

FPGA 遠隔再構成とリモート・ロジックアナライザ

永田 和生[†] 田代 輝[†] 身次 茂[†]
柴村 英智^{††} 久我 守弘^{††} 末吉 敏則^{††}

[†] 熊本大学 大学院 自然科学研究科

^{††} 熊本大学 工学部

〒 860-8555 熊本市黒髪 2 丁目 39 番 1 号

E-mail: †{nagata,kagayaki,mitsugi}@arch.cs.kumamoto-u.ac.jp,

††{sibamura,kuga,sueyoshi}@cs.kumamoto-u.ac.jp

あらまし 再構成可能な FPGA デバイスを搭載した機器は、製品出荷後もハードウェア機能の変更・拡張・アップグレードを行うことができ、不具合が見つかった場合でもデバッグできるという長所がある。しかし、遠隔地にある FPGA の再構成を行う際には、技術者や再構成に必要な機材を遠隔地に派遣しなければならず、費用や時間的な面で問題がある。そこで、我々はインターネットに接続されている機器に搭載された FPGA を遠隔操作により再構成する環境を構築した。またその回路が正しく動作していることを検証するリモート・ロジックアナライザを開発した。これにより、技術者の派遣が困難な場所にあるデバイスも、手元の開発環境からデバッグを行うことが可能となった。
キーワード FPGA, 遠隔再構成, ISP(In System Programming), ロジックアナライザ

FPGA Remote-Reconfiguration and Remote Logic Analyzer

Kazuo NAGATA[†], Kagayaki TASHIRO[†], Sigeru MITSUGI[†],

Hidetomo SHIBAMURA^{††}, Morihiko KUGA^{††}, and Toshinori SUEYOSHI^{††}

[†] Graduate School of Science and Technology, Kumamoto University

^{††} The Faculty of Engineering, Kumamoto University

2-39-1, Kurokami, Kumamoto City, Kumamoto, 860-8555, Japan

E-mail: †{nagata,kagayaki,mitsugi}@arch.cs.kumamoto-u.ac.jp,

††{sibamura,kuga,sueyoshi}@cs.kumamoto-u.ac.jp

Abstract The equipments carrying a reconfigurable FPGA have advantages, the functions of which can be changed, extended or upgraded after shipping. When we reconfigure the FPGA in a remote place, however, an engineer and equipments for reconfiguration must be dispatched to a remote place. and there is require cost and time. Then, we developed an environment which reconfigures FPGA carried in the equipment and a remote logic analyzer which verifies reconfigured circuits in an FPGA correctly with remote operation via Internet. Thereby, we get possible to debug a device in the place where dispatch of an engineer was difficult from local development environment.

Key words FPGA, Remote reconfiguration, ISP (In System Programming), Logic analyzer

1. はじめに

再構成可能な FPGA (Field Programmable Gate Array) デバイスを搭載した機器は ISP (In System Programming) 機能を利用して、製品出荷後もハードウェアの機能変更・機能拡張・アップグレードを行うことができ、また実装回路に不具合が

見つかった場合でもデバッグを行うことができるという長所がある。

近年、遠隔操作による FPGA デバイス再構成の要求が高まっている。その理由として、遠隔地への技術者派遣の困難性、コスト的な問題、構成時間増大の問題などが挙げられる。ネットワークを介した遠隔操作によるデバイスの再構成が可能になれ

ば、これらの問題を大きく削減することができる。しかし、デバイスの再構成には様々な手法があり、また、再構成を行う環境もアプリケーションの開発側とリモート側で異なる可能性がある。そのため、統一された環境での再構成を行うことができず、各々の手法に合わせてコンフィギュレーションプロセス全体を設計する必要があり無駄が多い。

本研究では比較的統一されたコンフィギュレーションを行える JTAG による ISP と、プログラムの実行がプラットフォームに依存しないという特徴をもつ Java の実行環境を利用した FPGA の再構成環境の開発を行ってきた [1]。また、再構成を行った FPGA に対して、従来は物理的に測定器のプロープを当てることで実際の動作の確認を行っていたが、遠隔操作による再構成を行った FPGA に対しては、動作確認もまた遠隔操作によって行う必要がある。そこで、プロープを物理的に当てる必要がなく、遠隔操作によって制御可能なリモート・ロジックアナライザの開発を行った。

