ALTERA FPGA への接続

須山 敬之
NTT コミュニケーション科学研究所
suyama@cslab.kecl.ntt.co.jp

1 はじめに

ここでは, 論理合成システム PARTHENON と ALTERA FPGA との接続について説明致します. 基本的には PARTHENON と ALTERA FPGA との接続は PARTHENON の論理合成系からネットリストを出力し, それを ALTERA FPGA のマッピングツールに入力することにより実現します. この操作は他の FPGA についてもほぼ同様ですが, それ以外に ALTERA 特有の注意点についても説明致します.

2 ALTERA

2.1 デバイス

ALTERA は表 1 のように多様な FPGA[1] を製品としてラインアップしています. 内部論理を書き換えるテクノロジとして EPROM, EEPROM, SRAM の三種類を採用しています. 各デバイスの詳細については [1], [2] 等を参照してください.

表 1: ALTERA のデバイスファミリー

Device Family	Technology	Usable Gates	Pin Count
Classic	EPROM	150–900	24–68
MAX5000	EPROM	600–3,750	28–100
MAX7000E/S/A	EEPROM	600–20,000	44–256
MAX9000A	EEPROM	6,000–120,000	84–356
FLEX6000/6000A(3.3V)	SRAM	5,000–24,000	144–256
FLEX8000A	SRAM	2,500–16,000	84–304
FLEX10K/10KV(3.3V)	SRAM	10,000–130,000	84–503
FLEX10KA(3.3V)/10KE(2.5V)	SRAM	10,000–250,000	84–600

2.2 マッピングツール

ALTERA のマッピングツールは MAX+plus II と呼ばれており, 回路図やハードウェア記述言語によるデザイン入力, コンパイル (FPGA へのマッピング), シミュレーション, タイミング解析, デバイスへのプログラミング等の機能があります. また, 動作環境として Windows ベース PC, SUN Sparc Station, HP9000 シリーズ, IBM RISC System/6000 シリーズをサポートしています.

3 PARTHENON と MAX+plus II の連携

本稿では, PARTHENON Version 2.3.x, MAX+plus II Version 8.3 を想定しています.

[1] ALTERA 自身は FPGA ではなく, CPLD と呼んでいます.

ご存じのように PARTHENON はハードウェア記述言語 SFL を用いてハードウェアの動作を記述し, それをシミュレータ (seconds) を用いて動作のシミュレーションを行い, 論理合成の処理系 (sflexp, opt_map 等) を用いて, 論理回路を合成するシステムです. SFL では回路が実際のデバイスとしてどのように実現されるのかを考える必要がなく, より抽象度の高いレベルで記述することができます. メーカの違い等による物理的な制約条件はセルライブラリという形で保持され, それを用いて制約違反を起こさないよう回路が合成されます. PARTHENON は元々, ゲートアレイやスタンダードセルをターゲットとして設計されているため, 標準的なセルライブラリは NAND ゲート, NOR ゲートといったゲートと相性がよい作りとなっています. そのため LUT (Look-Up Table) をベースとした従来の LSI と異なる構造を持った FPGA にはそのままではなじみません.

しかし, FPGA マッピングツールは従来の論理合成ツール等との互換性を取るため, ゲートアレイ等と同様のゲートから構成されたネットリストを入力として受け付けるようになっており, そのフォーマットとして業界標準である EDIF が使用されます. よって, PARTHENON と FPGA マッピングツールは EDIF を用いてお互いのインターフェイスを取ることが可能となります.

SFL 記述からマッピングまでの流れは以下のようになります.

(SFL 記述) ⇒ PARTHENON ⇒ (EDIF 形式のネットリスト) ⇒ MAX+plus Ⅱ ⇒ (マッピングデータ)

4 連携の手順

それでは, PARTHENON と MAX+plus Ⅱの基本的な連携の手順を説明致します.

4.1 PARTHENON

4.1.1 前準備

前述のように, PARTHENON はセルライブラリによりターゲットとなるデバイスに対応することができます. ALTERA の FPGA をターゲットとする場合, MAX+plus Ⅱが受け入れることができる要素 (cell) から構成されたセルライブラリを使用します. このような ALTERA 用のセルライブラリとしては既に公開されているものがありますので, それを使用すると良いでしょう[2].

