

1 GP600M 環境を利用するのに必要なホストの情報

GP600M 環境は目前の Windows マシンに接続されていない。そのため、設計したデータの中島研ファイルサーバに移動した上で、論理合成/マッピング、および、FPGA での動作検証を行う必要がある。以下は、GP600M 環境を利用するのに必要なホストの情報である。

これらのホストにログインするためには中島研内の NIS アカウントが必要であるが、中島研以外の受講生のユーザアカウントは Mandara アカウントと同じであり、初期パスワードは xxx(実習中に教える) である。

- arch.naist.jp

中島研ファイルサーバ。ここに転送したファイルは以下の cad10[12].naist.jp、および、alice11[0-3].naist.jp から参照可能なため、設計データはまずここに転送する。

- cad101.naist.jp, cad102.naist.jp

FPGA 向け論理合成/マッピングツールが入っているホスト。ここでファイルサーバに転送した設計データに対し、FPGA 向け論理合成/マッピングを行う。

- alice110.naist.jp, alice111.naist.jp, alice112.naist.jp, alice113.naist.jp

GP600 ボードが接続されているボード。FPGA 向け論理合成/マッピング終了後、このホストで FPGA コンフィギュレーションを行い、esim を起動し、esim より設計した回路に対する操作を行う。

GP600 ボードは複数人で同時に利用できないため、利用時には「alice11[1-4] 使用中」の物理タグを得た上で利用すること。また、物理タグ返却時には esim や FPGA コンフィギュレーションプログラムの動作を停止させてから返すこと。

2 GP600M 環境用 FPGA の論理合成方法

2.1 ストップウォッチ用テンプレートの利用

D:\¥ProjectPractice¥templates¥chronograph_template にテンプレートがあるので、これをコピーしてユーザ論理を追加する。

chronograph_template のディレクトリ構成は以下のようになっている

```
+chronograph\_template
- pe0.v
- ...
+ top
- Make-top-synplify.sh
- Make-top-synplify.prj
- ...
```

ユーザ論理追加

chronograph_template 下の pe0.v がユーザ論理を追加可能な Verilog ファイルである。82 行目より、155 行目がストップウォッチ用にあらかじめこちらで準備した資源とそれに関する記述が行われている。

以下が pe0.v に準備されている資源である。必要に応じて、不要な資源を削除したり、仕様を変更してもかまわない。

- 25MHz のクロック (clock)
- ハードリセット信号 (reset)
- 16 本の 64bit 結果出力用の wire(output_array)
- esim 側から操作される 8bit のフラグレジスタ FREG(FREG[7:0])
- 周期 100ms のクロック (clock_100ms)
- esim 側から操作される 8bit のフラグレジスタ FREG(FREG[7:0])

余談: GP600M は真のハードリセット信号 (RST) の posedge でリセットがかかるように設計されている。テキストではリセット信号は negedge で利用する形となっているので、テキストに合わせるため、reset 信号は RST 信号の否定となっている。

- 16 本の 64bit 結果出力用の wire(output_array)
output_array の信号は最終的に DREG に代入され、ホスト PC の esim から読み出される。
- 周期 100ms のクロック (clock_100ms)
- esim 側から操作される 8bit のフラグレジスタ FREG(FREG[7:0])
- 周期 100ms のクロックに同期したフラグレジスタのエッジ検出 (posedge_FREG[7:0], negedge_FREG[7:0])

ストップウォッチへの入力のサンプルとして、タイムリセット (time_reset) を FREG[0] の立ち上がり信号に、スタート/ストップ (start_stop) を FREG[1] の立ち上がり信号に、ラップ (lap) を FREG[2] の立ち上がり信号に、それぞれ wire を定義して接続してある。

なお、ハードリセットは GP600M 自体のリセットと共用しているため、esim shell からリセット要求を出すことになる。

- ストップウォッチモジュールへの接続のサンプル

なお、新たな Verilog ファイルを作成した場合、top/Make-top-synplify.prj の以下の場所に
“add_file -verilog ”新たな Verilog ファイル”” の形で追記することを忘れないようにすること。
これは、Xilinx ISE においてプロジェクトに Verilog ファイルを追加するのと同じである。

```
# Make-top-synplify.prj
# created by 20080609 kazuhiko-y
#

#add user file as belows ←ここに追加
#add_file -verilog "../userfile.v"
add_file -verilog "../chronograph.v" ←追加例
add_file -verilog "../counter10en.v" ←追加例
add_file -verilog "../counter6en.v" ←追加例

#add_file options
add_file -verilog "../ssramif.v"
```