以下、2. 章では FPGA 遠隔再構成技術について、3. 章ではリモート・ロジックアナライザについて述べる。4. 章で評価実験の結果について考察を行い、5. 章でまとめとする。

2. FPGA 遠隔再構成技術

2.1 想定する動作環境

遠隔操作による FPGA の再構成が応用される場面として、以下のようなものを想定する。

- 人が立ち入ることのできない場所での再構成
- 人の派遣に高いコストのかかる場所での再構成
- 多くの機器を対象とした再構成

以上のような場面を想定した場合、遠隔操作によって FPGA を再構成するにあたり、手元の開発環境とデバイスのあるリモート先の環境で、以下のような差異が問題となる。

- 構成データのダウンロード手法
- オペレーティングシステム (OS)
- デバイスの構成

このような状況では、ある 1 つの手法で画一的にデバイス再構成を行うのは困難である。本研究では比較的統一された手法でコンフィギュレーションを行うことが可能な IEEE std. 1149.1(JTAG) の ISP と、プログラムの実行がプラットフォームに依存しないという特徴をもつ Java 言語を利用して、FPGA の遠隔再構成技術を開発した。

2.2 IEEE std. 1149.1 (JTAG)

JTAG とはデバイス内部に対して「バウンダリスキャン」によって動作テストを行う規格であり、JTAG を利用した ISP が可能なデバイスが存在する。JTAG の ISP 機能を用いることによって、通常のコングフィギュレーション手法と比較して、統一された手続きでデバイスのコンフィギュレーションを行うことが可能である。この JTAG-ISP の特徴に着目し、本研究では、JTAG によるコンフィギュレーションを中心に研究を行った。

2.3 Java

Java 言語は現在広く普及し、異なるプラットフォームであっても、1 度コンパイルしたプログラムをそのまま他のプラット

フォームでも実行できるという “Write Once Run Anywhere” という特徴を持っている。また、組込み用の実行環境 (J2ME) が用意されている。そこで、再構成を行うプログラムと、それを実行する環境の構築に Java 言語およびその実行環境を用いることで、多くのプラットフォームで実行可能となり、またプログラムの変更がほとんど必要なくなるメリットがある。

2.4 JNI (Java Native Interface)

FPGA の再構成を行うには、プログラムからハードウェアを制御し、JTAG ビンへと信号を出力する必要がある。しかし、Java プログラムでは直接ハードウェアを制御することができない。そのため、Java 言語には JNI という API が用意されている。JNI を利用することで、C 言語などの他の言語で開発されたネイティブコードを Java プログラムからメソッドとして呼び出し、実行することができる。

2.5 遠隔再構成環境

文献 [1] の研究を元に、さらに自由度の高い遠隔再構成環境とするため、以下のような機能が必要となる。

- ローカルマシンからリモートサーバへの構成データの転送および記憶装置への格納
- リモートサーバの記憶装置内の構成データ管理
- 構成データを選択可能なデバイス再構成
- デバイス再構成の実行

これらの機能を持たせる形で開発した遠隔再構成環境を図 1 に示す。

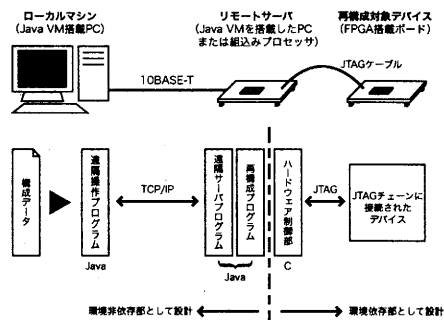


図 1 遠隔再構成環境

2.5.1 遠隔操作プログラム

遠隔サーバプログラムと通信するためのユーザインタフェースとなるプログラムである。ファイル転送画面では、ローカルマシンとリモートマシンのディレクトリ一覧を表示し、転送操作を行うことができる。また、コマンドプロンプト画面では再構成コマンドの送信や結果の表示など、シェルのような機能を提供する。すべて Java 言語で記述しており、プラットフォームに依存することなく動作可能である。