4.1.2 ネットリストの合成

上記のセルライブラリは $PARTHENON/cell_lib.dir/ALTERA/altera/ に格納しますので, auto コマンドを使用する場合, 以下の手順で ALTERA 用のネットリストを得ることができます[3]. <module> には SFL 記述のトップモジュール / ファイル名を指定してください.

```
% auto <module> ps ALTERA altera
```

エラーがなく終了したならば, 作業中のディレクトリに <module>.edf という EDIF 形式のネットリストのファイルが作成されています.

4.2 MAX+plus Ⅱ

次に MAX+plus Ⅱでの手順を説明致します.

4.2.1 前準備

MAX+plus Ⅱでは, EDIF 形式のネットリストを読み込む際にライブラリマッピングファイルを用います. PARTHENON 用のライブラリマッピングファイルは partenon.lmf[4]です. 前述の ALTERA 用セルライブラリのパッケージに含まれていますので, それを MAX+plus Ⅱから参照可能なところに置いて下さい.

[2] PARTHENON のホームページ (http://www.kecl.ntt.co.jp/car/parthe/) より入手が可能です.

[3] この例では PostScript 形式の回路図も得られますが, ネットリストを得るだけならば ps の代わりに nld4 を指定すれば十分です.

[4] parthenon.lmf ではありませんのでご注意ください. DOS 時代のファイル名の制限の名残です. 気になる方は rename してください.

4.2.2 マッピング

以下にマッピングの手順を示します.

1. MAX+plus Ⅱ を起動します.
2. File→Project→Name... を開き, Project Name に <module>.edf を指定し, OK を押します.
3. MAX+plus II→Compiler を開きます (図 1, 但しコンパイル終了時の図).
4. Interfaces→EDIF Netlist Reader Settings... を開き, Customize >> を押します (図 2).
5. Signal Names の VCC に VDD を, GND に VSS を, それぞれ設定します†.
6. Library Mapping Files に partenon.lmf を設定し, OK を押します.
7. 必要ならば, Assign→Device... や Assign→Pin/Location/Chip... 等から, デバイスやピンの割り当てを設定します.
8. Start を押します.

†PARTHENON では電源ピンの名称が VDD と VSS に固定されています.

