。

Lab1: BSB を使った PowerPC システムの構築

《この Lab で習得する事》

BSB (BaseSystemBuilder) を使ったツール基本操作と FPGA へのダウンロード手順を習得します。

《実習ボード設定》

実習ボードのスイッチは以下のように設定してください。

✓ DIP スイッチ

Bit1-3 : Don' t Care

Bit4 : ON Bit5 : OFF Bit6 : ON

✓ SW12

Don' t Care

1) Lab フォルダに移動し、サブフォルダ「lab1」を新規作成してください。

2) Xilinx Platform Studio を起動し、下記要件を満たす PowerPC システムを BSB (BaseSystemBuilder) で作成してください。

- system.xmp は lab1 フォルダ上に作成してください。
- interconnect type : PLB system
- new design を作成します。
- Board Vendor : Xilinx
- Board Name : Virtex4 ML403 Evaluation Platform
- Board revision : 1
- System Configuration : Single-Processor System
- Reference clock : 100MHz
- Processor Type : PowerPC
- Processor clock : 100MHz
- Bus clock : 50MHz
- On Chip Memory : None

- 使用する Peripheral は以下です
 - ✓ LEDs_4Bit (**Interrupt 使用**) ←チェックを忘れないで下さい!
 - ✓ LEDs_Positions
 - ✓ Push_Buttons_Position

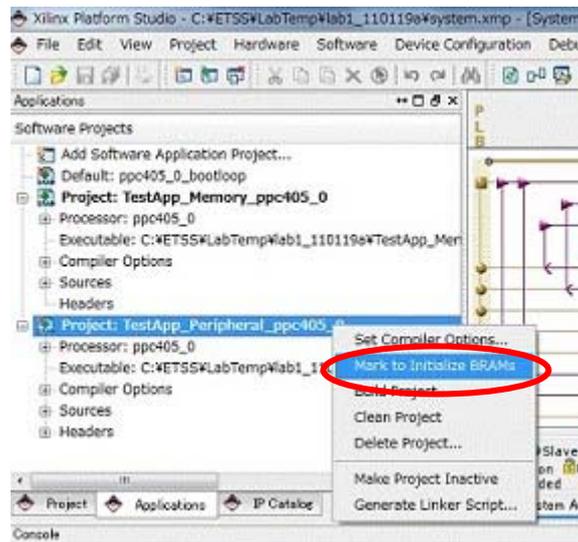
- RAM 構成は以下のようになっています
 - ✓ xps_bram_if_cntlr_0 64KB
 - ✓ xps_bram_if_cntlr_1 4KB
 - ✓ xps_bram_if_cntlr_2 4KB

- Instruction Cache/Data Cache : なし
- Boot Memory, Instruction, Data はすべて xps_bram_if_cntlr_0 に割り当ててください

確認画面で設定を確認の後、Finish を押してください。

- 3) “Start using Platform Studio” を選択し、
メニューの Hardware → Generate Bitstream を実行してください。
(多少時間がかかります)

- 4) Applications タブの「Project:TestApp_Peripheral_ppc405_0」を右クリックで選択し、
Mark to Initialize BRAMs にチェックを入れます。



- 5) メニューの Device Configuration → Update Bitstream を実行してください。

6) メニューの Device Configuration -> Download Bitstream を実行してください。

FPGA ボードの LED 点灯を確認してください。

再点灯させる時は、6) を実行してください。

7) Applications タブの” Sources” にある以下の C ソースを確認してください。LED を点灯させる記述になっています。

- ・ TestApp_Peripheral.c

- ・ xgpio_tapp_example.c

(xintc_tapp_example.c は今は関係ありませんので、確認不要です)

