

# JTAG はバウンダリ・スキャンだけではない ～ MITOUJTAG で広がる JTAG テストの可能性

特殊電子回路株式会社 内藤 竜治

様々な信号を取り出したり注入したりすることができる。

## ■1. はじめに

JTAG と聞くと基板検査を思い浮かべるだろう。確かに JTAG の本来の目的はバウンダリ・スキャンテストによる基板検査であったが、現在は基板検査だけではなく、FPGA や CPLD の書き込み、CPU のデバッグ、FPGA の内部回路のデバッグ様々な分野で用いられている(図1)。これらの基板検査以外の機能はオプション機能とされていたのであるが、皮肉なことに、昨今 JTAG が使用されている場面はオプション機能の方が多いただろう。

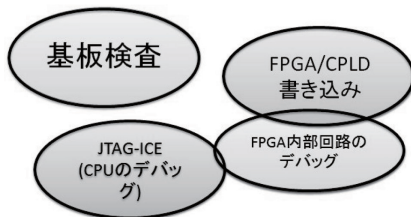


図1 JTAG の用いられている分野

JTAG には基板検査以外にも様々な用途があるが、お互いの分野はあまり関連していない

本稿では、従来の基板検査ソフトとは一風違ったバウンダリ・スキャンのやり方と、JTAG のオプション機能を活用する方法を紹介する。

## ●JTAG の基本的なしくみ

JTAG 対応 IC の中には IC 本来の機能を行う回路(コアロジック)があり、その周辺に JTAG の回路(TAP:Test Access Port)がつながっている(図2(a))。コアロジックは JTAG 回路とは無関係に動作していて、TAP は IC 本来の動作を妨げないようにしながら、

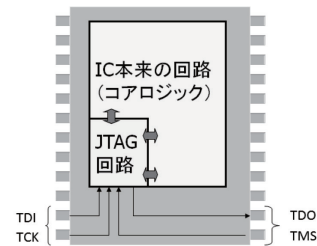


図2(a) JTAG デバイスの構造

TAP 内部には IR(命令レジスタ)と、複数の DR(データレジスタ)というシフトレジスタが設けられていて、これらのレジスタに様々な値やデータを入出力することによって様々な機能を発揮するようになっている(図2(b))。

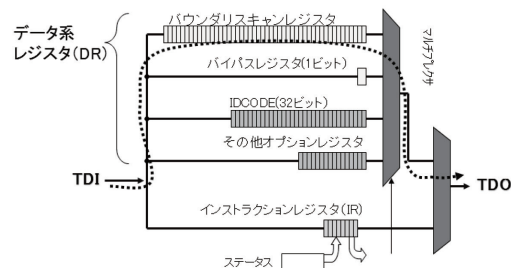


図2(b) 命令レジスタ(IR)とデータレジスタ(DR)

ところが、IEEE1149.1 の規格では DR と IR の操作方法としてバウンダリ・スキャンのやりかたしか定められておらず、FPGA の書き込みや、CPU のデバッグなどはプライベート命令とオプションレジスタを使って操作するという事になっている<sup>1</sup>。つまり、バウンダリ・スキャン以外のやり方であるオプション機能は完全に IC 依存なのである。

## ●従来のバウンダリ・スキャン・ツールの限界

が、採用されている事例は少なく、今後も発展することはないだろう。

<sup>1</sup> PLD の書き込みや CPU のデバッグ手順を標準化しようという動きもあった

従来の基板検査ツールではカバレッジ(検査可能範囲)を上げることを目的の一つとしていた。しかし、バウンダリ・スキャン検査のカバレッジに限界があることは言うまでもない。そもそもバウンダリ・スキャンでは、電源<sup>2</sup>、JTAG 端子そのもの、アナログ端子、一般的には USB や PCIe なども見ることができない。CPU/DSP では DDR メモリ用の信号にバウンダリ・スキャン・レジスタが通っていないものさえある。

また、面積やコストを削減するため SoC を採用したため基板上に JTAG 対応 IC が1つしかないという基板も多い。そういう基板では JTAG を使ってクラスタテストや、端子のプルアップ/プルダウンの判別程度のことではあるかもしれないが、JTAG 本来の強みであるインターコネクト検査はできない。