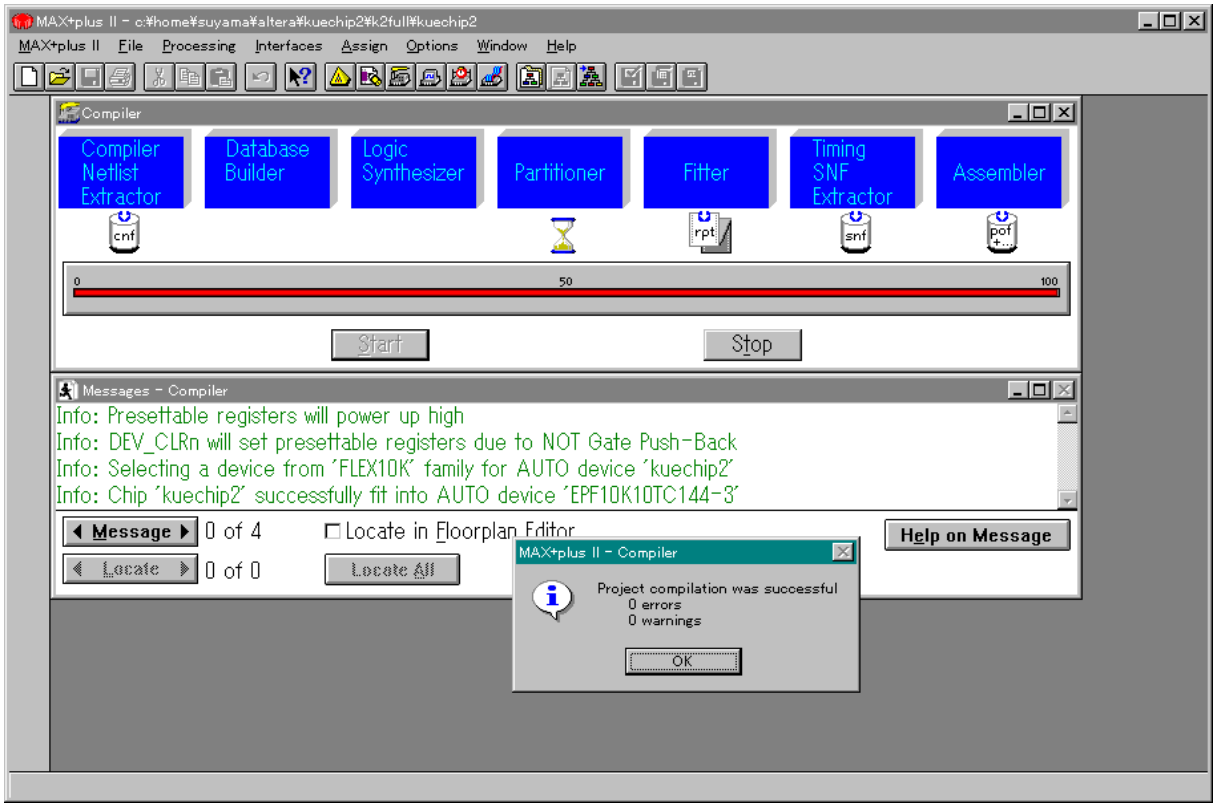

図 1: MAX+plus Ⅱ のコンパイラ

エラーなく終了したならば, マッピングは完了です. コンパイルされたマッピングデータをデバイスに転送することにより, デバイスを動作させることができるようになりますが, データの転送方法についてはデバイスファミリーごとに異なりますので, マニュアル等を参照してください. ピンの割り当てや内部の使用率等の情報は <module>.rpt というファイルに出力されています. テキスト形式で出力されていますので, テキストエディタ等で見ることができます.

また, MAX+plus II→Timing Analyzer... を開き, 更に Analysis→Registered Performance... を指定し, Start を押すことにより, 動作可能クロック周波数を見積ることができます (図 3).

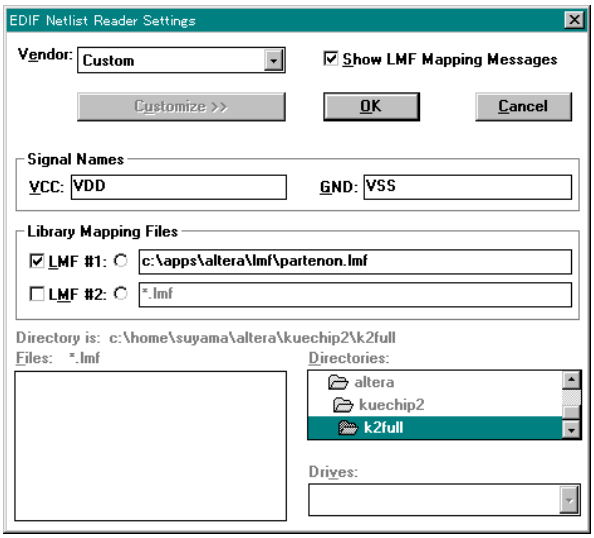

図 2: EDIF Netlist Reader Setting

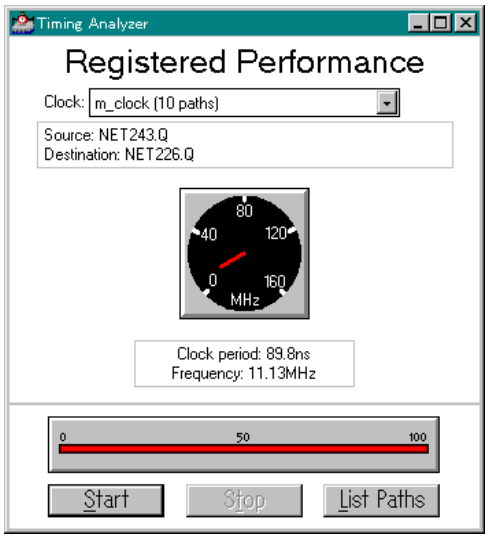

図 3: Timing Analyzer

5 EDIF (Electronic Design Interchange Format)

EDIF は広範な電子的データのやり取りをするために定められた標準フォーマットです．これまでに幾つかのバージョンが定められていますが，ここでは Version 2 0 0 (以下 EDIF200) を使用します．フォーマットの詳細は [3] を参照してください．

5.1 PARTHENON から出力される EDIF

EDIF200 のネットリストは PARTHENON の合成系のプログラムの一つである `opt_map` より出力されます[5]．`opt_map` は `auto` コマンドから使用されることが多いプログラムですが，本来はコマンド形式の対話型プログラムです．EDIF200 形式のネットリストは `opt_map` に `edif` というコマンドを与えることにより出力されます．

```
OPT_MAP> edif example.edf
** check level 1 **
pcd module nand2 written
          :
example.edf written
OPT_MAP>
```

