

## FPGA の自己動的再構成を利用したシステムの設計と開発

堀 洋平<sup>†</sup> 横山 浩之<sup>††</sup> 坂根 広史<sup>†</sup> 戸田 賢二<sup>†</sup>

<sup>†</sup>独立行政法人 産業技術総合研究所 〒305-8568 茨城県つくば市梅園 1-1-1 中央第 2 事業所  
<sup>††</sup>株式会社 KDDI 研究所 〒356-8502 埼玉県ふじみ野市大原 2-1-15  
E-mail: †{hori.y,hirofumi.sakane,k-toda}@aist.go.jp, ††yokoyama@kddilabs.jp

あらまし FPGA の自己動的部分再構成を利用したシステムの、設計方法と実装例について述べる。現在市販されている FPGA の中には、回路全体を停止することなく、回路の一部分のみを再構成可能なものがある。しかし、動的部分再構成を行う回路では様々な制約を満たす必要がある一方で、設計手法そのものが開発段階にあるため、どうすれば正しく設計できるか分かりづらい状況にある。そのため、再構成の機能を用いたアプリケーションの提案は多いにもかかわらず、実際にそれが開発された例は少ない。本稿では、ザイリンクス FPGA において自己動的部分再構成を行う回路の最新の設計方法について説明し、これを用いた DRM システムの開発例を紹介する。

キーワード FPGA, リコンフィギャラブル・コンピューティング, 動的部分再構成, コンテンツ保護, デジタル著作権保護 (DRM)

## Design and Implementation of Self Run-time Partial Reconfiguration System

Yohei HORI<sup>†</sup>, Hiroyuki YOKOYAMA<sup>††</sup>, Hirofumi SAKANE<sup>†</sup>, and Kenji TODA<sup>†</sup>

<sup>†</sup> National Institute of Advanced Industrial Science and Technology  
Tsukuba Central 2, 1-1-1 Umezono, Tsukuba-shi, Ibaraki 305-8568, Japan  
<sup>††</sup> KDDI R& D Laboratories, Inc. 2-1-15 Ohara, Fujimino-shi, Saitama 356-8502, Japan  
E-mail: †{hori.y,hirofumi.sakane,k-toda}@aist.go.jp, ††yokoyama@kddilabs.jp

**Abstract** We describe a design approach and its application of an FPGA-based system that utilizes self run-time partial reconfiguration. Some FPGAs can be reconfigured in a portion of their circuits while the other parts are working. However, the design flow of a partially reconfigurable circuit is quite complicated and misleading, and thus only few examples have been actually developed so far. This paper clarifies the detail design approach of self run-time partial reconfiguration system and demonstrates its sample implementation of Digital Rights Management (DRM) mechanisms on a Xilinx FPGA.

**Key words** FPGA, Reconfigurable Computing, Run-time Partial Reconfiguration, Content Protection, Digital Rights Management (DRM)

### 1 はじめに

再構成可能な論理デバイスである Field-Programmable Gate Array (FPGA) は、プロセス技術の進歩とともに集積度・回路規模が向上し、近年では開発用途だけでなく幅広い民生機器等に搭載されている。現在市販されている一部の FPGA は、他の部分の演算を停止させることなく回路の特定部分のみを書き換える動的部分再構成 (Dynamic Partial Reconfiguration: DPR) の機能を備えている。DPR を利用し、同一の領域において複数のモジュールを切り替えて使用することで、実装面積の削減、コストの削減、および消費電力の低減を実現しながら、様々なアプリケーションに柔軟に対応可能なハードウェアを作成することができる。

しかし、このような利点は早くから指摘されているにもかかわらず、DPR を利用したアプリケーションの実現例はまだ少ない。これは、DPR を行う回路の設計手順が非常に煩雑であることや、設計上の制約が極めて厳しいことが、回路を作成する際の大きな障害となっているためと考えられる。近年、DPR 回路の設計を支援するツールが整備され、DPR の利用に関する研究が活発化することが期待される。本稿は、DPR 回路の設計方法と応用例について説明する。

本稿では、第 2 章で DPR の概要を、第 3 章でザイリンクス FPGA における DPR の特徴を、第 4 章で DPR 回路の設計手順について述べる。第 5 章では、DPR を利用した DRM システムの開発例を紹介し、第 6 章で本稿についてまとめる。また、最新の CAD ツールを用いた DPR 回路設計の実例を付録に示す。

## 2 FPGAの動的部分再構成

### 2.1 動的部分再構成の概要

FPGAのような再構成可能論理デバイスの、全体ではなく特定部分のみを再構成することを、部分再構成 (Partial Reconfiguration: PR) という。特に、他の部分の演算を停止することなくデバイスの一部を書換えることを、動的部分再構成 (DPR) と呼ぶ。DPRは他に、Active Partial Reconfiguration や Run-time Partial Reconfiguration とも呼ばれる。また、外部回路を使用せず、デバイスが自分自身の回路を再構成することを特に、自己再構成 (Self Reconfiguration) と呼ぶ。デバイスの再構成は、以下の3つの視点から分類できるといえる。