たとえ特定の基板で高いカバレッジを実現できたとしても、平均的な製品の基板における JTAG テストのカバレッジは決して高くない。バウンダリ・スキャンだけですべての基板検査ができるわけでは決してないし、IC が高機能化するにつれてカバレッジはさらに狭くなるだろう。JTAG を基板検査だけに使うのでは非常にもったいない。オプション機能を含めてより活用していく方法が求められてきた。

## ■2. MITOUJTAG の登場

そこで筆者は MITOUJTAG(みとうジェイタグ)という新しい JTAG ツールを開発した(図3)。MITOUJTAG は、平成 15 年度に未踏ソフトウェア創造事業<sup>3</sup>で採択された「包括的 JTAG サポートソフトウェアの開発」という成果を元に製品化した純日本製の JTAG ソフトウェアである。

従来の JTAG ソフトウェアが、基板検査、FPGA/PLD の書き込み、CPU のデバッグなど分野ごとに特化されていたが、それらの垣根を越えてあらゆる JTAG ソフトウェアを統合したものを作ろうと目指したのである。

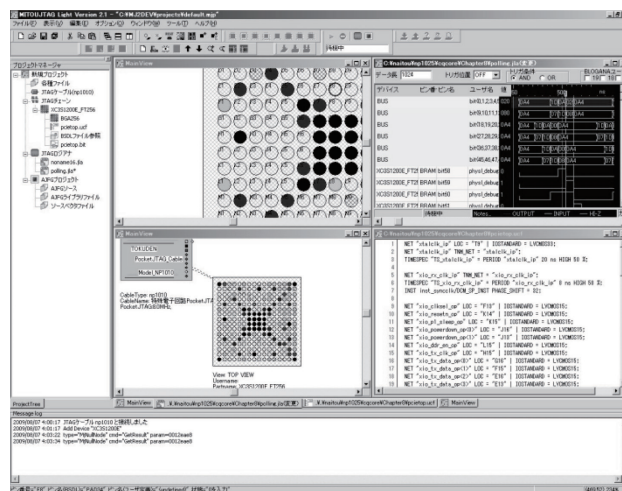


図3 MITOUJTAG の画面

MITOUJTAG は、『回路や基板を設計するエンジニアが、目の前にある1台の試作基板を、その場で思いついたいろいろなやり方で検査したい』という要求のもとに生まれてきた。BGA や CSP や多層基板を採用して、オシロスコープでは回路の信号を見ることができなくなった現在において、信号を見たり容易に操作したりできるようにすることを目的としている。

従来の JTAG 基板検査ツールが ICT や FCT のような工場の設備として発達してきたのに対して、MITOUJTAG は計測器として発達してきた。そのため、エンジニア一人一人の机の上におけるようなリーズナブルな価格設定になっている。具体的には、BGA や CSP を見るための測定器のように使える MITOUJTAG BASIC が¥99,330(税込)で、基板検査にも対応した MITOUJTAG Pro が¥312,900(税込)である。(表1)

表1 従来型 JTAG ツールと MITOUJTAG との違い

	従来型ツール	MITOUJTAG
用途と目的	基板検査のための設備	エンジニアのための測定器
基本的な使用方法	手順を作りこんでバッチ処理	画面を見ながらインタラクティブに操作
開発元	海外	日本

<sup>2</sup> 例えばパターンが広くフロー時の熱が逃げやすい電源端子に実装不良があっても JTAG ではわからない

<sup>3</sup> 独立行政法人情報処理推進機構(IPA)が主催し、斬新なソフトウェアを開発する個人を支援するという趣旨の事業

価格	高価	エンジニア一人ひとりに支給できる価格
----	----	--------------------

また、機能的にも「事前に準備するファイルがいない」「その場で思いついたことをインタラクティブにできる」といった手軽さで、回路設計を行うエンジニアに非常に高い評価を得ており、日本国内ではバウンダリ・スキャンツールの事実上の標準となっている。