FPGA 向け論理合成/マッピング

cad10[12].naist.jp に FPGA 向け論理合成/マッピングツール (Synopsys 社 Synplify) が入っている。そのため、以降の作業はまず WinSCP を用いて設計データを arch.naist.jp に転送した上、cad10[12] にログインした上で行う。

以下に、FPGA 論理合成/マッピングのための一連の処理を示す。

```

cad102% source /opt/setup/fpga_tools.cshrc.gp800
cad102% cd chronograph_template/top
cad102% less Make-top-synplify.prj
# Make-top-synplify.prj
# created by 20080609 kazuhiko-y
#

#add user file as belows
#add_file -verilog "../userfile.v"
add_file -verilog "../chronograph.v"
add_file -verilog "../counter10en.v"
add_file -verilog "../counter6en.v"
----後略----
cad102% ./Make-top-synplify.sh
*****

Warning: You are running on an unsupported platform
'synplify_pro' only supports Red Hat Enterprise Linux 4.0 and above

current platform: CentOS release 5.8 (Final)
Kernel \r on an \m

*****

SystemDesigner executable set to
/opt/synopsys/synplify_pro_vD-2010.03-SP1/system_designer/bin/system...
----中略----
Release 10.1.03 - Promgen K.39 (lin64)
Copyright (c) 1995-2008 Xilinx, Inc. All rights reserved.
0x29b474 (2733172) bytes loaded up from 0x0
Using generated prom size of 4096K
Writing file "top.mcs".
Writing file "top.prm".
cad102%

```

1つ目のコマンドは、論理合成/マッピングツールの設定ファイルを読み込む動作である。3つ目のコマンドは、新たに設計した Verilog ファイルが Make-top-synplify.prj に記述されているかの確認である(省略してもかまわない)。4つ目のコマンドが、論理合成/マッピングのシェルスクリプトの起動である。

正常に論理合成/マッピングが行われた場合、「Writing file "top.mcs".」が出力されて終了する。この top.mcs が FPGA の構成情報であり、次節で構成情報書き込みに利用する。

なお、論理合成/マッピングが正常に終了しなかった場合、以下の点をまず疑うこと。

- 呼び出されているモジュールがファイル内に存在しない

新たに設計した Verilog ファイルが Make-top-synplify.prj に記述されていないと、この問題により正常終了しません。

- 内部処理を実行できない always 文ができてしまっている

設計中に posedge reset と negedge reset が混ざってしまうとこの問題により正常終了しません。GP600M 環境は posedge reset でリセットする形で設計されているため、思い切って posedge reset でリセットする形に設計を改めてしまうのも手です。

2.2 プロセッサ用のテンプレートの利用方法

D:\¥ProjectPractice¥templates¥processor_template にテンプレートがあるので、これをコピーしてユーザ論理を追加する。

processor_template 下の pe0.v がユーザ論理を追加可能な Verilog ファイルである。82 行目より、143 行目がプロセッサ用にあらかじめこちらで準備した資源とそれに関する記述が行われている。

以下が pe0.v に準備されている資源である。必要に応じて、不要な資源を削除したり、仕様を変更してもかまわない。

- 25MHz のクロック (clock)
- リセット信号 (reset)
- esim 側から操作される 8bit のフラグレジスタ FREG(FREG[7:0])
- フラグレジスタのエッジ検出 (posedge_FREG, negedge_FREG)
サンプルとして、実行 (exec) を FREG[0] の立ち上がり信号に接続してある。
なお、リセットは GP600M 自体のリセットと共用しているため、esim shell からリセット要求を出すことになる。
- 命令メモリアドレス (iram_addr[7:0])
命令メモリは DREG の 0-255 に割り当てられている。
- 命令メモリ読み出しデータ (iram_data[15:0])
- データメモリアドレス (dram_addr[7:0])
データメモリは DREG の 256-512 に割り当てられている。
- データメモリ読み出しデータ (dram_rdata[15:0])
- データメモリ書き込みデータ (dram_wdata[15:0])
- データメモリ書き込み信号 (DREG_we)
(DREG_we == 1) の時に dram_wdata[15:0] がデータメモリ (実体は DREG) に書き込まれる。
- プロセッサモジュールへの接続のサンプル