- 部分再構成 / 全体再構成
- 動的再構成 / 静的再構成
- 自己再構成 / 外部再構成

現在、DPRの機能を有するFPGAには、ザイリンクスのSpartanシリーズとVirtexシリーズがある。この中の一部デバイスは、さらに自己動的部分再構成 (Self DPR) の機能を有する。これらのデバイスについては、第3章で詳細に説明する。

### 2.2 動的部分再構成の応用

DPR回路では、部分書換えが他の部分の演算に影響を与えてはならず、また、書換えの前後でモジュール間の信号が確実に結線される必要があるため、通常の回路にはない制約が存在する。DPR回路の設計を支援するCADツールが整備されていない頃は、設計者に多大な負担がかかっていた。2005年以降、CADツールの支援機能が改善されるようになり、比較的容易にDPR回路を作成できるようになった。そのため、DPR機能を活用した以下のようなアプリケーションの研究が活発化することが期待される。

- 実装面積を削減し、消費電力を抑えた組み込み機器の開発。
- 環境に合わせて回路構成を変更する適応型ハードウェア。
- 自己の動作の結果をフィードバックし、回路機能を改良する進化型ハードウェア。

このほか筆者らは、部分再構成モジュールと固定モジュールのインタフェースを認証に利用する、セキュリティシステムへの応用研究を行っている [1]~[4]。本稿の第5章では、Self DPRを利用したコンテンツ保護システムの例を紹介する。

## 3 ザイリンクスFPGAにおける動的部分再構成

### 3.1 Early Access PR

ザイリンクスから一般向けに公開されているDPRのドキュメントには、アプリケーションノート XAPP290 [5] や開発システムリファレンスガイド [6] 等があるが、その中で解説されている手順は古いものである。本稿ではこれを旧設計手順と呼ぶ。旧設計手順には多くの問題点があり、実際に動作する回路を作成することは極めて難しい。

DPR回路の最新の設計手順は、Early Access Partial Reconfiguration (EA PR) design flow と呼ばれている。本稿ではこれを新設計手順、あるいは単にEA PRと呼び、この手順に従って作成する回路をEA PR回路と呼ぶ。EA PRに関するドキュメント

やリファレンスデザイン等のリソースは、ザイリンクスのウェブサイト上にあるEA PR Lounge から入手可能であるが、このサイトへアクセスするにはユーザ登録をする必要がある。EA PR Lounge やユーザ登録申請ページのURLを公開することは禁止されており、アクセスを希望するユーザはザイリンクスの技術サポートに問い合わせなければならない (学生は指導教官の許可が必要)。

### 3.2 開発ツール

ER PRでは、ザイリンクスの統合開発環境ISEに、特殊なパッチを当てたものを使用する。このパッチは、前節で述べたEA PR Lounge から入手可能である。現在、ISE8.li+sp1 およびISE8.2i+sp1 用のパッチが公開されている。EA PR回路の設計では通常回路とは異なる作業ディレクトリ構造が必要となるため、ISEを用いて個々のモジュールの論理合成をすることはできるが、その後の設計手順を自動化することはできない。論理合成後、ユーザはngdbuild, MAP, PAR, bitgen等のコマンドを所定の作業ディレクトリにおいてコマンドラインから実行する。この作業は極めて煩雑である。

ザイリンクスのデザイン解析ツールPlanAhead [7]を使用すると、上述の一連のコマンドの実行が自動化されるほか、EA PR回路に特有の制約も簡単に設定できるため、設計が飛躍的に容易になる。PlanAheadは、フロアプラン、配置配線、タイミング解析、およびモジュールベースのインクリメンタルな設計等を行うためのツールであるが、EA PR回路の開発プラットフォームとして利用することが可能である [8]。ただし、初期状態のPlanAheadはEA PR回路開発の支援機能が無効になっており、これを有効にするためのTclコマンドに関する情報をEA PR Lounge から入手する必要がある。

### 3.3 DPR対応デバイス

現在市販されているFPGAの中でDPRが可能なものは、ザイリンクスのSpartanシリーズとVirtexシリーズである。第3.1節で述べたDPR回路の新旧の設計手順のうち、旧設計手順の開発が可能であるデバイスは、Virtex/-E/-IIである。リファレンスデザイン [6]の記述では、その他のVirtexシリーズやSpartanシリーズにおいても旧設計手順が適用可能とされていたが、最新の情報<sup>(注1)</sup>では適用不可能とされている。

EA PRでの開発が可能なデバイスは、Virtex-4までのVirtexシリーズと、Spartanシリーズである。Virtex-5は、近々EA PRに対応予定である。Virtex-II/-II Pro/-II Pro X/-4では、PlanAheadを利用したフロアプラン、制約の設定、および設計の自動化が可能である。Spartan-3ファミリでは、PlanAheadを利用したフロアプランや制約の設定は可能であるが、その後の配置配線やビットストリーム生成等の一連の作業を自動化することはできない。Virtex/-EやSpartan/-II/-IIIでは、PlanAheadを利用することはできない。

Virtexシリーズにおける部分再構成は、グリッチレスである。すなわち、DPRの実行中に固定モジュール内の信号の値が変化しないことを保証している。Virtexシリーズでは、再構成回

