

この資料は英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。こちらの日本語版は参考用としてご利用ください。設計の際には、最新の英語版で内容をご確認ください。

III51003-1.5

はじめに

この章では、MAX II デバイスの IEEE Std. 1149.1 規格に準拠するバウンダリ・スキャン・テスト (BST) 回路の使用方法和以下の項を説明します。

- 3-1 ページの「IEEE Std. 1149.1 (JTAG) バウンダリ・スキャンのサポート」
- 3-5 ページの「イン・システム・プログラマビリティ」

IEEE Std. 1149.1 (JTAG) バウンダリ・スキャンのサポート

すべての MAX[®] II デバイスは、IEEE Std. 1149.1-2001 に準拠した JTAG (Joint Test Action Group) バウンダリ・スキャン・テスト (BST) 回路を搭載しています。JTAG バウンダリ・スキャン・テストは、V_{CCINT} およびすべての V_{CCIO} バンクに完全に給電され、t_{CONFIG} 時間が経過すればいつでも実行できます。MAX II デバイスは、Programming Object ファイル (.pof)、Jam[™] Standard Test and Programming Language (STAPL) ファイル (.jam) または Jam Byte-Code ファイル (.jbc) を使用する Quartus[®] II ソフトウェアまたはハードウェアとともに、JTAG ポートをイン・システム・プログラミングに使用することもできます。

JTAG ピンは、1.5 V、1.8 V、2.5 V、または 3.3 V の I/O 規格をサポートします。サポートされる電圧レベルと規格は、そのピンが存在するバンクの V_{CCIO} によって決まります。専用 JTAG ピンは、MAX II デバイスのバンク 1 内に存在します。

MAX II デバイスは、表 3-1 に示す JTAG インストラクションをサポートしています。

表 3-1. MAX II の JTAG インストラクション (1 / 2)

JTAG インストラクション	命令コード	説明
SAMPLE/PRELOAD	00 0000 0101	通常動作中のデバイスのピンから信号を取り込んでテストすることができる。また、初期データ・パターンをデバイス・ピンに出力させることができる。
EXTEST (1)	00 0000 1111	出力ピンにテスト・パターンを強制的に与え、入力ピンでテスト結果を取り込むことによって、外部回路との接続とボード・レベルの配線がテストできる。

表 3-1. MAX II の JTAG インストラクション (2 / 2)		
JTAG インストラクション	命令コード	説明
BYPASS	11 1111 1111	TDI ピンと TDO ピンの間に 1 ビットのバイパス・レジスタを配置することによって、デバイスに通常の動作をさせながら BST データが選択されたデバイスをバイパスして、隣接したデバイスに同期転送させることができる。
USERCODE	00 0000 0111	32 ビットの USERCODE レジスタを選択して TDI ピンと TDO ピンの間に配置することによって、USERCODE を TDO にシリアルにシフト・アウトさせることができる。このレジスタは、Quartus II ソフトウェアで指定されていない場合、デフォルトですべて 1 になる。
IDCODE	00 0000 0110	IDCODE レジスタを選択して TDI ピンと TDO ピンの間に配置することによって、IDCODE が TDO にシリアルにシフト・アウトさせることができる。
HIGHZ (1)	00 0000 1011	TDI ピンと TDO ピンの間に 1 ビットのバイパス・レジスタを配置することによって、すべての I/O ピンをトライ・ステートにした状態で、デバイスに通常の動作をさせながら BST データが選択されたデバイスをバイパスして、隣接したデバイスに同期転送させることができる。
CLAMP (1)	00 0000 1010	TDI ピンと TDO ピンの間に 1 ビットのバイパス・レジスタを配置することによって、I/O ピンをバウンダリ・スキャン・レジスタ内のデータで定義される状態に保持し、デバイスに通常の動作をさせながら BST データが指定したデバイスをバイパスして、隣接したデバイスに同期転送させることができる。
USER0	00 0000 1100	この命令によって、ユーザは MAX II ロジック・アレイの TDI と TDO の間に独自のスキャン・チェーンを定義できる。この命令は、カスタム・ロジックおよび JTAG インタフェースにも使用される。
USER1	00 0000 1110	この命令によって、ユーザは MAX II ロジック・アレイの TDI と TDO の間に独自のスキャン・チェーンを定義できる。この命令は、カスタム・ロジックおよび JTAG インタフェースにも使用される。
IEEE 1532 命令	(2)	JTAG ポートを介して MAX II デバイスをプログラミングするときに使用される IEEE 1532 ISC 命令。