ユーザは、デバイスの構成データ (Altera 社デバイスの場合 RBF 形式) をネットワークを介して送信する。その後、送信した構成データを指定し、再構成用プログラムを実行することでデバイスの再構成を行う。図 3 に操作画面を示す。

2.5.2 遠隔サーバプログラム

リモートサーバ上の遠隔サーバプログラムには、遠隔操作プログラムからのコマンドの受信・応答機能、ローカルリモート間のファイル転送機能、リモートサーバに接続された記憶装置内のディレクトリ操作機能、接続された FPGA の再構成機能が必要となる。以上の機能を図 2 のような構造で実装した。コマンドを送信することでリモートサーバ上の操作を行うことができる、ターミナルのような環境を提供する。

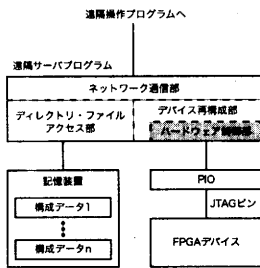


図 2 遠隔サーバプログラムの構造

ネットワーク通信部：ソケットを利用してコマンドの受信・応答、ファイルの転送、標準出力のリモート表示を遠隔操作プログラムとの間で行う。

ディレクトリ・ファイルアクセス部：記憶装置に対して一般的な UNIX 系のディレクトリ・ファイル操作コマンドと同様の操作を行うことができる。また、遠隔操作プログラムから転送されたファイルを記憶装置に格納する。

デバイス再構成部：FPGA に対して JTAG コンフィギュレーションによる再構成を行う。まず接続された JTAG チェーンからデバイスの ID コードを取得する。実行する際にデバイスのチェーン番号、デバイス名、構成データ名を引数として渡す。渡された引数に従って、デバイスに対応したプログラミング手法で記憶装置に格納された構成データによるデバイスの再構成を、パラレル入出力経由で行う。

この遠隔サーバプログラムを Java 言語で記述することで、実行環境の差異に応じて書き換えることなく動作が可能となる。

この際、環境に依存するハードウェア制御部は環境依存部として分離して実装した（図 2 のグレー部）。このように環境依存部を分離して設計することで、Java 言語への移行が容易となるとともに、異なる環境のデバイスを再構成する場合でも環境依存部のみを作成すればよい。

環境依存部は C 言語により実装し、Java プログラムから JNI を介してハードウェアの制御を行う。また、環境依存部はあくまでインタフェースであること、環境の再構築のときに大きな変更を加えなくてもよいこと、という観点からシンプルなものになるように設計した。

2.5.3 動作実証

動作実証には、ローカルマシンとして Windows 2000 を搭載し、Java 仮想マシンが動作する PC を用いた。リモートサーバとしては、将来的には対象デバイス内での部分再構成が可

能になることも念頭に置き、組み込みプロセッサの実装が可能な FPGA ボードとして、Altera 社製 Excalibur Nios 評価ボード（デバイス：Altera APEX EP20K200EFC484-2X）を用いた。評価ボードには Nios Ethernet Development Kit を装着し、ネットワークを使用可能としている。

Nios 上で動作する Java 仮想マシンとして KVM (K Virtual Machine) [3] があるが、KVM にはネットワーク機能が実装されていない。また、デバイスやボードの環境に依存するハードウェア制御部については Java プログラムから直接操作できない。そこで、今回は Java 言語で記述する前段階として、遠隔サーバプログラムを C 言語で記述している。

以上のような機器で遠隔再構成環境を構成し、エスケューエレクトロニクス社製 CM52A ボード（デバイス：Altera APEX EP20KE1000EFC672-1X）を対象デバイスとして再構成を行ったところ、正常に再構成できることを確認した。また、CQ 出版社 Stratix 評価キット（デバイス：Altera Stratix EP1S10F780C7）を対象として再構成を行ったところ、同様に正常に再構成できることを確認した。