(注1): E-mailによるXilinxとの私信。2006年12月13-15日。

表1 ザイリンクスデバイスの部分再構成対応状況

Device	Old XAPP290 flow	EA PR flow	Glitchless	PlanAhead support
Virtex	○	○ (untested)	○ (untested)	×
Virtex-E	○	○ (untested)	○ (untested)	×
Virtex-II	○ (untested)	○	○	All*
Virtex-II Pro/-II ProX	×	○	○	All*
Virtex-4	×	○	○	All*
Virtex-5	×	soon to be supported	○	soon to be supported
Spartan	×	○ (untested)**	×	×
Spartan-II/-IIE	×	○**	×	×
Spartan-3/-3L/-3E	×	○	×	floorplan, constrain, export
Spartan-3A	×	○	×	×

\* All = floorplan, constrain, export, map, PAR, assemble.

\*\* 現時点ではバスマクロが提供されていない。

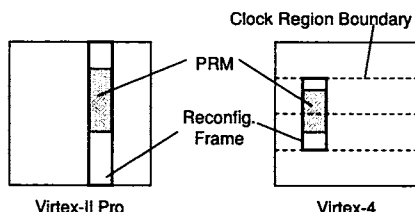


図1 部分再構成フレーム

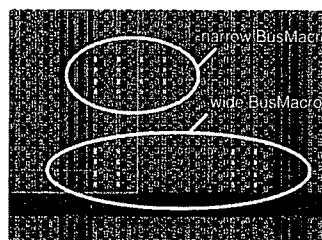


図2 スライススペースバスマクロ

路データが実装済み回路構成データと異なる場合のみコンフィギュレーションメモリが更新されるため、固定モジュール内の信号は DPR 中に周辺回路から影響を受けない。

Spartan シリーズでは部分再構成の実行中にグリッチが発生する可能性があり、グリッチが周辺モジュールの動作に影響を与えないよう設計者が注意を払う必要がある。

ザイリンクス FPGA の部分再構成対応状況について、表1にまとめる。表1は2006年12月時点の対応状況であり、今後変更される可能性がある。

### 3.4 部分再構成モジュール

EA PR 回路では、長方形のモジュール単位で部分再構成を行う。このとき、部分再構成の対象となるモジュールを **Partially Reconfigurable Module (PRM)** と呼び、そのモジュールが配置されるデバイス上の領域を **Partially Reconfigurable Region (PRR)** と呼ぶ。PRR に関する制約は、Virtex-II Pro 以前と Virtex-4 以降のデバイスで異なる。旧設計手順では、Spartan シリーズと Virtex-II Pro までの Virtex シリーズにおける PRR の大きさは4カラム単位である。この場合、PRR の高さは常にデバイスの高さと等しくなるため、複雑な回路では PRR を横切る信号の数が多くなり、設計が困難であった。

EA PR ではこの制約は取り除かれており、PRR は任意の大きさの長方形とすることができる。ただし、部分再構成フレームはカラム単位である(図1左)。Virtex-4 では、PRR は任意の大きさの長方形であり、部分再構成フレームの上下の境界はクロックリージョンの境界に等しい(図1右)。また、Virtex-4 においては、中央のカラムをまたがるような PRR を作ることはできない。なお Virtex-4 の Digital Clock Manager (DCM) と Multi Gigabit Transceiver (MGT) には、動的にパラメータを変更する

ための Dynamic Reconfigurable Port (DRP) と呼ばれるポートがあり、これを利用した再構成も DPR の一種であると言えるが、本稿では取り上げない。

### 3.5 バスマクロ

DPR では、部分再構成後の配線が正しく結線されることを保証する必要がある。ザイリンクス FPGA では、バスマクロと呼ばれるハードマクロを用いてこれを実現する。PRM と固定モジュール、あるいは2つの PRM の間の信号は、必ずバスマクロを通らなければならない(クロック等のグローバル信号を除く)。バスマクロの位置を常に一定にすることで、部分再構成後の信号が確実に結線される。XAPP290 のリファレンスデザインに付属するバスマクロは、トライステートバッファ (TBUF) を使用した配線済みのハードマクロであり、左から右、あるいは右から左のどちらか一方に4 bit 幅の信号を通過させる。このバスマクロは古く、EA PR では使用しない。

EA PR では、スライススペースのバスマクロを使用する。スライススペースバスマクロは、縦4個、横4個の合計16個の Look-up Table (LUT) から構成される配線済みハードマクロである。スライススペースバスマクロは、左から右、あるいは右から左のどちらか一方に8 bit 幅の信号を通過させる。また Virtex-4 に限り、上から下、および下から上方向のバスマクロも存在する。さらに、信号の向きの違いのほか、同期/非同期、イネーブル付き/イネーブルなし、narrow/wide の区別がある。Narrow バスマクロの横幅は2 CLB、wide は4 CLB である(図2)。Wide バスマクロを3つ並べて使用することにより、縦幅2 CLB の境界において最大24 bit 幅の信号を通過させることが可能となる。

旧設計手順では、固定モジュール間の信号であっても、PRR

表 2 ICAP と Select MAP ポートの対応 ICAP

ICAP	SelectMAP
BUSY	BUSY
O [0:7]	D [0:7] (bidirectional)
CE	CS_B
CLK	CCLK
I [0:7]	D [0:7] (bidirectional)
WRITE	RDWR_B

