自動テスト装置とサードパーティのツールのSVFファイルを使用するXC9500デバイスのイン・システム・プログラミング

XAPP067 (バージョン1.1)　　アプリケーション・ノート

概要

このアプリケーション・ノートでは、標準の自動テスト装置を使用してXC9500デバイスをイン・システムでプログラムする方法を説明します。

ザイリンクス・ファミリ

XC9500

1

はじめに

XC9500**デバイスでは、イン・システム・プログラミング**(ISP)**と**IEEE1149.1**バウンダリ・スキャン**(JTAG)**テストの両方に対して、標準の**4**線式テスト・アクセス・ポート**(TAP)**を使用しています。これにより、システム全体の製造コストを削減し、さらにこれらデバイスのプログラムとテストの際に、自動テスト装置**(ATE)**とバウンダリ・スキャンベースの開発ツールを使用することによる不必要な処理に起因するデバイスの損傷を削減することができます。図1に、**XC9500**バウンダリ・スキャンのアーキテクチャを示します。**

ザイリンクスEZTag™**ソフトウェアは、シリアル・ベクタ・フォーマット**(SVF)**ファイルを自動生成してこの実現を支援します。この**SVF**ファイルには**XC9500**デバイスに必要とされるプログラミングとテストのアルゴリズムが記述されています。大部分の**ATE**プラットフォームとバウンダリ・スキャンベースの開発ツールは、**SVFをテスト・ベクタの入力フォーマットとして使用することができます。このアプリケーション・ノートでは**SVF**ファイルを生成する各ステップについて説明し、さらに**ATE**がこのファイルを使用してデバイスのプログラムとテストを行う方法を説明します。**

SVF の概要

最初のシリアル・ベクタ・フォーマットは、テスト生成ソフトウェアやATE**などのようなツールの間でバウンダリ・スキャン・テスト・ベクタを交換する必要性に対して、**Texas Instruments**社と**Teradyne**社により共同開発されました。当時は、**IEEE**標準**1149.1**の使用は増えつつありましたが、共通のデータ交換要求を満たす共通フォーマットまたは言語は存在していませんでした。**

SVF**の開発者は、テスト・ベクタを使用して**IEEE 1149.1 TAP**に対して単に**TCK (**クロック**)と**TMS (**モード制御**)信号を提供するだけのフォーマットは選択しませんでした。**SVF**フォーマットの基礎となったモデルでは、代わりに全ての動作は安定状態で開始され終了するものと想定していました。この結果、スティミュラス・ベクタの記述は非常に単純で簡潔になりました。**

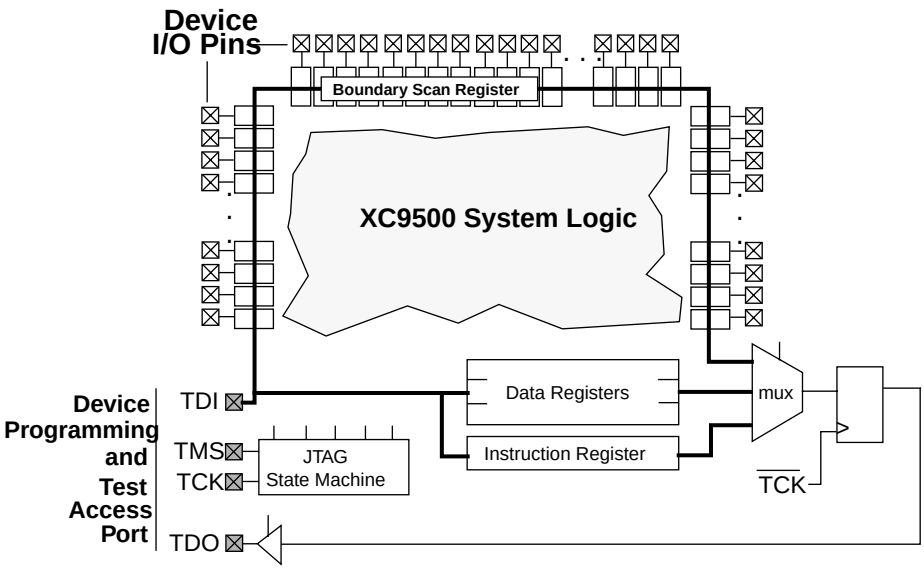

図1: XC9500 **バウンダリ・スキャン・アーキテクチャ**

1991年中頃と1994年秋の間に、SVFの3つのバージョンが開発されました。これらのバージョンではテスト・アプリケーションに依存しないフォーマットの生成が目標になっていました。1994年までには、100社以上がSVFベースのツールを開発し、CAEツールとATEの少なくとも10社のベンダがSVFをサポートするようになりました。