表 3-1 の注:

- (1) HIGHZ、CLAMP、および EXTEST 命令は、ウィーク・プルアップ抵抗やバス・ホールド機能をディセーブルすることはありません。
- (2) これらの命令は 1532 BSDL ファイルに記載されています。1532 BSDL ファイルは、アルテラ・ウェブサイト (www.altera.co.jp) に適宜掲載されます。



MAX II デバイスに不定状態になる可能性があるため、サポートされていない JTAG インストラクションを発行しないでください。デバイス動作を回復するにはパワー・サイクルが必要です。

MAX II デバイスのインストラクション・レジスタの長さは 10 ビットです。また、USERCODE レジスタの長さは 32 ビットです。表 3-2 および表 3-3 に、MAX II デバイスのバウンダリ・スキャン・レジスタの長さ と IDCODE 情報を示します。

表 3-2. MAX II のバウンダリ・スキャン・レジスタの長さ

デバイス	バウンダリ・スキャン・レジスタの長さ
EPM240	240
EPM570	480
EPM1270	636
EPM2210	816


表 3-3. MAX II デバイスの 32 ビット IDCODE

デバイス	バイナリ IDCODE (32 ビット) (1)				HEX IDCODE
	バージョン (4 ビット)	部品番号	メーカーの ID (11 ビット)	LSB (1 ビット) (2)	
EPM240 EPM240G	0000	0010 0000 1010 0001	000 0110 1110	1	0x020A10DD
EPM570 EPM570G	0000	0010 0000 1010 0010	000 0110 1110	1	0x020A20DD
EPM1270 EPM1270G	0000	0010 0000 1010 0011	000 0110 1110	1	0x020A30DD
EPM2210 EPM2210G	0000	0010 0000 1010 0100	000 0110 1110	1	0x020A40DD
EPM240Z	0000	0010 0000 1010 0101	000 0110 1110	1	0x020A50DD
EPM570Z	0000	0010 0000 1010 0110	000 0110 1110	1	0x020A60DD

表 3-2 の注：

- (1) 左側が最上位ビット (MSB) です。
- (2) IDCODE の最下位ビット (LSB) は常に 1 です。

 JTAG AC 特性については、「MAX II デバイス・ハンドブック」の「DC およびスイッチング特性」の章を参照してください。

 JTAG BST について詳しくは、「MAX II デバイス・ハンドブック」の「MAX II デバイスの IEEE 1149.1 (JTAG) バウンダリ・スキャン・テスト」の章を参照してください。

JTAG ブロック

MAX II JTAG ブロック機能により、JTAG TAP に USER0 または USER1 命令が発行されたときに、JTAG TAP およびステートにアクセスすることができます。USER0 命令および USER1 命令は、MAX II デバイスのバウンダリ・スキャン・セルではなくユーザ・ロジックを介して、JTAG バウンダリ・スキャン・チェーン (TDI) を伝達します。各 USER 命令によって、ロジック・アレイに対して独自のユーザ定義 JTAG チェインが1つ許容されます。