時間に余裕がある人は C ソースを変更しても構いません。変更後は5)、6) を再実行してください。

7) メニューの File -> Close Project でプロジェクトを終了してください。

Lab2: ループ文を使った LED ブリンク

《この Lab で習得する事》

C 言語の基本的な記述スタイルと PowerPC 及び Peripheral の初期化方法を習得します。

- 1) lab1 フォルダを同じ階層にコピーし、「lab2」にリネームしてください。
- 2) メニューの File -> Open Project で「lab2」フォルダの下の system.xmp を選択してください。 (※lab1 ではないので注意してください)
- 3) Applications タブの” Sources” と” Headers” に表示されているすべてのファイルを remove してください。
(remove はファイルを選択し、右クリック)
- 4) lab2¥TestApp_Peripheral_ppc405_0¥src の TestApp_Peripheral_LinkScr. Id 以外の*.c と*.h を削除します。
- 5) source_files¥forlab2¥TestAppLab2.c を lab2¥TestApp_Peripheral_ppc405_0¥src にコピーしてください。
- 6) TestAppLab2.c はループ文で LED ブリンクする記述です。未完成ですので incomplete の箇所に記述を追加し完成させてください。
- 7) Applications タブの” Sources” に TestAppLab2.c を追加してください。
(ファイル追加は、右クリックで Add Existing Files)
- 8) Update Bitstream, Download Bitstream を実行してください。
エラーが出るようでしたら、TestAppLab2.c を修正してください。
(※ソースコード以外のエラーは講師に聞いてください)

時間に余裕がある人は DS3, 11, 12, 13, 14 の LED もブリンクさせてみてください。

- 9) メニューの File -> Close Project でプロジェクトを終了してください。

Lab3: 割り込みを使った LED ブリンク

《この Lab で習得する事》

MHS ファイルの編集と割り込みの記述スタイルを習得します。

- 1) lab2 フォルダを同じ階層にコピーし、「lab3」にリネームしてください。
- 2) メニューの File -> Open Project で「lab3」の下の system.xmp を選択してください。
- 3) Applications タブの” Sources” の TestAppLab2.c を remove してください。
- 4) lab3¥TestApp_Peripheral_ppc405_0¥src¥TestAppLab2.c を削除します。
- 5) source_files¥for lab3¥TestAppLab3.c を
lab3¥TestApp_Peripheral_ppc405_0¥src にコピーしてください。
<ソースコード編集の前に以下の作業をします>
- 6) lab3 フォルダにある system.mhs の以下を修正してください。
 - ・ INSTANCE=LEDs_4Bit の以下のパラメータをコメントしてください。
PARAMETER C_INTERRUPT_PRESENT = 1
 - ・ INSTANCE=LEDs_4Bit の以下の PORT 宣言をコメントしてください。
PORT IP2INTC_Irpt = LEDs_4Bit_IP2INTC_Irpt
 - ・ INSTANCE=Push_Buttons_Position に以下のパラメータを設定してください。
PARAMETER C_INTERRUPT_PRESENT = 1
 - ・ INSTANCE=Push_Buttons_Position に以下の太字の行を追加してください。
BUS_INTERFACE SOPB = plb
PORT IP2INTC_Irpt = Push_Buttons_Positon_IP2INTC_Irpt
PORT GPIO_I0 = fpga_0_Push_Buttons_Position_GPIO_I0
 - ・ INSTANCE=xps_intc_0 の
PORT Intr = LEDs_4Bit_IP2INTC_Irpt
を以下のように書き換えてください。
PORT Intr = Push_Buttons_Positon_IP2INTC_Irpt

上記は Push ボタンの割り込み信号を割り込みコントローラに接続する修正です。

- 7) メニューの Hardware -> Generate Bitstream を実行します。
<実行中に下記のソースコード編集を行ってください>
- 8) TestAppLab3.c は、割り込みで LED ブリンクする記述です。未完成ですので incomplete の箇所に記述を追加し完成させてください。
- 9) Applications タブの” Sources” に TestAppLab3.c を追加してください。
- 10) [7が終了しているのを確認してから]
Update Bitstream, Download Bitstream を実行してください。
エラーが出るようでしたら、TestAppLab3.c を修正してください。
(※ソースコード以外のエラーは講師に聞いてください)
- 11) Push ボタンを押すたびに LED が点灯・消灯を繰り返す事を
確認してください。
Push ボタンは SW3-7 のどれでも構いません。