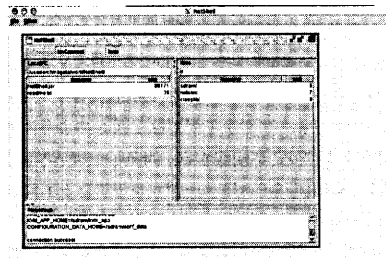


図 3 遠隔操作プログラムのユーザインタフェース

3. リモート・ロジックアナライザ

3.1 開発動機

2 章の遠隔再構成の技術により、ネットワークに接続された機器に搭載された FPGA デバイスの遠隔再構成を行うことが可能となった。しかしながら、FPGA 上に正しく回路が実装されているかを確認する方法が確立されておらず、確認自体は技術者が直接行う必要があった。そこで、遠隔操作による再構成を行った後、ネットワークを介して FPGA の動作の確認を行う方策として、リモート・ロジックアナライザの開発を行った。リモート・ロジックアナライザを使用すれば、図 4 のように遠隔地にある機器に搭載された FPGA デバイス内部の信号を、ネットワークを介して手元のプログラムから観測し、確認とともにデバッグを行うことが可能である。



図 4 リモート・ロジックアナライザ

3.2 組み込みロジックアナライザ

リモート・ロジックアナライザの開発には、FPGA 内部にユーザロジックとともに組み込むことで回路動作の確認を行う組み込みロジックアナライザの技術を活用している。組み込みロジックアナライザを使用する利点を以下に述べる。

- 物理的なプローブを必要としない
- 観測対象信号を外部ピンに割り当てる必要がない
- 使用するピンを固定化できる

組み込みロジックアナライザを任意のプログラムから使用する場合、ロジックアナライザコアのインタフェースが重要となる。現在、各 FPGA ベンダが提供している組み込みロジックアナライザコアは、同ベンダの波形観測プログラムと密接に関連している。またインタフェースが公開されていないため、ベンダ純正以外の波形観測プログラムから使用することはできない。

今回は、ロジック解析機能を的を絞った波形観測プログラムであることから、共著者の一人が代表を務めるエムビーウェア社との共同研究として同社の“ポケットロジアナ”をリモート・ロジックアナライザの研究に使用した。同社提供の組み込みロジックアナライザ [5] は Altera 社のデバイスを対象として AHDL (Altera HDL) で記述されているが、FPGA の製品シリーズやベンダの枠を越えた幅広いデバイスに対応するために、同等の機能を持ったロジックアナライザを Verilog-HDL で記述した。

リモート・ロジックアナライザでは、図 5 のように対象デバイスを“本体ロジアナ”に接続する。開発した組み込みロジックアナライザは、対象デバイスにユーザロジックとともに実装される。図 6 にそのブロック図を示す。制御部はホストからのコマンドに従って、トリガ条件の設定、サンプリングの開始、ホストへのサンプリングデータの転送を行う。トリガ部はトリガ条件の保持とトリガの検出を行う。FIFO 部はサンプリングデータを格納する。

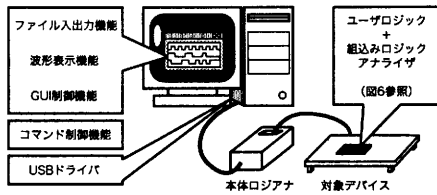


図 5 ポケットロジアナ構成

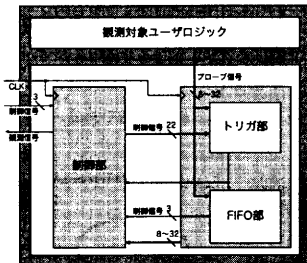


図 6 組み込みロジックアナライザの構成

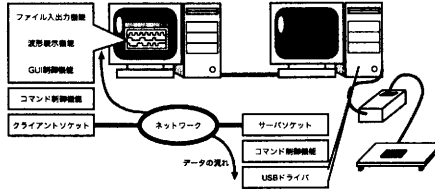


図 7 リモート・ロジックアナライザ構成

3.3 波形観測プログラム

図 5 は、ポケットロジアナで提供されているスタンドアロン版波形観測プログラムの機能を分類した図で、ロジックアナライザの機能と制御に関して重要な機能のみを示している。