### ■3. バウンダリ・スキャンの応用

#### ●バウンダリ・スキャン可視化

MITOUJTAG の最大の特徴は、図4に示すバウンダリ・スキャン可視化機能とあって、BGA や SOP や PLCC の IC の端子の状態を可視化してリアルタイムに表示するものである。図4の画面に表示された1つ1つの丸は BGA の端子を表していて、水色はロジックの L レベル、ピンクはロジックの H レベル、網掛けは入力を塗りつぶしは出力を表している。このように端子のロジックの H/L や入出力の状態がパソコンの画面上で見えるようになるが、IC 本来の動作を妨げることはなく、このテストを行っている間も IC は本来の動作を続けている。

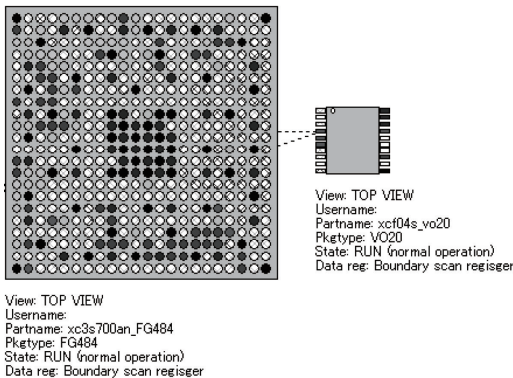


図4 バウンダリスキャン可視化の例

BGA と SOIC の動作を可視化している

MITOUJTAG で可視化を行うには、パソコンとターゲットボードとを専用の USB-JTAG アダプタでつなぎ(写真1)、MITOUJTAG を起動して「自動認識」ボタンを押すだけで非常に簡単である。

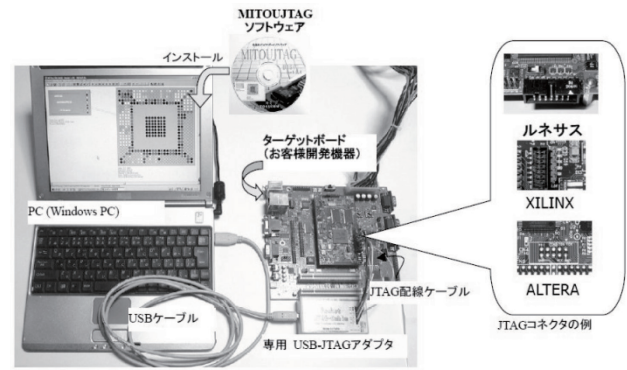


写真1 MITOUJTAG の使用例

MITOUJTAG には 7000 種類以上の JTAG 対応 IC の情報(BSDL)があらかじめ登録されているので、ネットリストもその他の定義ファイルもない。これほど簡単にバウンダリ・スキャンができるツールは他にない。

#### ●インタラクティブな端子の操作

画面に表示された BGA の端子をマウスでクリックすると、端子の H/L や入出力の方向を変えることができる。たとえば、MITOUJTAG で端子を操作しながら、それにつながっているであろう基板上的の Via やコネクタの端子にオシロスコープのプロブをあてておけば導通試験はすぐにできる。テストパターンを作ってからバッチ処理で流しこむのではなく、エンジニアが今やりたいようにやるのが MITOUJTAG である。

#### ●JTAG ロジックアナライザ

バウンダリ・スキャンで得られた端子の状態を波形として表示するのが「JTAG ロジックアナライザ機能」である。これは数百の信号の状態を波形として表示することができる(図5)。対応 IC のメーカーや品種は問わず、バウンダリ・スキャンに対応したあらゆる JTAG 対応 IC で使用できる。FPGA や CPLD に JTAG 用の回路(IP コア)や CPU のプログラムを書き込む必要もないので、IC が起動する前から使える。

信号の入出力に応じて、赤または緑の線で波形が描かれるので、従来のロジックアナライザやオシロスコープとは違い、信号の入出力も判別できる。また複数の信号を束ねてバス化し、16 進数やアナログ的な