を横切る場合はバスマクロを使用する必要があった。EA PR では、PRR を横切る場合であっても、固定モジュール間の信号にバスマクロを使用する必要はない。

### 3.6 Internal Configuration Access Port

Spartan-3 ファミリーおよび Virtex シリーズの FPGA には、内部ロジックからコンフィギュレーションメモリにアクセスするための **Internal Configuration Access Port (ICAP)** と呼ばれるプリミティブがある。ICAP を通じて、コンフィギュレーションメモリを読み書きすることが可能である。自己再構成を行う際は、ICAP が必須となる。ICAP のポートは、データポートが入力と出力に別れている点を除いて、Select MAP モードにおけるコンフィギュレーションピンと同等の働きをし、その対応は表 2 のようになっている。なお Virtex-4 では、ICAP の入出力データ幅を 8 bit または 32 bit に設定することができる。

## 4 EA PR 回路の設計手順

EA PR 回路の設計は、以下のような手順で行われる。

- (1) 各モジュールの論理合成
- (2) 制約の設定
- (3) 固定モジュールの配置配線
- (4) PRM の配置配線
- (5) 固定モジュールと PRM のマージ

本章では、これらの手順の要点について説明する。なお、PlanAhead を用いた EA PR の実例を、付録に掲載した。

### 4.1 モジュールの論理合成

各モジュールの HDL を論理合成し、ネットリストを得る。トップモジュール、固定モジュールおよび PRM の、HDL の作成時や合成時の注意点を以下に示す。

#### トップモジュール

- 回路中のすべてのグローバルなロジック (I/O、クロック関連プリミティブ) および ICAP は、トップモジュールに置く。
- 固定モジュール、PRM、バスマクロは、トップモジュールの中でブラックボックスとしてインスタンス化する。
- PRM 間や、PRM と固定モジュール間のすべての信号が、バスマクロを経由していなければならない。

#### 固定モジュール

- クロック関連のプリミティブを置いてはならない。
- KEEP\_HIERARCHY オプションを “yes” か “soft” に設定する。
- Add I/O buffer チェックボックスを OFF にする。

#### PRM

- クロック関連のプリミティブを置いてはならない。
- KEEP\_HIERARCHY オプションを “yes” か “soft” に設定する。
- Add I/O buffer チェックボックスを OFF にする。
- 同一の PRR に実装される PRM のポート名は、すべて同じでなければならない。

### 4.2 制約の設定

タイミング制約、配置制約、モジュラーデザイン制約等の制約を設定する。トップモジュール、固定モジュールおよび PRM に関する制約や設定時の注意点について、以下で述べる。

#### トップモジュール

- すべての I/O、クロックプリミティブ、バスマクロの位置を指定しなければならない。
- バスマクロを配置する際に指定するスライスの X、Y の値は、常に偶数である。
- バスマクロは、2 モジュールの境界をまたいで実装されなければならない。

#### 固定モジュール

- AREA GROUP 制約を設定し、すべての固定モジュールを同一のグループに含める。
- 固定モジュールには、モジュールの範囲を指定する AREA GROUP RANGE 制約を設定してはならない。

#### PRM

- それぞれの PRM に対し、AREA GROUP 制約を設定する。
- それぞれの PRM に対し、モジュールが使用するスライスや BRAM の範囲を AREA GROUP RANGE 制約を用いて設定する。
- それぞれの PRM に対し、AREA GROUP MODE 制約で “RECONFIG” を指定する。

### 4.3 固定モジュールの配置配線

まず、トップモジュールのネットリストと制約ファイルを用いて、NGD ファイルを生成する。NGD ファイルは、ザイリンクスのプリミティブを用いて記述された論理的な回路デザインである。次に、この NGD ファイルと固定モジュールのネットリストを用いて固定モジュールを含む NGD ファイルを作成する。その後、マッピング、配置配線を実行する。

配置配線では、static.used というファイルが生成される。このファイルには、固定モジュールによって使用された PRR のリソースのリストが記述されている。PRM のルーティングでは、このリストにないリソースを使用する。そのため、固定モジュールに変更が加えられた場合には、PRM の配置配線も再度実行する必要がある。

### 4.4 PRM の配置配線

固定モジュールの場合と同様に、各 PRM について配置配線を行う。固定モジュールの配置配線で生成された static.used というファイルは、arcs.exclude という名前で作業フォルダにコピーする。PlanAhead ではこの作業は自動的に行われる。配置配線ツールは、この名前のファイルにあるリソースを使用しない。

#### 4.5 モジュールのマージ

固定モジュールと PRM をマージし、回路全体のコンフィギュレーションビットストリームを作成する。また、PRM のパーソナルビットストリーム、および PRR に PRM が実装されていない状態のブランクビットストリームを生成する。

### 5 自己動的再構成を利用した DRM システム

Self DPR を利用し、セキュアなコンテンツ配信システム [1]～[4] を構築する例について紹介する。本システムは、PRM と固定モジュール間の信号が正しく結線されなければ回路が正常に動作しない点に着目し、バスマクロの位置を端末に固有とすることで、特定の端末のみでコンテンツを再生可能とする仕組みを提供する。