ファイル入出力機能： テストパターンジェネレータで生成する波形を読み込ませたり、サンプリングした波形のデータのログを取ることができる。

波形表示機能： 測定対象回路内の指定した信号を、波形として表示する。

GUI 制御機能： プログラムの使用をより簡単にする。

コマンド制御機能： ユーザからの GUI 操作などによるイベントの発生により、組み込みロジックアナライザに対してコマンドの送信を行う。逆に、組み込みロジックアナライザから内部状態やサンプリングしたデータを抽出するためにも使用される。

USB ドライバ部： 本体ロジアナと PC 間が USB で接続されており、アプリケーションからの信号制御に使用する。

これらの機能をそれぞれ、遠隔側と操作側に分割してネットワーク対応ロジックアナライザの作成を行った。図 7 に示すように、組み込みロジックアナライザに対して制御用のコマンド送信やデータ受信を行っていたコマンド制御機能部分を 2 つに分割し、それぞれリモート側とローカル側に持たせた。分割後のコマンド制御機能部を切口として、リモート側にはサーバソケット、ローカル側にはクライアントソケットの機能を追加してネットワークを介して接続する。

図 8 に、8 ビットバイナリカウンタ回路を例に、今回開発したリモート・ロジックアナライザの動作画面を示す。対象回路の波形が各チャンネルごとに表示され、値の変化を確認することができる。ユーザ側の画面では、波形表示やテスト波形の設定など、スタンドアロン版波形観測プログラムの機能を全て操作することができる。

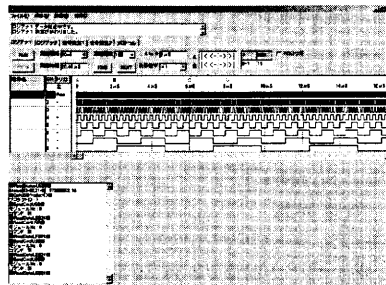


図 8 リモート・ロジックアナライザ ユーザ側画面

4. 評価

4.1 遠隔再構成環境

図1のような遠隔再構成環境を表1に示す機器を用いて構築した。その有効性を実証するため、再構成の所要時間についての評価を行った。評価の際には、ベンダ純正再構成プログラムである Altera 社製 Jrunner [4] との比較を行った。

表1 再構成に用いた PC, デバイス, 構成データ

ローカルマシン	Windows 2000 Professional CPU: Xeon 2.2GHz Dual Memory: 2GB
リモートサーバ	Excalibur Nios ボード 動作周波数: 33MHz
対象デバイス 1	Altera EP1S10F780C7
構成データ	RBF ファイル, 444.0 KB
対象デバイス 2	Altera EP20KE1000EFC672-1X
構成データ	RBF ファイル, 1,020.0 KB

表2 再構成時間の評価結果

再構成プログラム	ベンダ純正	開発したプログラム
対象デバイス 1	53,990ms	53,991ms
対象デバイス 2	136,143ms	136,143ms

再構成時間の評価では、表2のような結果が得られ、ベンダ純正の再構成プログラムとほぼ同等の時間で構成を完了することができることが実証できた。現在はC言語を用いてデバイス再構成部を記述しているため、今後はJava言語へと移行した場合にも同様の構成時間を実現できるかを調査する必要がある。

4.2 リモート・ロジックアナライザ

4.2.1 評価条件

組込みロジックアナライザ回路をユーザロジックに接続した場合、ユーザロジックに与える影響を明らかにするために評価を行った。

開発した組込みロジックアナライザ回路を、表3に示す環境において各ユーザロジックに接続した。8bit 全加算器回路では各加算器の出力8bitを、3DES回路では出力32bit中の下位から8bitずつを測定チャンネル8bitにそれぞれ割り当てた。パラメータとして、

- 測定チャンネル数 (8, 16, 32ch)
- 測定サンプル数 (512, 1,024, 2,048 samples)

を変化させて回路を EPXA10F1020C2 用に論理合成・配置配線した場合の、ロジックエレメント (LE) 数、メモリ使用量、動作速度、使用するピン数、回路合成時間について測定した。

表3 リモート・ロジックアナライザの評価

対象デバイス	EPXA10F1020C2
対象ユーザロジック 1	8bit 全加算器回路 × 4
対象ユーザロジック 2	3DES 暗号化回路
設計環境	Windows 2000 Server CPU: Xeon 2.8GHz Dual Memory: 1GB
設計ツール	Altera Quartus II 2.2