波形として表示することもできる。

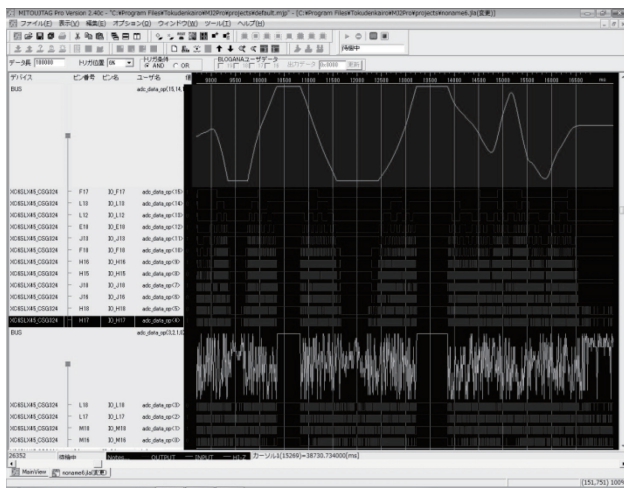


図 5 JTAG ロジックアナライザの画面

AD 変換器から受け取ったデータを FPGA の端子から出力し、その様子をバウンダリスキャンで可視化している

### ●パラレル・フラッシュ ROM の書き込み

NOR 型パラレル・フラッシュ ROM はマイコンのファームウェアの記憶によくつかわれている。MITOU JTAG ではパラレルフラッシュ ROM の読み書きが可能である。図6の画面でフラッシュ ROM の制御信号(CE、WR、D[\*]、A[\*]など)に接続されている JTAG 対応 IC の端子番号を設定していくだけで読み出し、書き込み、ベリファイなどが可能になる。

Node name	Dev num	Pin num	
CE	1	W19	xc3s700an_FG484, Pin=W19, Org.Name="PAD194", Name="NF_DE"
WE	1	W20	xc3s700an_FG484, Pin=W20, Org.Name="PAD193", Name="NF_DE"
WE	1	AA22	xc3s700an_FG484, Pin=AA22, Org.Name="PAD196", Name="NF_WE"
WP	1	E14	xc3s700an_FG484, Pin=E14, Org.Name="PAD74", Name="NF_WP"
A-1	1	T17	xc3s700an_FG484, Pin=T17, Org.Name="PAD192", Name="NF_A<0>"
A0	1	T18	xc3s700an_FG484, Pin=T18, Org.Name="PAD191", Name="NF_A<1>"
A1	1	R19	xc3s700an_FG484, Pin=R19, Org.Name="PAD184", Name="NF_A<2>"
A2	1	P18	xc3s700an_FG484, Pin=P18, Org.Name="PAD183", Name="NF_A<3>"
A3	1	N22	xc3s700an_FG484, Pin=N22, Org.Name="PAD182", Name="NF_A<4>"
A4	1	N21	xc3s700an_FG484, Pin=N21, Org.Name="PAD181", Name="NF_A<5>"
A5	1	N20	xc3s700an_FG484, Pin=N20, Org.Name="PAD180", Name="NF_A<6>"
A6	1	N19	xc3s700an_FG484, Pin=N19, Org.Name="PAD158", Name="NF_A<7>"
A7	1	N18	xc3s700an_FG484, Pin=N18, Org.Name="PAD156", Name="NF_A<8>"
A8	1	N17	xc3s700an_FG484, Pin=N17, Org.Name="PAD155", Name="NF_A<9>"
A9	1	K22	xc3s700an_FG484, Pin=K22, Org.Name="PAD144", Name="NF_A<10>"
A10	1	J22	xc3s700an_FG484, Pin=J22, Org.Name="PAD143", Name="NF_A<11>"
A11	1	J21	xc3s700an_FG484, Pin=J21, Org.Name="PAD138", Name="NF_A<12>"
A12	1	J20	xc3s700an_FG484, Pin=J20, Org.Name="PAD137", Name="NF_A<13>"
A13	1	H22	xc3s700an_FG484, Pin=H22, Org.Name="PAD136", Name="NF_A<14>"
A14	1	G22	xc3s700an_FG484, Pin=G22, Org.Name="PAD135", Name="NF_A<15>"
A15	1	H21	xc3s700an_FG484, Pin=H21, Org.Name="PAD130", Name="NF_A<16>"
A16	1	H20	xc3s700an_FG484, Pin=H20, Org.Name="PAD129", Name="NF_A<17>"
A17	1	F22	xc3s700an_FG484, Pin=F22, Org.Name="PAD128", Name="NF_A<18>"