#### 5.1 コンテンツ保護の仕組み

本システムでは、PRM をコンテンツ固有回路 (Content-Specific Circuit: CSC)、固定モジュールを端末固有回路 (Terminal Built-in Circuit: TBC) と呼ぶ。サーバからクライアント端末に CSC の回路構成データをダウンロードし、Self DPR を用いてこれを構築することで、初めてコンテンツを再生することができる。CSC と TBC が正しく結合し、全体として設計通りに動作することをインターロックするという。CSC-TBC がインターロックするには、CSC-TBC 間で以下の条件が全て成立する必要がある。

- (1) 空間的条件: 接続配線の位置が一致する。
- (2) 時間的条件: 送受信される信号のタイミングが適合する。
- (3) 電気的条件: 接続配線における電圧や電流等の条件が適合する。
- (4) 論理的条件: プロトコルが一致し、送受信されるデータが適合する。

以上の条件を CSC-TBC 間の組合せに固有とすることで、インターロックを認証の手段として利用することができる。

#### 5.2 システムの実装

CSC-TBC がインターロックするメカニズムを、ハイビジョン動画の再生システムにおけるデジタル著作権管理 (Digital Rights Management: DRM) に応じた例を示す。回路による高速な画像処理を行いつつ、FPGA の Self DPR を利用することで、コンテンツに応じた再生機能のカスタマイズとライセンスの管理を行うことができる。

##### 5.2.1 システムのハードウェア構成

DRM システムを、レクセオン・テクノロジー社<sup>(注2)</sup> の REX2 (REconfigurable EXperimental equipment 2) を用いて構築した。REX2 は、Virtex-II Pro XC2VP70 を 1 個搭載した FPGA 開発ボードである。REX2 を用いたシステムのハードウェア構成を図 3 に示す。システムは、CSC-TBC や動画再生回路が実装される主ボード (REX2 MAIN)、PC から映像ソースを供給するための PCI-X ボード (REX2 PCI-X)、および DVI 入力ボード (REX2 DVI) から構成される。

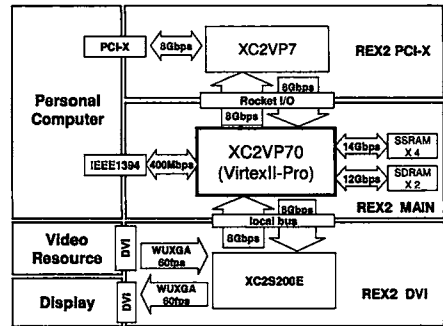


図 3 システムのボード構成

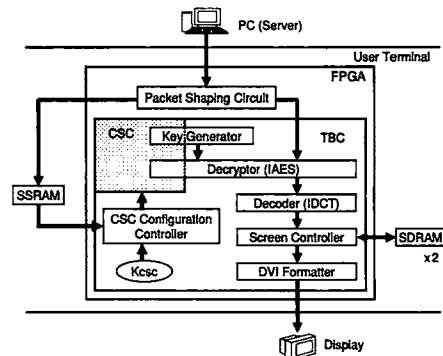


図 4 システムの内部構成

#### 5.2.2 システムの内部構成

図 4 にシステムの内部構成を示す。AES (CBC mode, 128-bit block size, 128-bit key) によって暗号化された動画像コンテンツは、CSC-TBC の結合回路によって生成された復号鍵を用いて TBC 内の復号回路で復号され、映像処理回路を経由してビデオ映像として外部モニタに出力される。鍵の生成回路と復号回路が CSC と TBC にまたがって実装されており、CSC と TBC が正しく結合しなければ動画を正常に再生することができない。

#### 5.2.3 システムの検証実験

CSC-TBC によるコンテンツ保護のメカニズムを FPGA 上に実装し、実際に部分再構成を行って動作を検証した。CSC-TBC の保護メカニズムは、暗号化された 1080p ハイビジョン (1920 × 1080 ピクセルのプロGRESSIVEモード) の動画像が正しく再生できるかどうかで検証した。CSC が以下の状態である場合において、それぞれ動画の再生実験を行った。

- A) CSC が構築されていない (システム初期状態)。
- B) コンテンツに対応した CSC が、正規の端末上に構築される (インターロック成功)。
- C) コンテンツに対応した CSC が、不正な端末上に構築される。
- D) コンテンツに対応していない CSC が、正規の端末上に構築される。
- E) コンテンツに対応していない CSC が、不正な端末上に構築される。

(注2) : レクセオン・テクノロジー (株) は、(独) 産業技術総合研究所の技術移転により設立されたベンチャー企業である。http://www.rexeon.com/

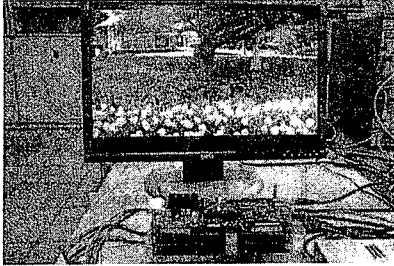


図5 動画再生実験の結果 (インターロック成功時)