SVFはバウンダリ・スキャンTAPとそれを駆動するソフトウェアの間でのデータ交換に対して有用で信頼性の高いフォーマットであることが実証されました。

SVFの仕様

XC9500 ISPの目的に対しては、標準で規定されている13個のSVFレコードの内の3レコードのみが必要です。この節ではこれら3個のレコードについて説明します。

SVFファイルには一組のASCIIステートメントが記述されています。1行の最大文字数は256ですが、1つのSVFステートメントは複数行に跨ることができます。各ステートメントは、コマンドと複数のパラメータで構成され、セミコロンで終了します。SVFでは大文字/小文字の区別はなく、コメントは行の先頭の感嘆符(!)またはダブル・スラッシュ(//)で始まり最後のキャリッジ・リターンで終了します。

ステートメント内のスキャン・データは16進数で表現され、必ず括弧で囲まれます。スキャン・データでは指定ビット長を超えるデータ・ストリング長を指定することはできません。16進表現の上位ビット(MSB)の"0"はデータ長に含めません。スキャン・データのビット順としては、TDIキーワードとSMASKキーワードにより指定されたスキャン・データをデバイスに入力する際と、TDOキーワードとMASKキーワードにより指定されるデータをスキャン出力する際に、LSB (右端のビット)をそれぞれ先頭ビットに定めています。

XC9500 EZTagソフトウェアでは次のSVFコマンドをサポートしています。

- SDR (スキャン・データ・レジスタ)
- SIR (スキャン・インストラクション・レジスタ)
- RUNTEST

以下の各コマンド説明ではパラメータは必須です。オプションのパラメータはカギ括弧([ ])で囲んであります。変数は斜字体で示してあります。16進数値を表示するときは"( )"を使用します。

スキャン動作はSIRコマンドまたはSDRコマンドとその関連ヘッダ・コマンドまたはトレイラ・コマンドの実行として定義されます。

SDR、SIR

SDR length [TDI (*tdi*)] [TDO (*TDO*)] [MASK (*msk*)] [SMASK (*smask*)][PIO (*pio*)];

SIR length [TDI (*tdi*)] [TDO (*TDO*)] [MASK (*msk*)] [SMASK (*smask*)][PIO (*pio*)];

これらのコマンドは、ターゲットのスキャン・レジスタに入力するスキャン・パターンを指定します。SDRコマンド(スキャン・データ・レジスタ)は、ターゲットのデバイス・データ・レジスタにスキャン入力されるデータ・パターンを指定します。SIRコマンド(スキャン・インストラクション・レジスタ)は、ターゲット・デバイス・インストラクション・レジスタにスキャン入力されるデータ・パターンを指定します。

パラメータ:

length—32ビットの10進整数でスキャン対象のビット数を指定します。

[TDI (*tdi*)]—(オプション)ターゲットにスキャン入力される値を16進で指定します。このパラメータが指定されない場合は、ターゲット・デバイスにスキャン入力されるTDIの値には前のSDR/SIRステートメントで指定されたTDI値が使用されます。新しいスキャン・コマンドが指定されると、前のスキャンのデータ・パターン長が変更されます。TDIパラメータは必ず指定する必要があります。指定しない場合はデフォルトのTDIパターンが不定になりエラーが発生します。

[TDO (*tdo*)]—(オプション)ターゲット・デバイスからスキャン出力された実際の値と比較するテスト値を指定し、16進文字列で表されます。このパラメータ指定が省略されると、比較は行われません。TDOパラメータの指定が省略されると、MASKは使用されません。

[MASK (*mask*)]—(オプション)TDO値とターゲット・デバイスからスキャン出力された実際の値とを比較する際に使用するマスクを指定し、16進文字列で表されます。"0"のビット位置でその位置に対する"don't care"を指定します。このパラメータ指定が省略されると、SIR/SDRステートメントに対して指定された前のMASK値が使用されます。新しいスキャン・コマンドが指定されると、前のスキャンのデータ・パターン長が変更されます。この場合、MASKパラメータは必ず指定する必要があります。指定しない場合はデフォルトのMASKパターンが不定になりエラーが発生します。TDOパラメータの指定が省略されると、MASKは使用されません。

