

4 J-8

ASIC用テストパターン言語の提案

牛久保 政憲, 辺保 久, 村上 道朗

沖電気工業株式会社 超LSI開発センタ

1. はじめに

ASICにおける論理検証用のテストパターンは、人手で設計されており、その際、記述性の優れた、コンパクトな記述が可能なテストパターン言語が要求される。現在、このような標準となるテストパターン言語は存在しない。そこで、我々は、標準となるテストパターン言語 (Test Pattern Language, 以下、TPLと呼ぶ。) を開発した。

本稿では、TPLの特徴、機能について述べ、TPLを標準のテストパターン言語として提案する。

2. TPLの特徴

TPLには、以下の特徴がある。

- ・ タイミングジェネレータの記述とパターンの記述を明確に分けている。
- ・ 回路の機能単位にパターンの記述 (任意のピンのみ考慮してパターンの記述) が可能である。
- ・ パターンの記述として、縦、横時間軸の方法がある。
- ・ パターンの圧縮表現を用意している。
- ・ テスト容易化設計を考慮して、スキヤン向きパターンが表現可能である。

3. TPLの概要

3. 1. TPLの構造

TPLの構造を図1に示す。

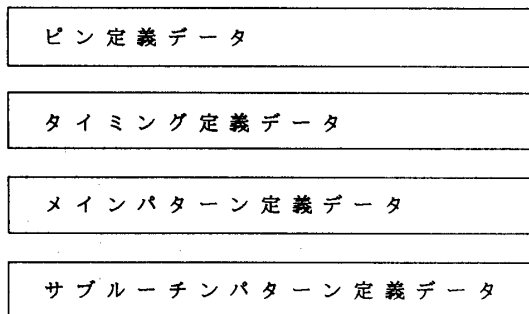


図1. TPLの構造

Suggestion of Test Pattern Language for ASIC
Masanori USHIKUBO, Hisashi HEMBO, Michio MURAKAMI
OKI Electric industry Co., Ltd.

3. 2. ピン定義データ

キーワードPINBLOCKとPINEND内に、全ピンのピン名及びそのピン種別を定義する。記述例を図2に示す。

```
PINBLOCK;
INPUT = IN1, IN2, IN3, SIN, SCLK, STST;
                                     入力ピン名
INOUT = I01, I02;
                                     入出力共用ピン名
OUTPUT = OUT1, OUT2, SOUT;
                                     出力ピン名
```

PINEND;

図2. ピン定義データの例

3. 3. タイミング定義データ

タイミングジェネレータ (テスト周期, 入力タイミング, 出力ストロブ) を定義する。記述例を図3に示す。キーワードTIMEBLOCKとTIMEEND内に定義し、複数のタイミングジェネレータを定義できる。

```
TIMEBLOCK TIME1;
                                     タイミング定義名
CYCLE = 1000N;
                                     テスト周期
IN1, IN2 = IN(100N, NRZ);
IN3      = IN(300N, 300N, RZ);
SCLK     = IN(300N, 400N, RZ);
                                     入力タイミング定義
OUT1, OUT2 = OUT(900N);
SOUT     = OUT(850N);
                                     出力ストロブ定義
```

TIMEEND;

TIMEBLOCK TIME2;CYCLE = 800N;IN3 = IN(300N, 300N, RZ);I01, I02 = IN(100N, NRZ), OUT(700N);

TIMEEND;

図3. タイミング定義データの例

3. 4. パターン定義データ

信号値を定義する。信号値として、以下の値を用意している。

- 0 : 信号値0入力
- 1 : 信号値1入力
- L : 信号値0期待
- H : 信号値1期待
- X : 不定値
- Z : ハイインピーダンス
- : 不変

パターン定義データには、以下の2種がある。

- ・メインパターン定義データ
- ・サブルーチンパターン定義データ

メインパターン定義データは、キーワードPATBLOCKとPATEND内に定義し、サブルーチンパターン定義データは、キーワードSUBPATBLOCKとSUBPATEND内に定義する。また、メインパターン、サブルーチンパターン定義データは、同じフォーマットであり、共に、複数定義可能である。

メインパターン定義データの例を図4に示す。また、サブルーチンパターン定義データを使用した例を、図5に示す。

パターン定義データは、以下の機能を有する。

- (1) 回路の機能単位にパターンの表現
PATBLOCKには任意のピンの信号値を定義可能である。したがって、PATBLOCK単位に回路の機能単位のテストパターンを記述できる。
- (2) 縦、横時間軸書きによる表現
縦、横時間軸書きのフォーマットを持っている。理由は、縦書き、横書きには、データ入力容易性の観点から、優位差があるからである。横書きは、既にテストパターンを机上で整理してある場合、データ入力が容易である。一方、縦書きは、テストパターンを設計しながら（考えながら）入力する場合、データ入力が容易である（図4参照）。
- (3) パターンの繰り返し表現
パターンをコンパクトに記述できる（図4参照）。
- (4) 信号値の16進表現
パターンをコンパクトに記述できる。
- (5) サブルーチンパターン
パターンをサブルーチン化できる。これにより、既成のパターンを再利用できる（図5参照）。
- (6) スキャン向きパターン
横時間軸書きのフォーマットで記述したパターンをスキャン用パターンに利用できる（図5参照）。

4. おわりに

標準テストパターン言語として、TPLを提案した。TPLにより、テストパターンの作成が効率よく行える。また、現在、TPLは弊社

```
PATBLOCK PAT/IN1,IN2,IN3,I01,OUT1;
      パターン定義名      信号値を定義する
/* III I 0                ピンの並び
   NNN O U
   123 1 T                コメント
      1 */
IC 010 0 X /<TIME1>;
C 1 L /;      使用するタイミングを定義
C 0 H /LOOP START 10;
C 0 /LOOP END;
C 11 /<TIME2>;
PATEND;
      縦時間軸書き
      10回繰り返し返し
```

```
PATBLOCK PAT ;
TIMING=1C/<TIME1>,23C/<TIME2>;
      使用するタイミングを定義
IN1=01(20:0)1;
IN2=11(10:10)1;
IN3=(23:0);
I01=0(22:L);
OUT1=XX(21:H);
PATEND;
      横時間軸書き
      信号値Hを21回繰り返し返す
```

※同じテストパターンを縦書き、横書きで記述した例

図4. パターン定義データの例

```
PATBLOCK PAT1/IN1,IN2,IN3,I01,I02,
      OUT1,OUT2,SCLK,STST;
IC 011 XX XX 00/<TIME1>;
      ↓
C 100 LL LL 00 /;
      $CALL SUBPAT ; → サブルーチン
                        パターンの参照
      ↓
C 101 XX XX 11 /;
      $CALL SCANPAT;
PATEND;
SUBPATBLOCK SUBPAT/IN1,IN2,I01,I02;
      サブルーチンパターン定義名
C 01 1X /<TIME1>;
C 1 L /;
C 0 H /LOOP 15 ;
C 1 L /;
SUBPATEND;
SUBPATBLOCK SCANPAT ;
TIMING=1C/<TIME1>;
SIN =(5:01)1001101100;
SOUT =(10:X)(5:LH);
SUBPATEND;
      SIN,SOUTにシリアル入力、出力
```

図5. サブルーチン、スキャンパターンの例

の標準テストパターン言語として使用されており、TPL(一部の機能を除く)を入力として、各種シミュレータ、各種テスター言語への変換が可能となっている。