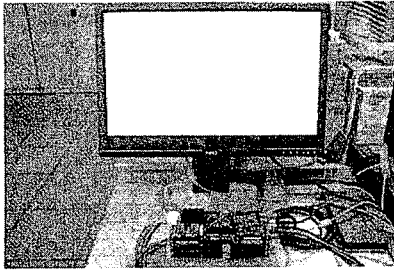


図6 動画再生実験の結果 (インターロック失敗時)

なお、正規の端末とは、認証されたユーザが所有する、コンテンツが再生されるべき端末である。不正な端末とは、コンテンツが再生されてはならない端末であり、攻撃者によって作成された端末のほか、他のユーザが所有する端末も含まれる。

実験の結果、B)のインターロック成功時には、図5のように動画が正しく生成されたが、A), C), D), E)のインターロック失敗時には、図6のように画面が砂嵐状となって動画は正しく再生されなかった。この結果、Self DPRを利用したCBC-TBCメカニズムにより、コンテンツ保護機能を提供可能であることが示された。

## 6 おわりに

本稿では、ザイリンクス FPGA の自己動的再構成 (Self Dynamic Partial Reconfiguration: Self DPR) を利用した回路の設計方法と、応用例について述べた。DPR 回路には通常回路と異なる制約が多く、従来の開発ツールのみによる設計は困難であった。しかし、PlanAhead のように DPR 回路の設計を支援するツールが整備されてきたことによって、煩雑であった設計は飛躍的に改善された。今後、DPR の応用研究が活発化することが期待される。

Self DPR の応用例として、モジュールのインターロックを認証として利用し、コンテンツの不正利用を防止するデジタル著作権管理 (DRM) システムを実装した。このシステムでは、再生回路の一部をコンテンツと共にダウンロードし、Self DPR を用いてこれを構築することでコンテンツの再生が可能となる。Self DPR を利用することで、ハードウェアによる高速な処理を実現しながら、コンテンツに応じた柔軟な DRM 機能を提供すると共に、攻撃手法の進歩に合わせて、DRM の回路構成を更新することが可能である。

## 文 献

- [1] 横山浩之, 戸田賢二, “FPGA を用いたコンテンツ保護システムの開発,” 信学技報 CPSY2004-114, pp.55-60, 2004.
- [2] H. Yokoyama, and K. Toda, “FPGA-based content protection system for embedded consumer electronics,” RTCSA, pp.502-7, 2005.
- [3] Y. Hori, H. Yokoyama, and K. Toda, “Secure content distributing system based on run-time partial reconfiguration,” FPL, pp.637-640, 2006.
- [4] 横山浩之, 畑洋平, 戸田賢二, “FPGA の部分書換方式を用いたコンテンツ保護システムの検討,” 信学技報 RECONF2006-34, 2006.
- [5] Xilinx, “Two flows for partial reconfiguration: Module based or difference based,” , Sept. 9, 2004 2004.
- [6] Xilinx, Development System Reference Guide, , for ISE8.1i edition, 2005.
- [7] Xilinx, PlanAhead User Guide, Release 8.2, , 2006.
- [8] N. Dorairaj, E. Shiflet, and M. Goosman, “PlanAhead Software as a platform for partial reconfiguration,” Xcell Journal, vol.55, pp.68-71, 2005.

## 付 録

### 1 PlanAhead を用いた DPR 回路の設計例

ここでは、PlanAhead を用いた DPR 回路設計の実例を紹介する。PlanAhead での設計を始める前に、トップモジュール、固定モジュール、PRM のすべての論理合成が終わっている必要がある。また、タイミング制約や、I/O, DCM, バスマクロ等の配置制約を記述した UCF ファイルをあらかじめ用意する。

今回設計に使用したツールは、EA PR バッチをあてた ISE8.1i+sp1 と PlanAhead 8.2.4 であり、設計時のフォルダ構成は図 A-1 のようになっている。回路中の PRR は 1 つであり、PRR に実装される 2 つの PRM を PRM\_1, PRM\_2 とする。

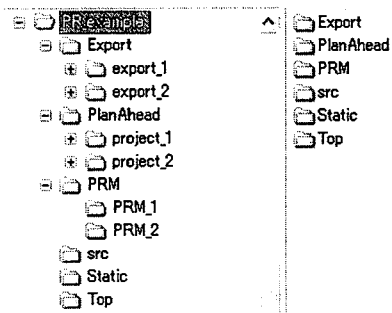


図 A-1 DPR 回路設計時のフォルダ構造の例

### 1.1 フロアプランの作成

(1) 新規プロジェクトを作成する。ここでは、プロジェクトの場所を PR\_example/PlanAhead とし、名前を project\_1 とする。次に、ネットリストとして、トップモジュールである PR\_example/Top/top.ngc を指定する。さらにサーチパスとして、固定モジュールおよび PRM のネットリストが含まれるフォルダ (Static および PRM/PRM\_1) を指定する (図 A-2)。以下、指示に従って、使用するデバイス、制約を記述した UCF ファイルを指定する。