[SMASK (*smask*)]—(オプション) これは"don't care"のTDIデータを指定し、16進文字列で表されます。"0"のビット位置でその位置にあるTDIデータの"don't care"を指定します。このパラメータ指定が省略されると、SIR/SDRステートメントに対して指定された前のSMASK値が使用されます。新しいスキャン・コマンドが指定されると、前のスキャンのデータ・パターン長が変更されます。この場合、SMASKパラメータは必ず指定する必要があります。指定しない場合はデフォルトのSMASKパターンが不定になりエラーが発生します。TDIパラメータの指定が省略されても、SMASKは使用されます。

例:

```
SDR 24 TDI (000010) SMASK (0) TDO (818181)
MASK (FFFFFF);

SIR 16 TDO (ABCD);
```

RUNTEST

RUNTEST run_count TCK;

このコマンドは、指定のTCKクロック数の間ターゲットIEEE 1149.1バスをRun-Test/Idleステートにします。TAPの動作でレイテンシ(latency)周期を指定するときに使用できます。

パラメータ:

*run_count*ー1149.1**バスが**Run-Test/Idle**ステートに留まる時間を指定する**32**ビットの符号なし数値で**TCK**クロック数で表します。**

例:

RUNTEST 1000 TCK;

XVF**ファイルの例を次に示します。**

```
! Begin Test Program
TRST OFF;      !disable test reset line
ENDIR IDLE;    !End IR scan in IDLE
SIR 8 TDI (FE) MASK (FF)
SDR 14 TDI (3afe) MASK (3ff) TDO (0003)
    SMASK (3ff)
RUNTEST 100 TCK
!End test program
```

EZTag を使用して SVF ファイルを生成する方法

このプロシージャはSVF**ファイルの生成法を示します。ここでは、**XC9500**フィッタ・ソフトウェアと**EZTag**ソフトウェアを含んでいるザイリンクス**XACT**バージョン**6.0.0**ソフトウェアまたはそれより新しいソフトウェアの使用を想定しています。**

1. XABEL-CPLD**または互換性を持つサード・パーティのデザイン入力ツールを使用してデザインを生成します。**
2. **デザインをフィットして**JEDEC**出力ファイルに保存します。**
3. **次のコマンドを使用して**XACT**コマンド・ラインから**EZTag**ソフトウェアを起動します。**

   eztag -svf

   次のメッセージが表示されます。

   ```
   Xilinx (R) EZTAG XC9500-CPLD-6.0.0 - JTAG
   Boundary-Scan Download
   Copyright (C) Xilinx Inc. 1991-1995. All
   Rights Reserved.
   --------------------------------------
   SVF GENERATION MODE.
   EZTAG?
   ```

4. EZTAG?**のプロンプトに対して次のコマンドを入力します。**

   part *deviceType1: designName1*
   deviceType2: designName2
   deviceTypeN: designNameN <CR>

   ここでdesignName**は**SVF**へ変換するデザインの名前です。複数の**deviceType:designName**の対はスペースで区切って指定されます。**

   このコマンドは、1個から任意数のデバイスによるJTAG**デバイス・チェーンを定義します。**part**コマンド内で指定された部品は先頭デバイスが**TDI**を入力し、最終デバイスが**TDO**を出力する順序で並べる必要があります。**

   注: JTAG**チェーン内にある非**XC9500**部品に対しては、指定部品に対する**BSDL**ファイルが**XACT**パス内に存在し、かつ**deviceType.bsd**となっていることを確認してください(例えば、**PC84**パッケージの**XC4003**に対しては**4003pc84.bsd)。

5. **次のいずれかのコマンドを入力します。**

   erase *designName* ー**指定の部品を消去するビット・シーケンスを定める**SVF**ファイルを生成します。**

   verify *designName* [-j *jedecFileName*]ー**デバイスの内容をリードバックし、指定の**JEDEC**ファイルの内容と比較するビット・シーケンスを定める**SVF**ファイルを生成します。**

   program [-b] *designName* [-j *jedecFileName*]ー JEDEC**ファイル名が**designName.jed (**あるいは、**"-j"**の後に指定した**JEDEC**ファイル名)のファイルを使用して指定の部品をプログラムするビット・シーケンスを定める**SVF**ファイルを生成します。プログラム・コマンド・オプション**により次の機能を追加することができます。

   -bー**工場から出荷されたばかりの新しいデバイスを使用するときは、このスイッチを使用して、プログラミングの前に実行される消去処理をスキップすることができます。前にプログラムされたことがないデバイスに対しては消去は不要です。**