時間に余裕がある人は DS3, 11, 12, 13, 14 の LED もブリンクさせてみてください。

- 12) メニューの File -> Close Project でプロジェクトを終了してください

Lab4: Codec コントローラ

《この Lab で習得する事》

ユーザ IP の追加の仕方を習得します。

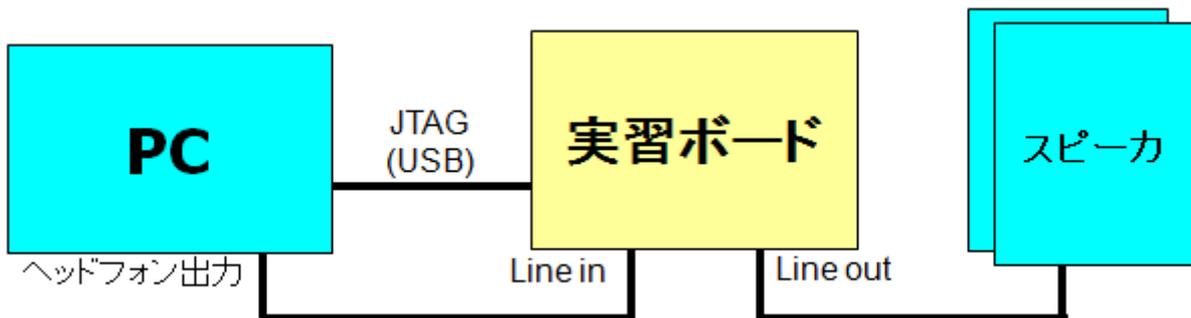
《実習ボード設定》

スイッチの設定は Lab1 を参照してください。

《実習ボード接続》

以下のように実習ボード、PC、スピーカを接続してください。

(電源コードは図から省いています)



- 1) lab3 フォルダを同じ階層にコピーし、「lab4」にリネームしてください。
- 2) メニューの File -> Open Project で lab4 の下の system.xmp を選択してください。
- 3) Applications タブの” Sources” の TestAppLab3.c を remove し、lab4¥TestApp_Peripheral_ppc405_0¥src¥TestAppLab3.c を削除してください。
- 4) メニューの Hardware -> Create or Import Peripheral を実行し、下記の要件を満たすユーザ IP を作成してください。
 - ・ Name : plb_codec_cntlr
 - ・ Version : 1.00.a
 - ・ 使用するバス : PLB
 - ・ “Interrupt control” はなし
 - ・ register 数 : 1(記載のない項目はデフォルト設定)
- 5) IP Catalog タブの” Project Local PCores” に PLB_CODEC_CNTLRLR がある事を確認し、右クリックで Add IP を選択してください。BASEADDR と HIGHADDR は以下に設定してください。

BASEADDR : 0x81460000
HIGHADDR : 0x81460fff
(他のパラメータはそのまま)

画面中央の System Assembly View 上に plb_codec_cntlr_0 が追加されている事を確認してください。

- 6) source_files¥for lab4¥vhdl の VHDL ファイルすべてを lab4¥pcores¥plb_codec_cntlr_v1_00_a¥hdl¥vhdl にコピーしてください。
(user_logic.vhd と plb_codec_cntlr.vhd は上書き保存してください)
- 7) lab4¥pcores¥¥¥vhdl の codec_if.vhd と user_logic.vhd は未完成です。
ソースファイル中の(1)～(4)の箇所に下記の動作をする記述を追加してください。
 - codec_if.vhd
 - (1) right_en1_d2==1 の時、rx_intr に 1 を代入し、
right_en1_d2==0 の時、rx_intr に 0 を代入
 - (2) left_en1_d1==1 もしくは right_en1_d1==1 の時、
pre_rx_wen に 1 を代入、それ以外は 0 を代入
 - (3) ある条件の時に pre_rx_addr に 01 を代入し、かつ pre_rx_data に right_data を代入
 - (4) pcm_right に代入する記述 (pcm_left の記述を参考にしてください)
 - user_logic.vhd
 - (1) im_rx_addr に以下のビットアサインで代入
 - (30 to 31) ← 00
 - (28 to 29) ← rx_addr (1 downto 0)
 - (0 to 27) ← All 0※im_rx_addr は Bit0 が MSB です。
 - (2) tx_data (19 downto 0) に im_tx_data (44 to 63) を代入
 - (3) リセット時は intr_trg に 0 を代入、
req_rx_buf==1 の時は intr_trg に 1 を代入、
req_rx_buf==0 の時は値を保持
(clock は positive edge を使用)

(4) codec_wrapper のインスタンス記述(rx_intr は rx_intr_w と接続してください)