(2) Netlist ツリーで PRM 以外のすべてのモジュールを選択した状態で、Tools メニュー、または右クリックメニューから

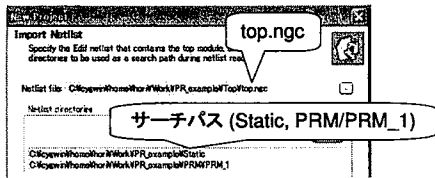


図 A-2 新規プロジェクトの作成とネットリストのインポート

“New Pblock” を実行し、Pblock の名前を指定する。ここでは “AG.static” とする。Physical Hierarchy ツリーに、AG.static が追加される (図 A-3)。これにより、フロアプランのエクスポート時に AREA GROUP 属性が設定された UCF ファイルが生成される。なお、ここで Pblock の範囲を指定してはならない。

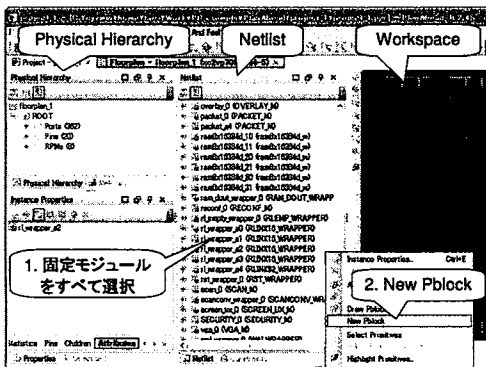


図 A-3 固定モジュールのセットアップ

(3) Netlist ツリーで PRM を選択した状態で、右クリックメニューから “Draw Pblock” を実行し、Workspace 上で PRM の範囲を指定する (図 A-4)。Workspace 上で右クリックし、“Select→Clock Region” とすると、クロックリージョンの境界が表示されるので、PRM の上下の境界をそれに合わせるとよい。範囲を指定後、Pblock の名前を指定する。ここでは “AG.PRIM.1” とする。Physical Hierarchy ツリーに AG.PRIM.1 が追加される。これにより、AREA GROUP RANGE 制約が設定される。

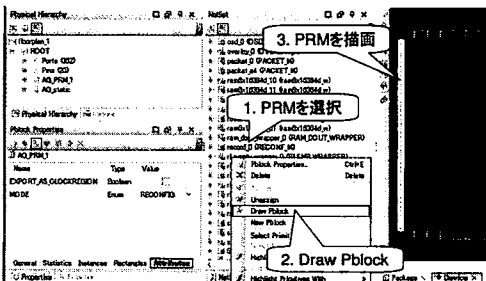


図 A-4 PRM の範囲の設定

(4) Physical Hierarchy ツリーで AG.PRIM.1 を選択し、Pblock Properties ウィンドウの Attribute タブで属性を設定する。“Define new attribute” アイコンを押して “MODE” を選択し、

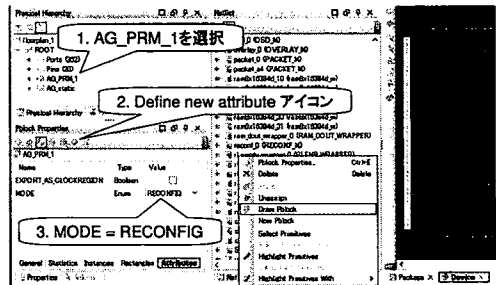


図 A-5 PRM のモードの設定

プルダウンメニューから “RECONFIG” を設定する (図 A-5)。これにより、AREA GROUP MODE 制約で RECONFIG が設定される。

(5) Netlist ツリーの Primitives フォルダの中のバスマクロを選択し、Workspace 上でバスマクロが正しく配置されていることを確認する。PRM がモジュール境界をまたいでいない場合は、ドラッグして適切に配置する。

(6) ウィンドウの最下段のコマンドプロンプトから、PR 設計の支援機能を有効にするための Tcl コマンドを入力する。または、メニューの “Tools→Run Tcl Script” からスクリプトファイルを指定する。これで、PR 回路のデザインルールチェック (DRC)、PR モードでのフロアプランのエクスポート、PR フローウィザードを利用することができるようになる。

(7) “Tools→Run DRC” を実行する。チェックされる項目は、図 A-6 のようになっている。

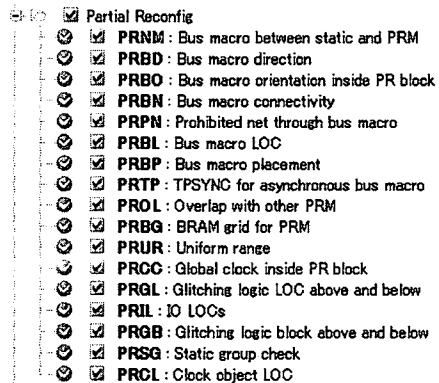


図 A-6 DPR 回路のデザインルールチェック項目

(8) “Files→Export Floorplan” を実行する。出力先のフォルダは、ここでは PR.example/Export/export.1 とする。Export mode で “Partial Reconfig” が選択されていることを確認し、Next を押す。Summary 画面 (図 A-7) で、export floorplan mode や出力フォルダが正しいことを確認して Finish を押す。

(9) 出力先フォルダ export.1 と、export.1/static フォルダの中に、バスマクロの NMC ファイルを置く。PlanAhead のバージョンによってはバスマクロもエクスポートされるので、その場合はこの作業は必要ない。