パラレル・フラッシュ・ローダ

JTAG トランスレータは JTAG デバイスを非 JTAG デバイスにインタフェースできるため、イン・サーキット・テスト中にプログラミングを必要とする汎用フラッシュ・メモリ・デバイス (Intel または Fujitsu ベースのデバイス) に最適です。フラッシュ・メモリ・デバイスは、FPGA コンフィギュレーション用またはシステム・メモリのパーツとして使用できます。多くの場合、MAX II デバイスは FPGA とフラッシュ・デバイス間のコンフィギュレーション・デバイスとして、これらのデバイスに接続済みです。ISP 対応の CPLD デバイスとは異なり、バルク・フラッシュ・デバイスには JTAG TAP ピンや JTAG TAP 接続はありません。小型フラッシュ・デバイスの場合、接続されたデバイスのシリアル JTAG スキャン・チェーンをしようして、非 JTAG フラッシュ・デバイスをプログラムするのが一般的です。この方法は、ほとんどの場合は低速かつ非効率であり、また大型パラレル・フラッシュ・デバイスに対しては実用的ではありません。MAX II デバイスの JTAG ブロックをパラレル・フラッシュ・ローダとして使用して、フラッシュの内容をプログラムおよび検証すると、テスト中にイン・システム・プログラミングを高速かつ経済的に実行できます。図 3-1 に、パラレル・フラッシュ・ローダとして使用される MAX II を示します。

図 3-1. MAX II パラレル・フラッシュ・ローダ

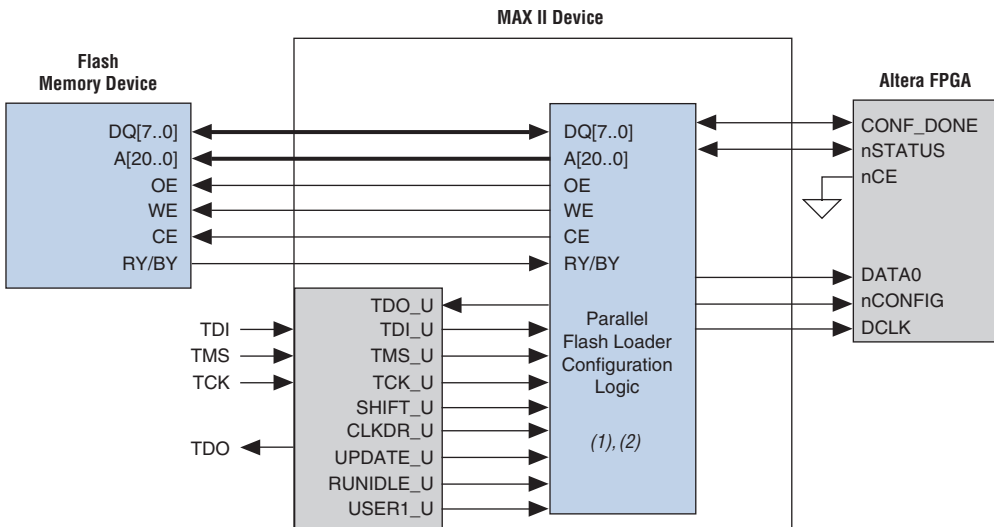


図 3-1 の注:

- (1) このブロックは LE に実装されています。
- (2) この機能は Quartus II ソフトウェアでサポートされています。

イン・システム・プログラマビリティ

MAX II デバイスは、業界標準の 4 ピン IEEE Std. 1149.1 (JTAG) インタフェースを介して、イン・システムでプログラムできます。イン・システム・プログラマビリティ (ISP) によって、開発およびデバッグ段階での設計変更とプログラムを迅速かつ効率的に繰り返し行うことができます。MAX II アーキテクチャのロジック、回路、配線は、フラッシュ・バースの SRAM コンフィギュレーション・エレメントでコンフィギュレーションされます。これらの SRAM エレメントでは、デバイスに電源が供給されるたびにコンフィギュレーション・データをロードする必要があります。SRAM のデータをロードするプロセスは、コンフィギュレーションと呼ばれます。オンチップ・コンフィギュレーション・フラッシュ・メモリ (CFM) ブロックは、SRAM エレメントのコンフィギュレーション・データを格納します。CFM ブロックは、再プログラム可能なフラッシュ・アレイにデザインのコンフィギュレーション・パターンを格納します。ISP の間に、MAX II JTAG および ISP 回路は、CFM ブロックの不揮発性フラッシュ・アレイにデザイン・パターンをプログラムします。

