



-- 設計資産をそのまま合成用ライブラリに登録

設計再利用のインフラとなる 高位合成手法を提案(後編)

Daniel D. Gajski/Viraphol Chaiyakul/Allen C. H. Wu

†マークのついた単語については、pp. 117-118の「用語の説明」を参照して下さい。

演算器やレジスタよりも複雑な大規模回路ブロックをアロケートできるハイレベル合成ツールを紹介する。筆者らはこのツールを「IPセントリック合成ツール」と呼んでいる。人手で設計した回路ブロックだけでなく、このツールで開発したRTLモデルも合成用ライブラリに登録できる。こうしたツールの利用が進むと、自動的に設計再利用のインフラが整備されるという。なお、本記事の前編(1節~3節、図1~図4を含む)は、DESIGN WAVE MAGAZINE No. 19, pp. 106-112に掲載されている。合わせて参照していただきたい。(編集部)

4. ビヘイビア合成より高位のモデルを扱えるツール

ここでは、3節で解説したIPセントリック設計手法とその技術要求に基づいて筆者らが開発したIPセントリック合成ツール「Explorations Environment(XE)」を紹介する。これは、社内開発のコアや外部から調達したコアを含むさまざまなコンポーネントを活用しながら、3.1節で説明したモデルを、3.2節の各アーキテクチャにマッピングするハイレベル合成ツールである^{注1}。

4.1 大規模マクロ対応の高位合成

5種類すべてのモデルを扱える

既存の論理合成ツールやビヘイビア合成ツールと異なり、筆者らが開発したツールは3.1節で説明した5種類のすべてのモデルを扱える。すなわちSFSMDモデルとコンカレントSFSMDモデル[†]を扱える点が新しい。これらの入力モデルはVHDLで記述する。

IPセントリック合成ツールはSFSMDモデルやコンカレントSFSMDモデルを、それぞれIPセントリック・プロセッサやIPセントリック・システム[†]にマッピングする。そして、市販の論理合成ツールに入力できるRTLモデルを出力する。

さらにユニークな特徴は、データベースにある。3.3節で説明したすべてのコンポーネント(設計に利用するさまざまなマクロ・セルやコア)を使用するために必要なプロトコルを暗号化して格納している。

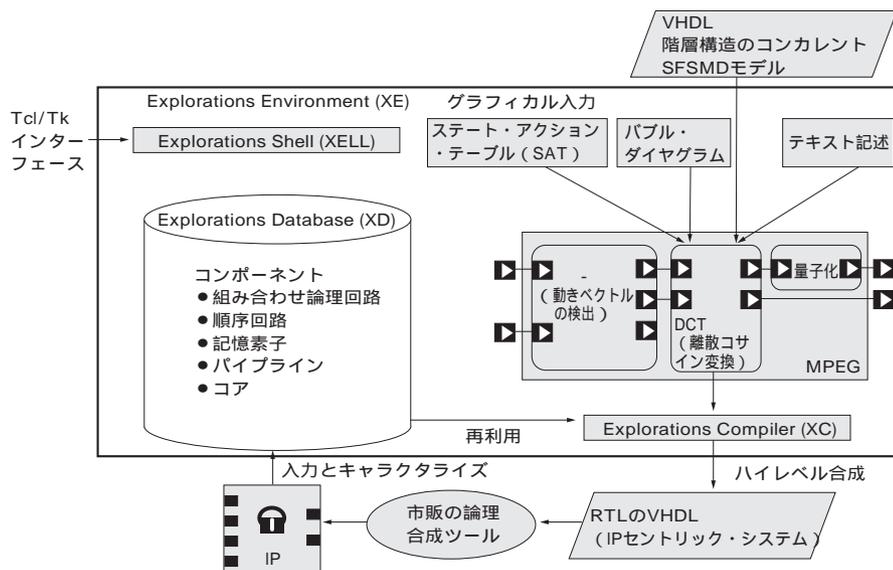
また、VHDLからRTLモデルを自動合成する機能に加えて、設計者がモデルを対話的に修正したり、微調整する環境も備えている。

三つのサブシステムからなる

IPセントリック合成ツールの構造を図5に示す。IPセントリック合成ツールは、以下の三つのサブシステムからなる。

① Explorations Compiler(XC)

Explorations Compilerは、入力としてHDL記述を取り込む。これをRTL記



【図5】 Explorations Environment

設計再利用を考慮した合成環境「Explorations Environment(XE)」の構造を示している。XEは三つのサブシステムからなる。すなわち、① Explorations Compiler(XC)、② Explorations Database(XD)、③ Explorations Shell(XELL)である。