4.2.2 評価結果

(1) ロジックエレメント数への影響

表4に、ユーザロジックにリモート・ロジックアナライザを接続し、FPGAへ実装した際のLE使用数を示す。表中のNoneはロジックアナライザ回路を使用しないユーザロジックのみの結果を示す。下段()内はユーザロジックのみの場合と比較した増加量を示す。この結果から、ユーザロジックのサイズに関わらず、チャンネル数とサンプル数に対応したほぼ一定のLE数増加のみでロジックアナライザを追加可能なことがわかる。

表4 ロジックエレメント数

samples \ CHs		CHs			
		None	8ch	16ch	32ch
FA	512	52	552 (+500)	634 (+582)	865 (+813)
	1,024		562 (+510)	667 (+615)	884 (+832)
	2,048		549 (+497)	680 (+628)	897 (+845)
3DES	512	3,267	3,767 (+500)	3,849 (+582)	4,080 (+813)
	1,024		3,778 (+511)	3,882 (+615)	4,112 (+845)
	2,048		3,777 (+510)	3,895 (+628)	4,121 (+854)

(2) メモリ使用量への影響

ユーザロジックにリモート・ロジックアナライザを接続し、FPGAへ実装した際のメモリ使用量を表5に示す。ユーザロジックに用いた全加算器回路と3DES暗号化回路は、ともにESB (Embedded System Block) を使用していないため、どちらも同じメモリ使用量となり、単純にサンプリングしたデータを蓄積するために必要なチャンネル数にサンプル数を乗じたメモリビット数となることが確認できた。

ESBを用いたユーザロジックでは、回路内で用いるメモリビットとロジックアナライザで用いるメモリビットの和が、回路全体のメモリ使用量となることが予測される。

表5 メモリ使用量 (単位:bit)

CHs samples	None	8ch	16ch	32ch
512	0	4,096	8,192	16,384
1,024		8,192	16,384	32,768
2,048		16,384	32,768	65,536

(3) 動作速度への影響

表6, 表7, 図9に、ユーザロジックにリモート・ロジックアナライザを接続した際の動作速度の変化を示す。全加算器は組み合わせ回路であるため、周波数ではなくクリティカルパスにおける遅延時間を用いた。全加算器回路では最大60%の遅延増加が見られた。3DES回路では最大14.33%の速度低下にとどまった。最大約6%の速度向上も見られ、影響の大きさに規特性が見られない。この点についてはさらなる調査が必要である。

全加算器に対する評価結果から、回路規模の小さいユーザロジックへの影響は少なくない。実際にリモート・ロジックアナ

ライザを用いて回路の動作確認を行う場合は、ユーザロジック単体の場合とは動作速度が異なることを念頭に置く必要がある。

また、今回用いたユーザロジックは回路規模が小さいため、FPGA の論理資源をあまり使用していない。ユーザロジックが FPGA の論理資源を多く使用する場合は、実装結果によって動作速度が大きく左右される恐れもある。

表 6 最大回路遅延 (全加算器) (単位:ns)

CHs samples	None	8ch	16ch	32ch
512	18.96	25.745 (+35.79%)	28.906 (+52.46%)	29.236 (+54.20%)
		27.461 (+44.84%)	26.182 (+38.09%)	28.430 (+49.95%)
1,024	18.96	24.541 (+29.44%)	29.593 (+56.08%)	30.058 (+58.53%)
		27.461 (+44.84%)	26.182 (+38.09%)	28.430 (+49.95%)

表 7 動作周波数 (3DES) (単位:MHz)

CHs samples	None	8ch	16ch	32ch
512	38.39	36.62 (-4.61%)	35.58 (-7.32%)	40.57 (5.68%)
		38.87 (1.25%)	32.89 (-14.33%)	36.31 (-5.42%)
1,024	38.39	36.28 (-5.50%)	37.78 (-1.59%)	37.19 (-3.13%)
		38.87 (1.25%)	32.89 (-14.33%)	36.31 (-5.42%)