8) lab4¥pcores¥…¥data の MPD ファイルと PA0 ファイルは未完成です。
source_files¥for lab4¥mpd_pao の MPD と PA0 を
lab4¥pcores¥…¥data に上書き保存後、incomplete 部分を完成させてください。

9) lab4 フォルダの system.mhs は未完成です。
source_files¥for lab4¥mhs の system.mhs を lab4 フォルダに
上書き保存後、incomplete 部分を完成させてください。

10) source_files¥for lab4¥ucf の system.ucf を lab4¥data に
上書き保存してください。
編集する必要はありませんが、内容は確認しておいてください。

11) メニューの Hardware -> Generate Bitstream を実行します。
(エラーが出るようでしたら修正してください)

<しばらく様子を見て、ツール処理が流れているようであれば、その間に下記のCソース修正を行ってください>

12) source_files¥for lab4¥src¥TestApp.c と cdc_hw.h を
lab4¥TestApp_Peripheral_ppc405_0¥src にコピーしてください。
TestApp.c は未完成です。
incomplete の箇所に記述を追加し完成させてください。
cdc_hw.h は編集の必要はありませんが、内容を確認しておいてください。

13) [11が終了していたら]
Applications タブの”Sources” に TestAppLab3.c を、
”Headers” に cdc_hw.h を追加してください。

14) Update Bitstream, Download Bitstream を実行してください。

15) PC で音楽データを再生してください。音が聞こえなければ C ソースに問題があるかもしれません。見直してみてください。

Lab5: ノイズ付加

《この Lab で習得する事》

IIR フィルタを利用したソフト処理が行えるようになります。

- 1) lab4 フォルダを同じ階層にコピーし、「lab5」にリネームしてください。
- 2) TestApp.c を編集して、IIR フィルタを利用した下記のサイン波「ノイズ」を生成し、元の音楽データに付加してください。

- ・ サイン波の周波数 5KHz
- ・ サンプリング周波数 48KHz

ノイズ付加は割り込みハンドラ (GDC_Handler) で行ってください。

- 3) 記述が完成したら、Update Stream, Download Stream を実行してください。
音楽に混じって、異音が聞こえるはずです。
- 4) 次はプッシュスイッチを使ってノイズの ON/OFF をする記述を追加してください。
 - ・ プッシュスイッチが押されている間、ノイズ OFF
 - ・ プッシュスイッチを離している間、ノイズ ON

プッシュスイッチからの値は GPIO を使って取得してください。
GPIO で入力値を取得する記述はテキスト第 3 章を参考にしてください。

- 5) 完成したら、ダウンロードして ON/OFF できるか試してみてください。

Lab6: LCD 表示

《この Lab で習得する事》

GPIO を利用して外部機器 (LCD パネル) を制御できるようになります。

- 1) lab5 フォルダを同じ階層にコピーし、「lab6」にリネームしてください。
- 2) MHS ファイル、MSS ファイル、UCF ファイルを以下の要領で編集してください。
 - ・ MHS ファイル
source_files¥for lab6¥mhs の system.mhs の中身をコピーし、lab6 フォルダの MHS ファイルに追加で貼り付けてください。
 - ・ MSS ファイル
source_files¥for lab6¥mss の system.mss の中身をコピーし、lab6 フォルダの MSS ファイルに追加で貼り付けてください。
 - ・ UCF ファイル
source_files¥for lab6¥ucf の system.ucf の中身をコピーし、lab6¥data フォルダの UCF ファイルに追加で貼り付けてください。
- 3) ツールで lab6 の system.xmp を開き、Generate Bitstream を実行します。
(エラーが出るようでしたら修正してください)

くしばらく様子を見て、ツール処理が流れているようであれば、その間に下記の C ソース修正を行ってください>

- 4) TestApp.c を編集して、LCD パネルに以下を表示する機能を追加してください。
 - ・ サイン波「ノイズ」の周波数
 - ・ サイン波「ノイズ」の状態 (ON または OFF)

記述が難しい場合は complete フォルダ内のソースコードを参考にしてください。
complete¥lab6¥TestApp_Peripheral_ppc405_0¥src¥TestApp.c

- 5) 記述が完成したら、Update Stream, Download Stream を実行してください。
LCD パネルに文字列が表示されるはずですが、
プッシュスイッチを ON/OFF して、LCD パネルの表示内容が変わる事を確認してください。