ところで，EDIF200 では `cell` 等の「名前[6]」に使える文字がアルファベット，数字，'_' (underscore) に制限されていますが[7]，`edif` コマンドのみでは，内部のモジュール名，ピン名等の名前がそのまま用いられたネットリストが出力されます (すなわち EDIF200 では不正な名前でもそのまま出力されてしまう)．そこで `opt_map` には全ての名前を EDIF200 に適合する名前に変更する `cnn` というコマンドが用意されています．不正な名前を出力しないようにするためには `edif` コマンドの前に必ず `cnn` コマンドを実行する必要があります．

```
OPT_MAP> cnn
OPT_MAP> edif example.edf
          :
```

5.2 セルライブラリ

正常に MAX+plus Ⅱでネットリストを読み込むためには，MAX+plus Ⅱが解釈できる要素 (`cell`) のみでネットリストが構成されている必要があります．一方，PARTHENON でもあらかじめその存在が規定されている論理 `cell`

[5] `$PARTHENON/com/` にある `nld_edif` は EDIF Version 1 1 0 を出力しますのでご注意下さい．
[6] 正確には *identifier*．
[7] 詳しくは [3] を参照してください．

があり, 初期回路はそれを用いて合成されます. 両者で対応している cell は良いのですが, 片方にしか存在していない cell は何らかの形で整合を取る必要があります. PARTHENON ではその不整合をセルライブラリを用いて吸収しています.

MAX+plus Ⅱでどのような cell が使用可能かは MAX+plus Ⅱのヘルプファイル等で調べることができます. PARTHENON のセルライブラリでは記述言語として PCD[8] と NLD[9] を用いますが, MAX+plus Ⅱで受付可能な cell に対応する論理 cell についてはそのまま PCD で記述し, 存在しない論理 cell については NLD により, その論理 cell と等価な回路を記述することにより実現します. 下は reg---1.pcd を reg---1.nld[10] で置き換えた例です. 図 4 は reg---1.nld の回路図表現です. PCD, NLD の詳細については [4], [5] を参照してください. またセルライブラリの作成方法については [6] を参照してください.

reg---1.pcd (DEMO/demo)

```
(def-module reg---1 power 23 area 3.4 gates 12

(def-pin set     type input  load 0.25)
(def-pin reset   type input  load 0.13)
(def-pin m_clock type input  load 0.12 note clock)
(def-pin s_clock type unused)
(def-pin p_reset type input  load 0.17)
(def-pin out     type output load 0.21)
(def-pin nout    type output load 0.089)

(def-symbol REGISTER set reset m_clock s_clock p_reset out nout)

(def-logic out (free_reg
(or set (and ^reset out)) 0)
)
(def-logic ^nout out)

(def-function out   (reg---1 m_clock p_reset set reset))
(def-function ^nout (reg---1 m_clock p_reset set reset))

(def-delay /m_clock /out    (+ 1.6 (* 1.8 ~out)))
(def-delay /m_clock /nout   (+ 2.0 (* 1.8 ~nout)))
(def-constraint set_setup   (- (+ /m_clock ?cycle) /set ...)
                             :
(def-constraint drive (- 0 ~nout) type max)
)
```

⟹

reg---1.nld (ALTERA/altera)

```
(def-module reg---1

    (def-pin set     type input)
    (def-pin reset   type input)
    (def-pin m_clock type input)
    (def-pin s_clock type unused)
    (def-pin p_reset type input)
    (def-pin out     type output)
    (def-pin nout    type output)

    (def-comp dff- reg)
    (def-comp nor--2 and)
    (def-comp nor--2 or)
    (def-comp inv- inv)
    (def-comp high- high)

    (def-net set or.in2)
    (def-net reset and.in2)
    (def-net m_clock reg.clk)
    (def-net p_reset reg.prn)
    (def-net out inv.nout)
    (def-net nout reg.q and.in1 inv.in)
    (def-net and.nout or.in1)
    (def-net or.nout reg.d)
    (def-net high.out reg.clrn)
)
```