6. **次のコマンドにより**EZTag**を終了します。**

   quit

   注: SVF**ファイルには**designName.svf**の名前が与えられ、カレント・ディレクトリ**(EZTAG**が動作しているディレクトリ**)内に生成されます。同一の**designName**ファイルに対する以降の操作により、その都度**SVF**ファイルが上書きされます。**SVF**ファイルには、指定の機能を実行するために必要な全てのデータとコマンドが含まれています。**

SVF の解釈

SVFの単純性は主要な弱点でもあります。多くのSVF**の動作は標準では規定されないままになっています。アプリケーションに対して**SVF**スティミュラスを最適化すためには、幾つかの動作の解釈をもっと詳細に規定する必要があります。**

RUNTEST TCK

多くのATE**バウンダリ・スキャンツールメーカは、テスト・ベクタ・ファイルのサイズが非常に大きくなってしまうため、**TCK**のバースト的な動作の発生を好みません。ファイル・サイズが大きくなると、テストの全体コストが増えて、ベクタ・セットの動作効率も低下します。**RUNTEST**レコードは何かの発生を待つときに使用するため、多くの**ATE**メーカは指定された**TCK**バーストを時間値として解釈します。**

数値はμsで表した待ち時間であるという解釈が好まれています。これが、EZTagの生成したSVFファイルに対する解釈を表しています。

SDRの予測TDO値

SVF仕様では、TDOの予測値を指定する方法を規定していますが、TDO値が期待値に一致しない場合に何をすべきかは規定していません。

ザイリンクスXC9500部品を使用する場合には、TDOの予測値は直前に完了した動作(例えば消去動作またはプログラム動作)のステータスを反映しています。ステータスが予測通りのステータス(SVFファイルの生成時にTDO値として予測した値)でない場合、次に示す1149.1 TAPコントローラの状態遷移シーケンスを使用してください(EXIT1-DRステートでTDOの評価エラーが検出された場合を想定しています)。

1. EXIT1-DR
2. PAUSE-DR
3. EXIT2-DR
4. SHIFT-DR
5. EXIT1-DR
6. UPDATE-DR
7. RUN-TEST/IDLE

上記の状態遷移シーケンスを図2の1149.1 TAP状態図に示します。

この状態遷移シーケンスの本当の意味は、シフト入力されたばかりのプログラミング・データまたは消去データを無効にして、前のプログラム・データまたは消去データを再使用することです。SVFを解釈しているATEアプリケーションは正しく動作しており、従って現在のSVFレコードを超えて先に進まないことを確認する必要があることに注意してください。