3 esim の使い方

D:¥ProjectPractice¥templates¥esim に esim が置いてあるので、これをファイルサーバ (arch.naist.jp) に移動させた上、alice11[0-3] から使用する。

以下は、esim のビルド方法である。

```
alice110% cd esim
alice110% make clean
alice110% make
```

3.1 FPGA コンフィギュレーション

FPGA コンフィギュレーションは esim ディレクトリにある Config-gp600m を用いて行う。実行の書式は、“./Config-gp600m FPGA 構成情報ファイル” である。

以下に、Config-gp600m を用いて FPGA コンフィギュレーションを行う様子を示す。Config-gp600m を立ち上げるとモニタが出るため、“[L]: Down Load” と “[Q]: Exit” を利用してコンフィギュレーションを行う。

```
alice110% cd esim
alice110% ./Config-gp600m ../sample_chronograph/top/top.mcs
Driver Module Open --

* * * * * GP600M-PCI TINY MONITOR * * * * *
[L]: Down Load
[3]: BAR3 Access (For Memory)
[Q]: Exit
* * * * *
L
-- [DownLoad] --
Reset
Ready to Configuration!
Ready to Flash ROM!
Downloading..

Complete!

* * * * * GP600M-PCI TINY MONITOR * * * * *
[L]: Down Load
[3]: BAR3 Access (For Memory)
[Q]: Exit
* * * * *
Q
alice110%
```

なお、複数のプロセスが FPGA を操作しようとするとう PCI バスが高い確率でハングアップするため、FPGA コンフィギュレーション前に「alice11[1-4] 使用中」の物理タグを得た上、他のターミナルで esim や Config-gp600m が動作してないことを確認したでコンフィギュレーショ

ンや esim の起動を行うこと。また、esim や Config-gp600m から応答が無くなり、PCI バスがハングアップしたと思われる場合は、シェルから reboot コマンドを実行すること (ユーザから実行できます)。

3.2 esim からユーザ回路操作

esim は以下の形で起動します。

```
./esim [options] (mem_file)
```

mem_file はメモリイメージのバイナリファイルであり、指定された場合は DREG は mem_file の内容で初期化されます。指定されなかった場合、DREG は全てのビットが "1" で埋められます。

以下は、無引数で esim を立ち上げた場合です。

```
alice110% ./esim
==== GP600M Host Program Version 1.1 2011/03/21 09:32:57 nakashim Exp nakashim...
trace_flag=0x00000000
FREG[7:0]: 00000000
DREG[ 0]: 00000000_00000000 00000000_00000000 00000000_00000000 00000000_00000000
DREG[ 4]: 00000000_00000000 00000000_00000000 00000000_00000000 00000000_00000000

esim_shell>
```

3 行目は現在の FPGA 内の FREG の状態です。5,6 行目は現在の DREG の状態です。DREG は 64bit 幅のため、デフォルトでは 64bit 幅で表示されます。また、デフォルトでは DREG[0-7] までが表示されます。この DREG の表示については、起動時の引数で変更できます。7 行目は esim shell のコマンド受け付け状態です。ここから 1 文字のコマンドで esim を操作します。“h” でヘルプが出ますので、迷ったら “h” を押して下さい。“q” で esim が終了します。

esim の引数オプション

esim の引数オプションを以下に示します。esim 起動時に存在しない引数オプションを指定すると (例: ./esim -h)、引数オプションの簡単な説明が出ます。

- -a: 自動 DREG 読み出しを有効化
一定時間間隔で DREG を読み出し (esim shell の “d” コマンドに同じ) を行います。読み出し間隔は -i オプションで指定します。
- -i: 自動 DREG 読み出し間隔指定
-a オプションや esim shell 上の “a” コマンドを利用時の、自動 DREG 読み出し間隔を指定します。単位は 1/128 秒で、デフォルト値は 64(0.5 秒) です。
- -l: mem_file の長さ指定
mem_file を指定した時、DREG に書き込む長さを指定します。デフォルト値は 128 ワードです。
- -bf: FREG のビット反転処理を反転のみ (書き込みなし) に変更
esim shell 上の “0”-“7” コマンドによる FREG のビット反転は、デフォルトでは反転したビットを FPGA 側へ書き込みます。これを、反転のみで書き込みを無しにします。この場合、書き込みを行いたい場合は esim shell 上から “d” コマンドを入力して下さい。一度に複数 bit を反転させたい場合、このモードで複数ビットを反転させた後、“d” コマンドで書き込んで下さい。