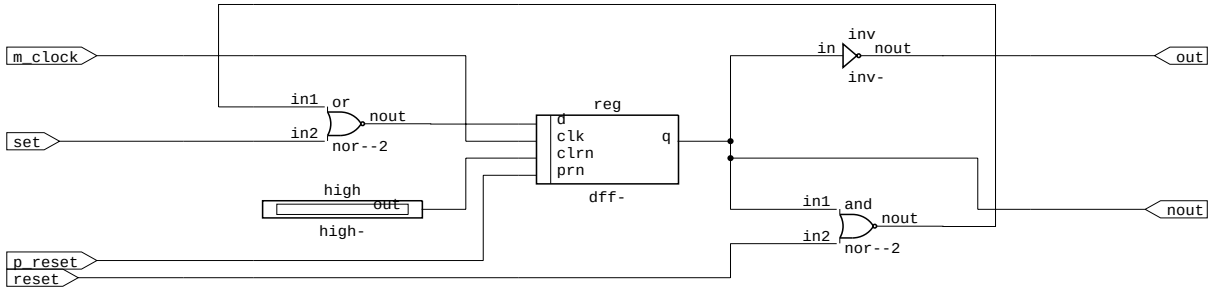

図 4: reg---1.nld

[8] Physical Characteristic Description. 論理回路の最下位要素を記述する. 遅延等の物理特性の情報も記述される.

[9] Net List Description. ネットリストを記述する.

[10] リスト中の dff- は MAX+plus Ⅱで用意された DFF に対応するダミーセルです.

5.3 ライブラリマッピングファイル

EDIF 形式でネットリストを表わす場合, 論理回路は内部を構成する要素 (AND/OR ゲート等) とその接続で表現されます. 内部に含まれる構成要素は cell として以下のように表わされます.

```
(cell and2
 (cellType GENERIC) (view VIEW (viewType NETLIST)
  (interface
   (port in1 (direction INPUT))
   (port in2 (direction INPUT))
   (port out (direction OUTPUT))
  )
 )
)
```

この例では “and2” という文字列は単に cell を識別するための名前として用いられ, 機能を表わしているのではありません. そのため, この cell がどのような機能を持っているかを認識するための手段としてライブラリマッピングファイルを用います. 以下にライブラリマッピングファイルの例 (一部) を示します.

```
BEGIN
FUNCTION        and2 ( IN1, IN2 )
RETURNS         ( OUT )
FUNCTION        "and2"( "in1", "in2" )
RETURNS         ( "out" )
END
```

この例では, MAX+plus Ⅱの基本ゲートである and2 と EDIF ファイルの "and2"が対応していることを表わしています.

6 ピンの極性

SFL にはピンの極性という考え方はなじみませんが, 実際にデバイスを動作させる場合には論理値の “0”/“1”, 論理レベルの “L”/“H”, “低電位側”/“高電位側” 等の対応を取る必要があります. 通常, “0” と “低電位側”, “1” と “高電位側” が対応する場合が多いのですが, セルライブラリの作りによって必ずしもそうであるとは限りません. 例えば, ALTERA 用のセルライブラリの reg-1.nld, reg--1.nld では flipflop のプリミティブとして MAX+plus Ⅱの DFFE に対応する dffe- が用いられており, 合成の際に付け加えられるグローバルな外部端子である p_reset は DFFE の CLRN 端子に直結するようにライブラリが作られています. この様な場合, p_reset に加えられる入力により回路がどのように振る舞うかは CLRN 端子の定義そのものになります. DFFE は CLRN 端子が “L” の時, flipflop の出力が “L” となり, “H” の時, flipflop としての通常の動作をしますので, 通常は高電位にしておき, リセットする時に電圧を下げる必要があります. 逆に電圧が上がった時にリセットがかかるような動作を行う回路を得るためには, opt_map の inv コマンドか, MAX+plus Ⅱの回路図入力等を用いてピンの極性を反転させる必要があります.

参考文献

[1] *Data Book*, ALTERA, 1998.

[2] *ALTERA Home Page*, http://www.altera.com/.

[3] *Electronic Design Interchange Format Version 2 0 0*, Electronic Industries Association, ISBN 0-7908-4, May 1987.

[4] 小栗 清, 名古屋 彰, 野村 亮, 雪下 充輝: 初めての PARTHENON, CQ 出版社, 1994.

[5] *PARTHENON Home Page*, http://www.kecl.ntt.co.jp/car/parthe/.

[6] 永野 秀尚, 須山 敬之, 名古屋 彰, 上田 義勝, 中村 行宏: PARTHENON と FPGA マッピングツールの連携に関する検討 -GateField の場合 -, 第 11 回パルテノン研究会資料集, pp. 67–76, 1997.