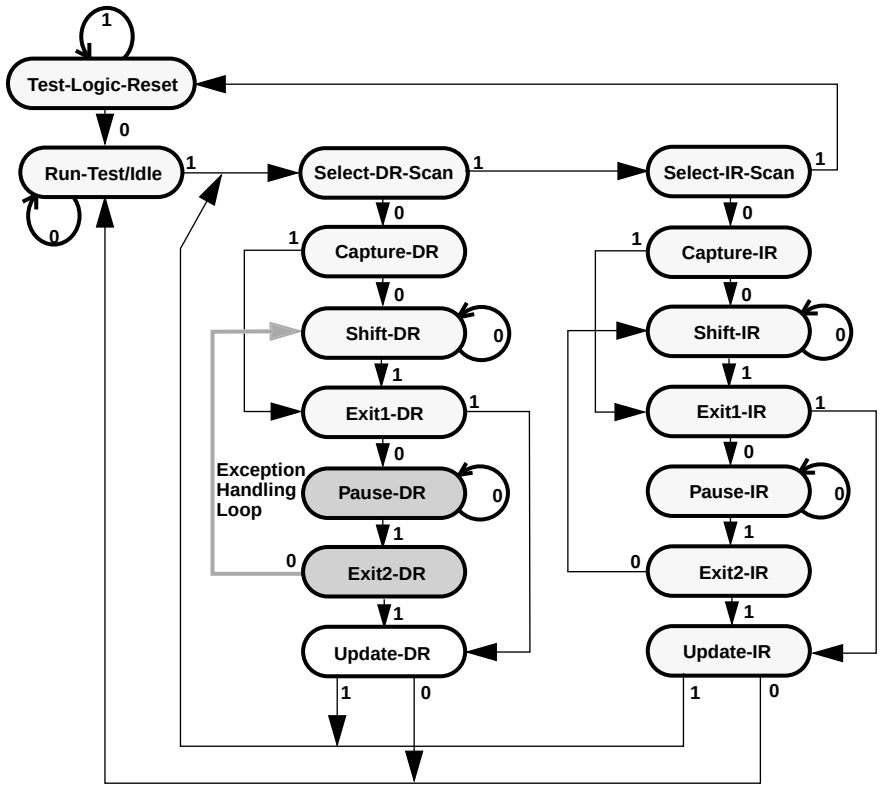

図2:テスト・アクセス・ポートの状態図

```
// First SDR record
SDR 27 TDI (000003fe) SMASK (07ffffff); // Just apply the value - no test for
TDO
RUNTEST 160000 TCK; // Wait for 160 msec.
// Second SDR record
SDR 27 TDI (008003fe) SMASK (07ffffff) TDO (00000003) MASK (00000003);
// Apply value to TDI read TDO test for concurrence
// if not as expected do “failure recovery loop” - hold
// at this SDR instruction.
RUNTEST 160000 TCK; // Wait for 160 msec
// Third SDR record
SDR 27 TDI (010003fe) SMASK (07ffffff) TDO (00000003) MASK (00000003);
RUNTEST 160000 TCK; // Wait for 160 msec
// Fourth SDR record
SDR 27 TDI (018003fe) SMASK (07ffffff) TDO (00000003) MASK (00000003);
RUNTEST 160000 TCK; // Wait for 160 msec
```


図 3: ATE **フローを説明する** SVF **ファイル(ファイルの一部分)**

図3**に示す**SVF**ファイルを例として使用した場合、必要とされる** ATE **動作は次のようになります。**

1. **指定の**TDO**を使用して、**SDR**インストラクションを読み出すとき(図**3**の**2**番目のように)、**TDO**の予測値はテスタ上のデバイスの出力値と一致する必要があります。一致しない場合は、エラーの**recovery loop**が実行されます。**RUN-TEST/IDLE**ステートでは、前に入力された**RUNTEST**インストラクションで指定された時間だけ停止が挿入されます。**
2. RUNTEST**インストラクションの終了時に、同じ**SDR**レコード(この場合は、ファイル内の**2**番目のもの)を再使用して**TDO**値を再度テストします。**
3. TDO**が予測値と一致する場合は、**TAP**ステート・マシンは通常に**RUN-TEST/IDLE**ステートに戻り(**EXIT1-DR**と** UPDATE-DR**を経由して)、次の**SDR**レコードが使用されます。**
4. **この**"recovery loop"**は**32**回までしか実行されません。**TDO**値が**32**回とも不一致の場合は、その部品は故障していると見なされ、ユーザに対する故障表示をして処理をアボートします。**

通常、TDO**チェックでは**1%**以下のアドレスが不合格になり、エラーの**recovery loop**の実行に対応する消去時間またはプログラム時間の追加が必要になります。**

SVF ベース ISP 用疑似コード アルゴリズム

次の疑似コードは、一般的なSVF**プロセッサー**(ATE**もしくはバウンダリ・スキャン開発ツール)上で**SVF**ファイルを解決するときに使用される動作シーケンスを規定しています。**

1. Go to Test-Logic-Reset state
2. Go to Run-Test Idle state
3. Read SVF record
4. if SIR record then
   go to Shift-IR state
   Scan in <TDI value>
5. else if SDR record then
   set <repeat count> to 0
   store <TDI value> as <current TDI value>
   store <TDO value> as <current TDO value>
6. go to Shift-DR state
   scan in <current TDI value>
   if <current TDO value> is specified then
      if <current TDO value> does not equal <actual TDO value> then
         if <repeat count> > 32 then
            LOG ERROR
            go to Run-Test Idle state
            go to Step 3
         end if
         go to Pause-DR
         go to Exit2-DR
         go to Shift-DR
         go to Exit1-DR
         go to Update-DR
         go to Run-Test/Idle
         increment <repeat count> by 1
         pause <current pause time> microseconds
         go to Step 6)
      end if
      else
      go to Run-Test Idle state
      go to Step 3
   endif
7. else if RUNTEST record then
   pause tester for <TCK value> microseconds
   store <TCK value> as <current pause time>
   end if

まとめ

EZTagの生成したSVFファイルを使用すると、自動テスト装置とサードパーティ提供のバウンダリ・スキャンツール上でのXC9500部品のプログラミングが可能になり、製造フローを能率的にすることができます。これにより、装置製造工程のプログラム・ステップとテスト・ステップを統合することが可能になります。この統合により、装置の歩留まりと生産性が向上し、さらに部品の在庫管理が簡単化されます。

Reference

Serial Vector Format Specification, Rev C., Texas Instruments.

The Boundary-Scan Handbook, Kenneth Parker, Klewer Academic Publishers, 1994.

IEEE Standard Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1-1990 (including IEEE Std 1149.1a-1993)