図 6 フラッシュ ROM の設定

WR や CE などの端子が FPGA や CPU などのどの端子につながるかを指定している。

### ●導通検査機能

導通検査機能を使うと基板上にある複数の JTAG 対応 IC 間の接続を自動的に調べ上げることができる。また、1つ1つの端子がオープンなのか、プルアップ/プルダウンされているのか、VCC や GND にショートしているのかも判断できる。したがって、基板上に JTAG 対応 IC が 1 つしかない場合でも何らかの検査ができる。(図7)

使い方としては、確実に動作すると判明している基板でゴールデンデータを取得して、問題が発生している基板との相違を調べたり、回路図を見ながら想定外の接触がないかなどを調べるといった感じになる。

Device Name	Pin Number	Pin Name/Signal	Pin Name/Signal	Structure	Default	Structure	Structure	Pin Connection
104	92	PER4_10	Undefined	BEOR			DN0	Open
105	93	PER4_10	Undefined	BEOR			DN0	Open
106	94	PER4_11	Undefined	BEOR			DN0	Open
107	95	PER4_13	Undefined	BEOR			DN0	Open
108	97	PER4_14	ROM_NCE	BEOR			DN0	2056ROM_NCE
109	98	PER4_16	ROM_MCE	BEOR			DN0	2056ROM_MCE
110	99	PER5_01	ROM_D<7>	BEOR			DN0	2A48ROM_D<7> 3A48D0_A48
111	92	PER5_02	ROM_D<5>	BEOR			DN0	2A48ROM_D<5> 3A48D0_A48
112	92	PER5_04	ROM_D<5>	BEOR			DN0	2V12ROM_D<5> 3V12D0_V12
113	94	PER5_05	ROM_D<4>	BEOR			DN0	2W12ROM_D<4> 3W12D0_W12
114	94	PER5_07	ROM_D<3>	BEOR			DN0	2V12ROM_D<3> 3V12D0_V12
115	94	PER5_09	ROM_D<3>	BEOR			DN0	2V12ROM_D<3> 3V12D0_V12
116	97	PER5_10	ROM_D<1>	BEOR			DN0	2A484ROM_D<1> 3A484D0_A484
117	98	PER5_11	ROM_D<0>	BEOR			DN0	2A484ROM_D<0> 3A484D0_A484
118	98	COLK_A482	Undefined	BEOR			DN0	1206FCOLK_A482 3A482COLK_A482
119	98	COLK_A481	Undefined	BEOR			DN0	1158FCOLK_A481
120	95	FE48W_EN_B0	Undefined	BEOR			DN0	N/A
121	96	M0_A82	Undefined	BEOR			DN0	N/A
122	97	M1_A81	Undefined	BEOR			DN0	N/A
123	98	M2_A80	Undefined	BEOR			DN0	N/A
124	93	D0_A83	ROM_A<3>	BEOR			DN0	1132ROM_A<3>
125	94	D0_A84	ROM_A<3>	BEOR			DN0	1120ROM_A<3>
126	95	D0_A85	ROM_A<5>	BEOR			DN0	1118ROM_A<5>
127	97	D0_A87	ROM_A<12>	BEOR			DN0	1113ROM_A<12>
128	98	D0_A88	ROM_A<10>	BEOR			DN0	1116ROM_A<10>
129	99	D0_A89	ROM_A<3>	BEOR			DN0	1118ROM_A<3>
130	100	D0_A10	ROM_A<3>	BEOR			DN0	1126ROM_A<3>
131	101	D0_A11	USER_CLK	BEOR			DN0	Open
132	102	D0_A12	PCLAD<0>	BEOR			DN0	3A13PCLAD<0>
133	103	D0_A13	PCLAD<1>	BEOR			DN0	3A13PCLAD<1>
134	104	D0_A14	PCLAD<2>	BEOR			DN0	3A14PCLAD<2>
135	105	D0_A15	PCLAD<3>	BEOR			DN0	3A14PCLAD<3>

図7 導通検査中の画面

1つ1つの信号がどこにつながるかを自動的に調べている

### ●スクリプト機能

MITOUJTAG では、C++ 言語を用いてバウンダリ・スキンのやりかたを記述できる。たとえば、次のリスト1は簡単な例である。

リスト1 簡単な JTAG スクリプトの例

```
for (int i=0; i<10; i++) {
    ADDR = i;
    CLK = 1;
    CLK <= 0;
}
```