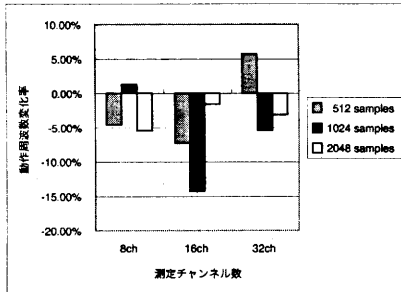


図 9 動作周波数への影響 (3DES)

(4) ピン数への影響

ユーザロジックにリモート・ロジックアナライザを接続した際の使用ピン数の増加量について測定を行った。リモート・ロジックアナライザ単体で使用するピンは 5 本であるため、動作クロックをユーザロジックと共用しない場合は 5 本、共用する場合は $5 - 1 = 4$ 本の追加のみでロジックアナライザを追加することができる。

(5) 回路合成時間への影響

表 8 に、ユーザロジックにリモート・ロジックアナライザを接続し、FPGA へ実装した際の回路合成時間を示す。この結果から、ユーザロジックのサイズに関わらず、チャンネル数とサンプル数に二乗倍的に合成時間が増加していることがわかる。現在は ESB の使用を明示的に記述していないため、コンパイラに依存する形で ESB の合成が行われていることが原因と考えられる。

表 8 回路合成時間 (単位:分'秒")

samples	CHs				
	None	8ch	16ch	32ch	
FA	512		2'04"	4'00"	10'47"
	1,024	0'47"	3'42"	10'52"	34'04"
	2,048		11'13"	37'23"	136'46"
3DES	512		4'05"	5'10"	8'53"
	1,024	3'18"	5'02"	8'49"	22'05"
	2,048		8'56"	22'54"	73'39"

5. まとめ

本研究では、遠隔再構成に対する技術的な課題を提示し、その問題解決の糸口として JTAG-ISP と Java に着目した。Java 言語を用いる前段階として、C 言語を用いてデバイス再構成プログラムを開発した。その設計にあたり、コンフィギュレーション手続きの差異に対応するため、プログラムを環境依存部と非依存部に分割する方針を取った。この結果、ネットワークを介して Java 言語を用いた FPGA デバイスの再構成を容易に行うための基盤となる環境を構築することができた。

また、組込みロジックアナライザの技術を応用し、ネットワークを介して遠隔地にあるデバイスの内部信号を観測するリモート・ロジックアナライザの開発を行った。この結果、遠隔地にあるデバイスに対して、テスト波形を与え、出力波形を取得し、ネットワークを介した PC 上から動作の検証を行うことが可能となった。

以上の技術を組み合わせ、インターネットに接続されている機器に搭載された FPGA を遠隔操作により再構成し、回路が正しく動作していることをリモート・ロジックアナライザによって検証することで、手元の開発環境からデバッグを行うことが可能となった。

従来はリモートサーバとして PC を用いていたが、組込み環境への適用を念頭に置き、Nios プロセッサをリモートサーバとする構成で動作を確認した。今後は遠隔サーバプログラムと再構成プログラムを Java 化することで、広く実用化に向けたシステム構築を行っていく予定である。

また検討すべき課題として、第三者からの予期せぬ再構成や構成データの漏えいを防ぐためのセキュリティ問題や、構成データの管理方法、Xilinx 社など他社製デバイスへの適用などが挙げられる。

文 献

- [1] 武田直樹, 土黒功司, 當摩佳重, 柴村英智, 久我守弘, 末吉敏則: “遠隔操作による FPGA/PLD デバイス再構成の実現,” DA シンポジウム 2001 論文集, pp.73-78, 2001 年 7 月.
- [2] 坂巻桂壽美: “JTAG テストの基礎と応用,” CQ 出版社, 1998.
- [3] “KVM ホワイトペーパー,” サン・マイクロシステムズ株式会社, <http://jp.sun.com/software/consumer-embedded/kvm/wp.html>.
- [4] “The JRunner Software Driver: An Embedded Solution for PLD JTAG Configuration,” 米 Altera 社, <http://www.altera.co.jp/literature/wp/wpjrunner.pdf>.
- [5] “ユーザーロジックの原理と測定手順,” 有限会社エムピーウェア, http://www.mbeware.com/01_3userlogi/u.genri.htm, 2001 年 5 月.