- -bi: FREG のビット反転処理の自動回復間隔指定
次の-br オプションのビット反転処理の自動回復間隔を指定します。単位はミリ秒で、デフォルト値は 200 ミリ秒です。
- -br: FREG のビット反転処理の自動回復をゆう奥か
esim shell 上の”0”-”7” コマンドによる FREG のビット反転は、デフォルトでは反転したビットを FPGA 側 側に書き込むだけであり、ビット反転の立ち上がりパルスを利用するような回路組んだ場合、次の立ち上がりパルスを入力するためにはもう一度コマンドを送って反転したビットを元に戻さなくてはなりません。このオプションをつけると、一定時間 (デフォルト 200 ミリ秒、-bi オプションで指定) で反転したビットが元に戻ります。
- -dc: DREG の列方向表示数指定
読み出された DREG の列方向の表示個数です。ターミナルの表示幅に応じて調整して下さい。デフォルト値は 4 です。
- -dn: DREG の表示個数
DREG の読み出し個数です。デフォルト値は 8 です。プロセッサのテンプレートの場合、データメモリは DREG[256-512] にマッピングされますので、必要に応じて、そこまで表示して下さい。
- -dw: DREG の表示幅
DREG は 64bit 幅 (8 バイト) で準備されていますが、上位ビットを使わない場合、上位部分をこのオプションで非表示にできます。単位はバイトで、デフォルト値は 8 バイトです。他に 4 バイト、2 バイト、1 バイトが指定できます。
- -vg: GP600M との通信状態の表示
GP600M との通信状態の表示します。GP600M との通信が途中で止まる時の確認用です。
- -vh: ホストに保存されている DREG の初期値を表示
ホストに保存されている DREG の初期値を表示します。この値は、esim shell 上の”w” コマンドを利用して、任意のタイミングで書き込むことが可能です。

esim shell のコマンド

esim shell のコマンドを以下に示します。“h” コマンドでも表示されます。

- 0-7: FREG の対応するビット反転
FREG の対応するビットを反転します。デフォルトでは反転させた値を FPGA に書き込みますが、反転させるのみで書き込まないこと (“b” コマンド)、反転させた値を自動的に元に戻すこと (-br 引数オプション) もできます。
- a: DREG 自動読み出し有効化/無効化
DREG 自動読み出しを有効化/無効化します。自動読み出しの間隔は-i 引数オプションで指定します。
- b: FREG のビット反転時の FPGA 書き込み有効化/無効化
“0”-”7” コマンドによって反転させた FREG のビットを FPGA に書き込む動作を有効化/無効化します。無効化された場合、FREG の書き込みは”d” コマンドを実行するまで行われません。複数ビットを同時に反転させたい場合に、書き込み動作を無効化した上で複数ビットを反転させ、“d” コマンドでまとめて書き込む形で使います。

- d: DREG 読み出し
DREG や FREG を読み出します。また、FREG の書き込みも行われます。
- h: ヘルプ表示
- q: 終了
- r: GP600M リセット
GP600M とユーザ回路をリセットし、DREG に初期値を書き込み直します。ハードリセット扱いで利用します。
- w: DREG 書き込み
esim 上にある DREG 初期値を書き込みます。

おすすめの esim の使い方

- `./esim -a -i 8 -dw 4 -dc 8 -br -bi 200`
ストップウォッチの動作テスト時におすすめ。
自動 DREG 読み出し、自動 DREG 読み出し間隔 1/16 秒、DREG 表示幅 4 バイト、DREG 列方向表示数 8 個、FREG の bit を”1”に反転した後に自動的に”0”に戻す、FREG の自動戻しの間隔 200 ミリ秒。
- `./esim -dw 2 -dc 16 -dn 512 -br -bi 200 mem_file`
プロセッサの動作テスト時におすすめ。mem_file には命令を格納したメモリイメージを指定する。DREG は必要に応じて”d”で読みだす。
DREG 表示幅 2 バイト、DREG 列方向表示数 16 個、DREG の表示個数 512 個、FREG の bit を”1”に反転した後に自動的に”0”に戻す、FREG の自動戻しの間隔 200 ミリ秒。