プログラム中の ADDR と CLK という変数は、C++ 言語でいうところのオブジェクトであり、この変数が変更されると基板上の信号がバウンダリ・スキンの操作されるという仕組みになっている。つまり、このプログラムは基板上の ADDR という信号に 0~9

までの値を出力し、クロックパルスを与えるものである。

このプログラムは、Visual C や、Borland C、GNU C といった標準的な C++コンパイラを使ってコンパイルされ、生成されたバイナリファイルが MITOUJTAG に組み込まれて動作する。独自仕様の言語ではなく標準的な C コンパイラを使用するため、習得が容易であり、コンパイラに起因するバグもなく安心して使用できる。もちろん、DDR メモリのテストなど再利用可能なものはライブラリ化されているので、ユーザがゼロから開発する必要はない。

スクリプトはあくまでも標準的な C++言語のプログラムであるので、Windows API の呼び出しや様々なソフトウェアライブラリとの結合も容易である。いわばユーザが作成した C/C++言語のプログラムに、バウンダリ・スキャンができるようなライブラリを提供しているといったほうが良いだろう。そのため、従来の ICT や FCT との連携も可能である。

### ●IC 固有のバグに対応

あまり公表されていないが、バウンダリ・スキャンレジスタが IEEE1149.1 に従っていない構成の IC というのも存在する。半導体メーカーが JTAG の回路を間違えて作ってしまい、そのまま修正されていないのであろう。MITOUJTAG ではそういう IC を見つけては特別な対策を追加してきたため、結果的に正しくスキャンできるようになっている。IC 固有のバグを知らないとバウンダリ・スキャンで入力に設定した端子から出力されてしまうという事故も起きかねないので危険である。

## ■4. オプション機能の活用

本章では JTAG のオプション機能を用いたデバッグ方法を紹介する。

### ●FPGA や CPLD の書き込み

MITOUJTAG は、ほぼすべての XILINX FPGA や CPU、ROM の書き込みに対応しているほか、一部の

ALTERA、LATTICE デバイスにも書き込みできる。

確かに従来の基板検査ツールでも PLD への書き込みができるものがあつたが、それらは SVF ファイルというものを經由していた。SVF は文法的に条件分岐ができないので、IC の消去待ちで最大時間を待たなければならず、非常に遅いという欠点があつた。それに対して MITOUJTAG は、各種 FPGA や CPLD の書き込みアルゴリズムを内部に持っているため、非常に高速な書き込みができ、操作も楽である。

### ●シリアルフラッシュ ROM の書き込み

近年、マイコンのファームウェアや基板固有のデータを記憶するため、シリアルフラッシュ ROM(SPI ROM)が多く使用されるようになってきている。

SPI ROM は 4 本の信号線を通じてアドレスやコマンドやデータをやりとりする。バウンダリ・スキャンを使って SPI ROM の内容を読み書きすることもできるが、SPI ROM では最初の 1 バイトのデータを読み出すためには 80 回程度の信号の遷移をしなければならず、バウンダリ・スキャンでこれを行うのは遅く、現実的ではない。

MITOUJTAG では XILINX FPGA に接続された SPI ROM に対して高速な読み書きが可能である。FPGA を SPI ROM アクセス用の回路にいったん書き換え、JTAG を通じて FPGA 中の回路と通信を行うことで SPI ROM に書き込むためである。つまり、FPGA の書き換えと、FPGA 内の回路との通信という 2 つのオプション機能を使っているわけである。(図8)

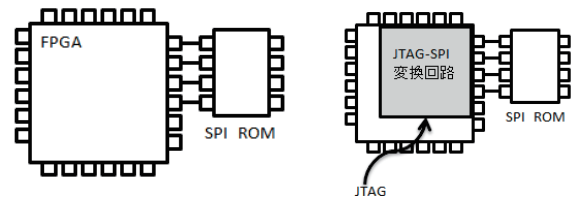


図8(a) 通常動作時

JTAG は使用されていない

図8(b) FPGA を書き換える

る

JTAG を用いて FPGA に

JTAG-

SPI 変換回路を書き込む

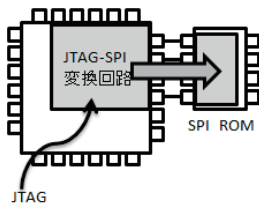


図8(c) JTAG で SPI ROM を書き換える

JTAG で FPGA の内部回路と通信し、SPI ROM を書き換える

### ●高速なロジックアナライザ

昨今の FPGA は非常に大きなメモリを搭載している。MITOUJTAG には、数百 MHz といった速さで高速にサンプリングして、FPGA の内蔵メモリに波形を溜めこんで表示するという機能もある。これを使うと内部信号など非常に高速な信号を観察できる。(図9)