注1: IPセントリック合成ツール「Explorations Environment」は、米国のY Explorations社が製品化している。同社のツールは、日本ではソリトンシステムズが販売する。

```

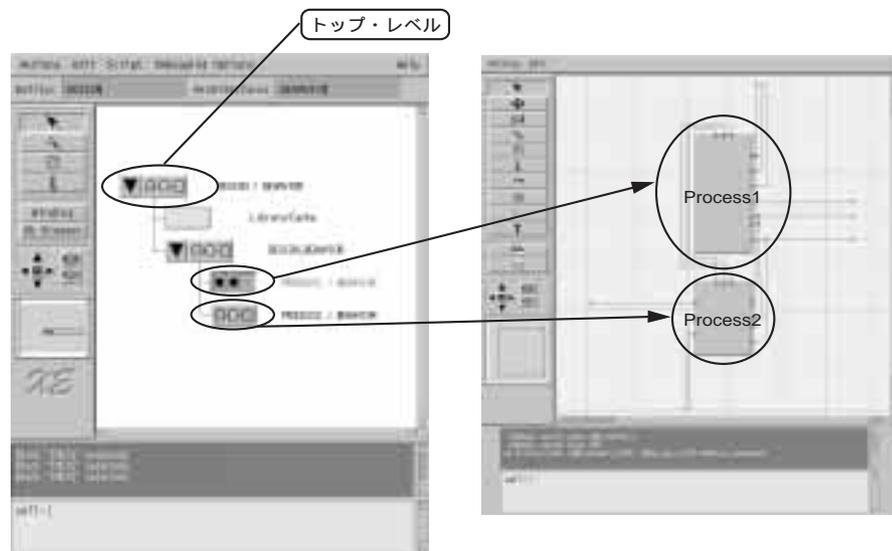
entity Process1 is port(...);
architecture behavior of Process1 is
begin
  process
    variable Tem1,.....
  begin
    Tem1:=A+B+C;
    Tem2:=C+D;
    If(cond='1')then
      Y<=A*C;
    end if;
    .....
  end;
entity Process2 is port(...);
architecture behavior of Process2 is
.....

entity Design is port(...);
architecture Behavior of Design is
....
begin
  Inst1:Process1(...);
  Inst2:Process2(...);
end

```

【図6】
VHDLの入力記述の例

IPセントリック合成ツールの入力には抽象度の高い設計記述である(現在はVHDL)。入力記述を、[図3](#)前号を参照)に示した五つのモデルのいずれかに変換して取り込む。さまざまな抽象度のモデルや階層構造を扱える。



【図7】ブロック図の表示画面

IPセントリック合成ツールはシステム・レベルのHDL記述を取り込んだり、モデル化する。さまざまな抽象度の表現に対応している。もっとも抽象度の高い表現はブロック図である。ブロック図によって、SFSMDモデル([図3\(d\)](#))を並列に、あるいは階層的に表現する。[図7](#)は、[図6](#)のVHDL記述をもとに自動生成したものである。

述やハード・コア、ソフト・コアなどの混在したIPセントリック・プロセッサ型アーキテクチャに変換する。Explorations Compilerは、コンポーネントのアロケーション(割り付け)、スケジューリング、入力/出力インターフェースの合成とい

た処理を行う。そして、Explorations Databaseに登録されたコンポーネントを使って最適な回路を合成する。

② Explorations Database(XD)

Explorations Databaseはデータベース・エンジンである。内部情報を秘匿(ブ

ロックボックス化)したままコアやインターフェース・プロトコルのデータを取り込んだり、キャラクタライズしたり、検証できる。

③ Explorations Shell(XELL)

Explorations ShellはTcl/TkをベースにしたAPIルーチンである。ドキュメンテーションや設計データのバージョン管理などのためのスクリプトを生成することができる。

記述をブロックやバブルに変換

IPセントリック合成ツールの入力記述はビヘイビア・コードである(現在はVHDLのみ)。3.1節で解説した5種類のモデルを利用して、合成したい回路の動作を記述する。異なる抽象度の記述を混在させたり、階層構造の記述を利用してもかまわない。

たとえば[図6](#)は、二つのビヘイビア・コードのサブブロックからなる設計データである。IPセントリック合成ツールはまずVHDL記述を解析し、[図7](#)のようなブロック図を生成する。各ブロックはビヘイビア・レベルのVHDL記述を含んでいる。つまり一つのブロックが一つのコンポーネント、あるいは一つの構造レベルのVHDL記述に対応する。

各ブロックに含まれるVHDL記述は、一つのプロセスを表現している。このプロセスはコンパイルによってバブル・ダイアグラム(一種の状態遷移図)に変換される。それぞれのバブルはSFSMD(スーパー状態FSMD)のスーパー状態に対応する([前号の図3\(d\)](#)を参照)。[図8](#)は、[図6](#)の「Process1」の記述から生成されたSFSMD表現である。

テーブルに合成結果を表示

ブロック図やバブル・ダイアグラムの生成が完了すると、次は合成の作業に入る。アロケーションとスケジューリングによって、各バブル(スーパー状態)をクロック・サイクルに対応したコントロール・ステップに分割していく。この過程は、既存のビヘイビア合成ツールの考え方とよく似ている。ただし、ビヘイビ