MAX II JTAG および ISP コントローラは、内部で CFM セルのプログラムに要求される高いプログラミング電圧を生成するため、どの推奨動作外部供給電圧 (3.3 V/2.5 V または MAX IIG デバイスの場合は 1.8 V) でもイン・システム・プログラミングが可能です。ISP を実行することができます。V_{CCINT} およびすべての V_{CCIO} バンクに完全に給電され、コンフィギュレーション・パワーアップ・タイムが経過した後は、いつでも ISP を実行することができます。デフォルトでは、I/O ピンイン・システム・プログラミング中にトライ・ステートになり、V_{CCIO} にウイーク・プルアップされて、ボードの競合を回避します。イン・システム・プログラミング・クランプおよびリアルタイム ISP 機能によって、ISP 中にユーザが I/O の状態や動作を制御することができます。

詳細については、3-9 ページの「イン・システム・プログラミング・クランプ」および 3-9 ページの「リアルタイム ISP」を参照してください。

これらのデバイスは、イン・システム・プログラミングが中断されたときに安全に動作させる ISP_DONE ビットを提供します。この ISP_DONE ビットは最後にプログラムされるビットで、このビットがプログラムされるまですべての I/O ピンのドライブを防止します。

IEEE 1532 サポート

MAX II デバイスの JTAG 回路および ISP 命令セットは、IEEE 1532-2002 プログラミング規格に準拠しています。これによって、JTAG チェイン内にある複数のベンダのプログラマブル・ロジック・デバイス (PLD) 間でのイン・システム・プログラミングに対応する業界標準ハードウェアおよびソフトウェアが実現します。

MAX II 1532 BSDL ファイルは、アルテラ・ウェブサイトですぐリリースされます。

STAPL (Jam Standard Test and Programming Language)

Jam STAPL JEDEC 規格、JESD71 は、イン・サーキット・テスト、PC、またはエンベデッド・プロセッサを搭載した MAX II デバイスをプログラムするのに使用できます。MAX II デバイスでは、Jam バイト・コードもサポートされています。これらのソフトウェア・プログラミング・プロトコルは、MAX II デバイスをプログラムするためのコンパクトなエンベデッド・ソリューションを提供します。



詳細については、「MAX II デバイス・ハンドブック」の「エンベデッド・プロセッサを使用した ISP 用 Jam STAPL の使用」の章を参照してください。

プログラミング・シーケンス

イン・システム・プログラミング中に、1532 の命令、アドレスおよびデータは、TDI 入力ピンを通して MAX II デバイスにシフトされます。データは、TDO 出力ピンを通してシフト・アウトされ、所期のデータと比較されます。デバイスにパターンをプログラムするには、以下の 6 つの ISP ステップが必要です。プログラムされたパターンのスタンドアロン検証には、ステージ 1、2、5、および 6 だけが必要です。これらのステップは、サードパーティ・プログラム、Quartus II ソフトウェア、または Jam STAPL および Jam Byte-Code Players によって自動的に実行されます。