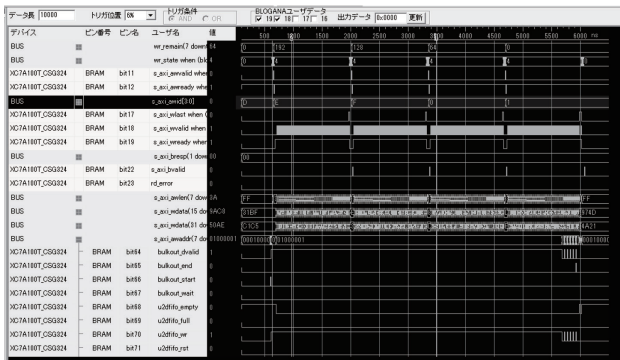


図9 FPGA 内部の信号を見たようす

100MHzで動作しているUSB3.0 インタフェース回路の内部信号

### ●CPU のデバッグ

おそらくJTAG のオプション機能で最も広く使われているものは CPU のデバッグ機能、いわゆる JTAG エミュレータであろう。MITOUJTAG にはルネサスエレクトロニクス製のマイクロプロセッサであるRX621/62N シリーズのデバッグ機能が搭載されている<sup>4</sup>。

エミュレータ機能を使用すると、CPU が現在命令を実行しているアドレス付近のアセンブラコードを表示し、内蔵レジスタを表示して停止する(図10)。この状

態でステップ実行させたり、メモリの内容を読み書きしたり、内蔵レジスタを書き換えたりできる。CPU が異常な状態に陥っていないかを瞬時に判断することができる。

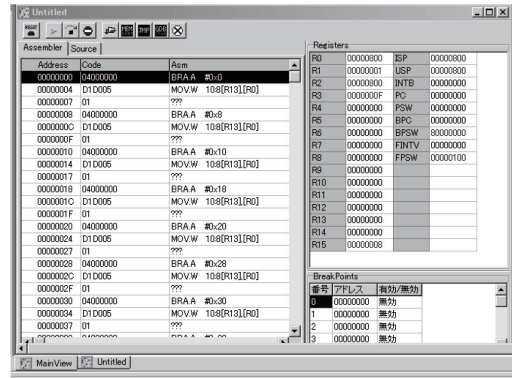


図10 CPU デバッグの様子

ルネサス製 CPU の動作を止めてステップ実行している

この機能を使うために何かのファイルやデータを用意する必要はない。つなぐだけで、すぐに現在の CPU 状態が見えるのである。もちろん、基板上の ROM の内容を書き換えずに使用できる、不良解析において故障の原因を上書きして消してしまうこともない。

### ■まとめ

MITOUJTAG は、エンジニア一人一人が使うための測定器として発展してきた。面倒な設定ファイルを用意することなく、オシロやロジアナや FPGA 書き込みツールとして気軽に使用できる。まさに「JTAG をつなげばすぐに見える」のである。

筆者が自ら開発しサポートしている日本製のソフトウェアであるため、技術的な問い合わせに対して速やかに回答することができるし、機能追加の要求に応えることもできる。実際に顧客の基板の特殊な仕様に合わせてするためにソフトウェア自体の改良も何度も行ってきた。

本稿では書き切れなかった JTAG バウンダリ・スキャンの原理やオプション機能の詳しい説明については、Web サイト「JTAG 情報」(<http://www.tokuden>)

<sup>4</sup>過去には ARM7 のエミュレータを内蔵していた。将来的には MIPS や SH、

ARM Cortex シリーズなどにも対応を広げる予定)

kairo.co.jp/jtag/)」に記載している。JTAG はバウンダリ・スキャンだけではなく様々な応用機能があり、それらを使えば基板検査の可能性も広がるということをご理解いただければ幸いです。