

VHDL- 5

備忘録編 アトリビュートについて

この文書の目的

これまでに一通りの VHDL / V s y s 環境について解説してきましたが、今回は VHDL 言語仕様としては比較的漠然とした規定となっており、デバイス側フィッタ / E D I F リーダー側で受け取られた時に、実デバイスへの実現方法を制御するパラメータとなる「アトリビュート」について、現時点でわかる範囲で解説しておきます。

アトリビュートの例としては、ピン / ピン配列の番号指定、プルアップ / ダウン抵抗のオン・オフをコントロールするものが代表的ですが、これらの内容は果たして VHDL のソース文中に記述すべきものか、あるいはコンパイル時に VHDL ソースとは別に T c l など与える指示記述とするべきものか、は議論の分かれる処だろうと思います。現時点での私なりの判断では「使うほうの技術レベルに依存するが、どちらでも良いのではないか」と思っています。

基本的なアトリビュートの記述方法

は、階層トップレベルのエンティティ記述部分の最後で、

```
type string_array is
    array(natural range<>,natural range<>)of character;
attribute pin_number: string;
attribute array_pin_number: string_array;
```

のように、そもそもの文字列 `string_array` 自体を最初に定義して、実際にフィッタ環境に渡す文字列 `pin_number` / `array_pin_number` を、あらかじめ「これは文字列として扱ってくださいよ」と宣言しておいてから、その内容を (VHDL では `string` は基本的な定義済キーワードとなっていますので、ここで新規に定義するのはその配列への拡張 `string_array` だけです) 以下のように記述します。

```
attribute pin_number of P_HIOCLK: signal is "156";
attribute array_pin_number of P_MODEIN: signal is
    ("168","167","166","165","163" );
```

上の例で、`P_HIOCLK`、`P_MODEIN` は、エンティティ部分で定義してある信号ポートの名前をつかいます。

フィッターに渡されるダブルクォテーションで囲まれた文字列の形式は、ターゲット側の仕様に依存します。アクテルがターゲットの場合は `"156"` ですが、ザイリンクスの場合には `"P156"` となりますし、アルテラの場合には別途チップ識別子が必要となります。

配列の並び順は、エンティティ部の該当信号ポートの配列順に従って処理されるようです。かつてのバージョンではこれが正しく処理されないケースがありましたので、実用時には予め小さな設計で「実際のフィッターのバージョンについて」確認しておくほうが良いでしょう。

ザイリンクスの場合、デバイスのピンに内部プルアップ抵抗などを付けて使用したい場合があります。もしこれを VHDL ソースから直接指示したい場合には以下のように記述します

```
attribute PULL : string;  
attribute PULL of P_XBUS1 : signal is "PULLUP";
```

これらの文字列形式でフィッターに渡されるアトリビュートのほかに、論理値として渡すようにしなくてはならないパラメータもあります。例えば I/O ピンの動作モードを指定するザイリンクスの例ですと下のようになります。

```
attribute FAST : boolean;
```

まとめ

アトリビュートを FPGA への利用面だけから見ると、基本的にはフィッター側に別途処理させる(場合によっては別途処理すべき)内容のパラメータ引き渡しに関するものが主体で、これを VHDL ソースで記述する場合の主な目的は、私の場合には文書作成 / 交換の時の説明負担を簡便にする為です。

個別のアトリビュートについての詳細仕様は、デバイスメーカーのサポートに照会されるよう(エグゼンプラーのサポートでは TCL / GUI からの設定方法を基本的な手順として奨励していますので、開発設計チームの全員がソースコードの管理方法に習熟した後であれば、私も TCL で処理する方法のほうが良いと思うかも知れません) お勧めします。