1. ISP への移行 — ISP への移行ステージによって、I/O ピンをユーザ・モードから ISP モードにスムーズに遷移させます。
2. ID のチェック — プログラムまたは検証プロセス前に、シリコン ID がチェックされます。このシリコン ID の読み出しに要する時間は、プログラミング時間全体と比較すると短時間です。
3. セクタ消去 — イン・システムでのデバイスの消去には、デバイスを消去するための命令をシフト・インして消去パルスを印加します。消去パルスは、CFM ブロックに対して指定された消去パルス時間 500 ms、そして UFM ブロックの各セクタに対して指定された消去パルス時間 500 ms の間、実行 / テスト / アイドル・ステートで待機することにより、内部で自動的に生成されます。
4. プログラム — イン・システムでデバイスをプログラムするには、アドレス、データ、およびプログラム命令のシフト・イン、そしてフラッシュ・セルをプログラムするためのプログラム・パルスの生成が必要です。プログラム・パルスは、指定されたプログラム・パルス時間 75 μ s の間、実行 / テスト / アイドル・ステートで待機することにより、内部で自動的に生成されます。このプロセスは、各 CFM および UFM ブロックの各アドレスごとに繰り返されます。
5. 検証 — イン・システムで MAX II デバイスを検証するには、アドレスのシフト・イン、検証命令によるリード・パルスの生成、および比較のためのデータのシフト・アウトを実行します。このプロセスは、各 CFM および UFM アドレスごとに繰り返されます。
6. ISP への終了 — ISP への終了ステージによって、I/O ピンを ISP モードからユーザ・モードにスムーズに遷移させます。

表 3-4 に、ハードウェアのアルゴリズム・ベクタを実行するためのイン・サーキット・テストを使用した MAX II デバイスのプログラミング時間を示します。ダウンロード・ケーブルを使用したソフトウェア・ベースのプログラミング・ツールは、データ処理および転送制限により、若干遅くなります。

説明	EPM240 EPM240G EPM240Z	EPM570 EPM570G EPM570Z	EPM1270 EPM1270G	EPM2210 EPM2210G	単位
消去 + プログラム (1 MHz)	1.72	2.16	2.90	3.92	秒
消去 + プログラム (10 MHz)	1.65	1.99	2.58	3.40	秒
検証 (1 MHz)	0.09	0.17	0.30	0.49	秒
検証 (10 MHz)	0.01	0.02	0.03	0.05	秒
プログラム・サイクルの完了 (1 MHz)	1.81	2.33	3.20	4.41	秒
プログラム・サイクルの完了 (10 MHz)	1.66	2.01	2.61	3.45	秒

UFM プログラミング

Quartus II ソフトウェアは、POF、Jam、または JBC ファイルを使用することにより、CFM ブロックに格納されたロジック・アレイ・デザイン・パターンに関係なく、各ユーザ・フラッシュ・メモリ (UFM) ブロックのプログラミングをサポートします。これによって、現在のロジック・アレイ・デザインを変更することなく、ISP を経由して UFM の内容の更新または読み出し (またはその逆の処理) が可能になります。デフォルトでは、これらのプログラミング・ファイルおよび方法は、CFM ブロックと UFM の内容を含むフラッシュ・メモリ全体の内容をプログラムします。スタンドアロンのエンベデッド Jam STAPL Player および Jam Byte-Code Player は、フラッシュ・メモリ全体 (UFM と CFM の両方) またはそれぞれを個別にプログラムしたり、読み出すための動作コマンドを提供します。



詳細については、「MAX II デバイス・ハンドブック」の「[エンベデッド・プロセッサを使用した ISP 用 Jam STAPL の使用](#)」の章を参照してください。

イン・システム・プログラミング・クランプ

デフォルトでは、ISP への移行に使用される IEEE 1532 命令は、ISP シーケンスの間、ウィーク・プルアップ抵抗ですべての I/O ピンを自動的にトライ・ステートにします。ただし、一部のシステムでは、イン・フィールド・アップデートの間、MAX II デバイスの特定のピンで特定の DC ロジック・レベルを維持する必要があります。これらのシステムの場合、ISP シーケンス中の I/O 動作を制御するために、MAX II 回路にオプションのイン・システム・プログラミング・クランプ命令が存在します。イン・システム・プログラミング・クランプ命令によって、デバイスは出力ピン上の値をサンプリングして保持する（サンプリングされた場合、入力ピンはトライ・ステートに維持される）が、あるいは任意にピンス上でロジック High、ロジック Low、またはトライ・ステート値を明示的に設定できます。これらのオプションの設定は、Quartus II ソフトウェアを使用して、個々のピンごとに制御されます。