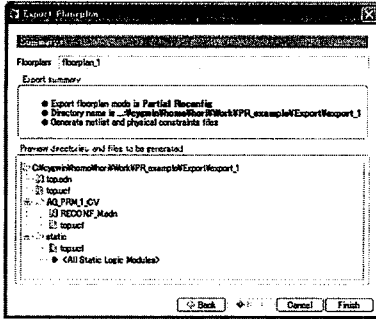


図 A-7 フロアプランのエクスポート

## 1.2 PRM.1 を含む回路のコンパイル

(1) “Tools→ Run Partial Reconfig” を実行すると、PR フローウィザードが起動する。PR フローは、

- i) Budgeting (トップモジュールの NDG ファイル作成)
- ii) Static Logic Implementation (固定モジュールの PAR)
- iii) PR Block Implementation (PRM の PAR)
- iv) Assemble (ビットファイルの生成)

の順に実行される。PR Block Implementation を選択し、PRM が正しくリストに現れているか確認する(図 A-8)。初期状態では PRM が正しく選択されていない場合があるので注意する。

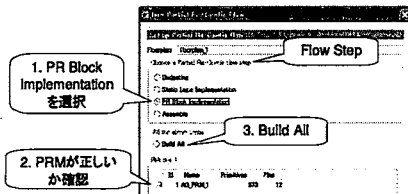


図 A-8 PR flow ウィザード

(2) Next を 3 回押し、“Flip-flops or latches may be packed into” という項目で “None” をチェックする。これは、PRM のポートが誤って入出力バッファに割り当てられることを防ぐためである。

(3) Back で flow step 画面に戻る。各ステップを 1 つずつ実行していくこともできるが、ここでは “Build All” をチェックする(図 A-8)。Next を押しと Summary 画面が現れ、Finish を押しと ISE が呼び出され、全ステップの処理が順に実行される。

(4) すべてのステップが終わると、export\_1/merge フォルダの中に、以下のビットファイルが生成される。

- static\_full.bit (PRM.1 を含む回路全体のデータ)
- ag\_prm\_1\_cv\_routed\_partial.bit (PRM のみ)
- ag\_prm\_1.blank.bit (ブランクモジュール)

## 1.3 PRM.2 を含む回路のコンパイル

(1) “File→ Save Project As” を実行し、プロジェクト名を変更して保存する。ここではプロジェクトの場所は PR.example/Project とし、名前を project\_2 と変更する。

(2) “File→ Update Netlist” を実行する。“Replace a specific module” が選択されていることを確認して Next を押し、PRM.2

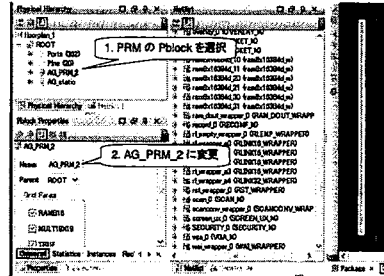


図 A-9 PRM.2 のセットアップ

のネットリストを指定する。

(3) Physical Hierarchy ツリーで PRM の Pblock を選択し、Pblock Properties ウィンドウの General タブで、名前を変更する(図 A-9)。この作業は必須ではないが、どの PRM に対する作業かを明確にすることができる。ここでは、AG\_PRM.2 とする。

(4) “Tools→ Run DRC” を実行する。エラーが出たら、指示に従って修正する。

PRM.1 の場合と同様に、エクスポートして “Build All” をすることもできるが、flow step の最初の 2 つは PRM.1 と同じである。ここでは以下のようにし、PRM.2 については 3 つ目の flow step (PB Block Implementation) から始める。

(5) export\_1 フォルダにある、トップモジュールの NGD ファイル、エクスポートされた固定モジュール、バスマクロ(図 A-10)を、PR.example/Export/export\_2 にコピーする。

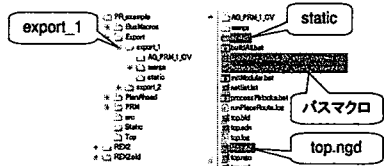


図 A-10 トップ、固定モジュール、バスマクロのコピー。

(6) “Files→ Export Floorplan” を実行する。出力先のフォルダは、ここでは PR.example/Export/export\_2 とする。Export mode で “Partial Reconfig” が選択されていることを確認し、Next を押し、さらに Finish を押す。

(7) “Tools→ Run Partial Reconfig” を実行する。Flow step で “PR Block Implementation” を選択し、PRM が正しくリストに現れていることを確認する(ここでは AG\_PRM.2)。

(8) Next を押し、“Flip-flops or latches may be packed into” という項目で “None” をチェックして Finish を押し、PRM の配置配線を実行する。

(9) Flow step で “Assemble” を選択して Next を押し。Summary 画面を確認して Finish を押し、ビットファイルの生成を行う。処理が成功すると、export\_2/merge フォルダの中に以下のビットファイルが出力される。

- static\_full.bit (PRM.2 を含む回路全体のデータ)
- ag\_prm\_2\_cv\_routed\_partial.bit (PRM のみ)
- ag\_prm\_2.blank.bit (ブランクモジュール)