詳細については、「MAX II デバイス・ハンドブック」の「MAX II デバイスのリアルタイム ISP および ISP クランプ」の章を参照してください。

リアルタイム ISP

I/O ピンを DC ロジック・レベルより高く制御する必要があるシステムでは、リアルタイム ISP 機能を利用することにより、現在のデザインが SRAM ロジック・アレイおよび I/O ピンで動作している間に、新しいデザイン・イメージで CFM ブロックをアップデートできます。新しいプログラミング・ファイルは、元のデザインの動作を停止させることなく、アップデートされて MAX II デバイスに転送されるため、リモートまたはフィールド・アップグレードのためのダウン・タイム・コストが節約されます。アップデートされた CFM ブロックは、次のパワー・サイクルで新しいデザインを SRAM にコンフィギュレーションします。また、ISP コマンドの特定のシーケンスを使用することによって、パワー・サイクルなしで SRAM のコンフィギュレーションをすぐに実行することも可能です。パワー・サイクルを使用しないで SRAM をコンフィギュレーションするには、一定時間 (t_{CONFIG}) を要します。この期間中、I/O ピンはトライ・ステートになり、 V_{CCIO} にウィーク・プルアップされます。

デザイン・セキュリティ

すべての MAX II デバイスには、CFM ブロックにプログラムされたデータへのアクセスを制御するプログラム可能なセキュリティ・ビットが組み込まれています。このビットがプログラムされているときには、CFM ブロックに格納されたデザイン・プログラミング情報をコピーしたり、取り出すことはできません。この機能により、フラッシュ・メモリ・セル内のプログラム済みデータを見るのが不可能になるため、高度なデザイン・セキュリティが実現されます。この機能を制御するセキュリティ・ビットは、他のすべてのプログラム済みデータと同様に、デバイスが消去されたときのみにリセットされます。セキュリティ・ビットの設定に関係なく、SRAM も隠蔽されアクセスできません。UFM ブロック・データは、セキュリティ・ビットで保護されないため、JTAG またはロジック・アレイ接続を介してアクセスできます。

外部ハードウェアによるプログラミング

MAX II デバイスは、イン・サーキット・テスト、エンベデッド・プロセッサ、アルテラの ByteblasterMV™、MasterBlaster™、ByteBlaster™ II、および USB-Blaster ケーブルを介して情報をダウンロードすることによってプログラムできます。

BP Microsystems、System General、およびその他のプログラミング・ハードウェア・メーカーから、アルテラ・デバイスのプログラミング・サポートが提供されています。デバイスのサポート情報については、各メーカーのウェブサイトを参照してください。

参考資料

この章では以下のドキュメントを参照しています。

- 「MAX II デバイス・ハンドブック」の「DC およびスイッチング特性」の章
- 「MAX II デバイス・ハンドブック」の「MAX II デバイスの IEEE 1149.1 (JTAG) バウンダリ・スキャン・テスト」の章
- 「MAX II デバイス・ハンドブック」の「MAX II デバイスのリアルタイム ISP および ISP クランプ」の章
- 「MAX II デバイス・ハンドブック」の「エンベデッド・プロセッサを使用した ISP 用 Jam STAPL の使用」の章

改訂履歴

表 3-5 に、本資料の改訂履歴を示します。

表 3-5. 改訂履歴		
日付 & ドキュメント・バージョン	変更内容	概要
2007 年 12 月 v1.5	<ul style="list-style-type: none"> ● 表 3-1 の後にワーニング・ノートを追加。 ● 表 3-3 および表 3-4 を更新。 ● 「参考資料」の項を追加。 	—
2006 年 12 月 v1.4	改訂履歴を追加。	—
2005 年 6 月 v1.3	テキストおよび表 3-4 を追加。	—
2005 年 6 月 v1.3	ページ 3-5 ~ 3-8 のテキストを更新。	—
2004 年 6 月 v1.1	図 3-1 を修正。頭字語の CFM を追加。	—

