

ハードウェア仕様設計エキスパートシステム

3R-2

— 仕様記述から動作記述への展開について —

打橋 知孝、仲西 秀基、中村 行宏
(NTT 電気通信研究所)

1. はじめに

近年、半導体技術の急速な進歩に伴い論理装置の規模の増大化、複雑化が進んでおり、設計工数の削減が重要な課題となっている。これに対処するため、設計工程の計算機支援についての研究が行われている。機能・動作設計工程以後の計算機支援については多くの成果が発表されている[1]が、より高位である仕様設計工程の支援についての発表は少ない。

我々は、設計工程のより高位のレベルからの計算機支援を目的として研究を行っており、既に機能・動作設計工程の上位のレベルである仕様設計工程の支援を行う仕様記述システムを発表している[2]。

本稿では、仕様記述から機能・動作記述(SFL[3])への知的展開方法について述べる。

2. 展開の概要

本システムの処理の流れを図1に示す。

入力である仕様記述では、論理装置中の記述の対象となるオブジェクト(プロセス、データなど:図2)、及びそれらの間の(リレーション)データの流れ・制御の流れ・構成の関係を記述する。また、出力となるSFLによる動作記述では使用したいハードウェア要素(機能回路、モジュール、レジスタなど)を定義し、それらのクロック・条件毎の動作を記述する。

システムの処理は大きく次の3つに分けられる。

オブジェクト分類:仕様記述のリレーションのパターンに基づく分類ルールを用いて入力を解析し、SFLのハードウェア要素への対応付けを行う。

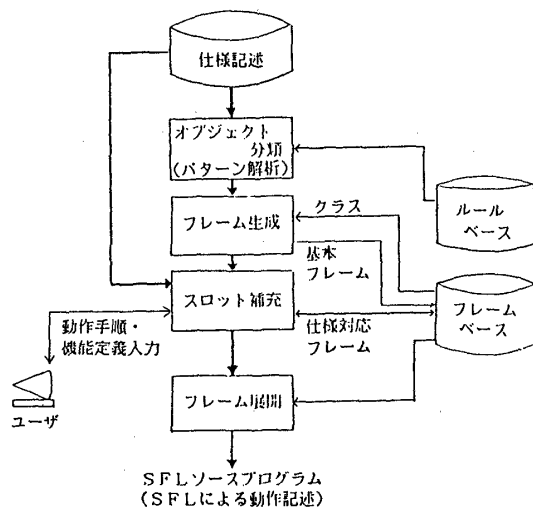


図1. 展開の概要

フレーム生成 : SFLのハードウェア要素毎に定められている記述形式に即したフレームに対応づけ、SFL記述に必要な情報を人力の仕様記述から得る。

フレーム展開 : 生成されたフレームからSFL記述を生成する。

3. オブジェクト分類(パターン解析)

仕様記述のオブジェクト間のリレーションのパターンに着目・解析し、仕様記述中の各プロセスとSFL記述中のモジュール/ステージ/レジスタ/機能回路との対応をつける。

対応付けのルールは、μPLANNERで書かれており、構造/データの流れ/制御の流れに関するものを基本とし、これらの組み合わせによってプロセスの分類を行っている。表1にルールの分類を示す。

機能回路を例に分類の考え方を以下に示す。

機能回路の特徴は、以下の4つである。

- ①データを持つ。
- ②データの加工を行う
- ③他のプロセスから使用される。
- ④他のプロセスを制御しない。

故に、あるオブジェクトがそれぞれに対応する記述

- ①～を保持する、
～を受け取る/送られる
- ②～にする、
～を更新する
- ③～から利用される
を備えており、かつ
- ④～を利用する、
～を起動/終了する、
～を真/偽にする

を備えていなければ、そのオブジェクトは機能回路であると同定する。

プロセス: 論理装置内の処理の単位。動作の中の一つ一つの処理から複数の処理の集合までプロセスとする。

データ: 論理装置の動作の中で操作される情報を示す。

図2. オブジェクトの種類

表1. ルールの分類

プロセスの分類	プロセスが機能回路に対応するか プロセスがレジスタに対応するか プロセスがモジュールに対応するか プロセスがステージに対応するか
構造	オブジェクトが下位のオブジェクトを持つか オブジェクトが構造の最下位であるか
データの分類	データがプロセスの中から外へ流れるか データがプロセスの外から中へ流れるか
条件、事象の分類	条件、事象がデータの操作に影響するか 条件、事象がプロセスから影響を受けるか

図3の仕様記述の例で演算器に着目すると、演算器は命令セットから利用され(③)、メモリデータを送られ(①)、演算結果を更新し(②)、かつ④に示した記述がないので、機能回路に該当している。

4. フレーム生成

分類されたモジュール/ステージ/レジスタ/機能回路からSFLの記述へ展開するために必要な情報をスロットとして持つフレームを生成する。

フレームベースにはSFLの記述形式に対応したフレーム(クラス)が用意されており、パターン解析によって得られた情報を基にフレーム(インスタンス)が生成される。

生成されるフレームにはSFLの記述形式に対応し、プロセス毎に与えられるもの(基本フレーム)と仕様記述のリレーションに対応し、オブジェクト間のデータの流れ、制御の流れを示すもの(仕様対応フレーム)がある。

図3の例において、演算器は機能回路であったので、図4.aに示す機能回路のSFL記述形式に対応するフレーム(クラス:図4.bのFUNCTIONAL-COMPONENT)に対応づけしたインスタンスが生成される。

4.1 スロット補充

分類されたプロセスに対応する基本フレームのスロットを埋めるにあたり、仕様記述では陽に記述されていないがSFL記述では明示されなければならないものがある(例、端子)。

機能回路、モジュールの境界は必ず端子によって定義されるので、データの流れを解析し、機能回路・モジュールの境界を越えるデータの流れを端子に対応づけることにする。

この際、基本フレームのスロットである仕様対応フレーム(データの流れを示す)がそのプロセスに対するデータの入力であるか、データの出力であるかを捜し出し、入出力端子に当てはめる。

図3の例において、演算器に関係するデータの流れは、

①"ANDXはアキュムレータの内容とメモリデータを演算器に送る",

演算器は、ADDとANDから成り、
命令セットによって利用される。

演算器は、演算結果を更新する。

ANDXは、演算器とインデックスレジスタとアキュムレータとメモリインタフェースを利用し、
インデックスレジスタの内容をメモリインタフェースへ送り、
アキュムレータの内容とメモリデータを演算器へ送り、
演算器の結果をアキュムレータへ送る。

メモリデータは、アキュムレータとインデックスレジスタと演算器に送られ、
bit長が8である。

演算結果はbit長が8である。

図3. 仕様記述の例

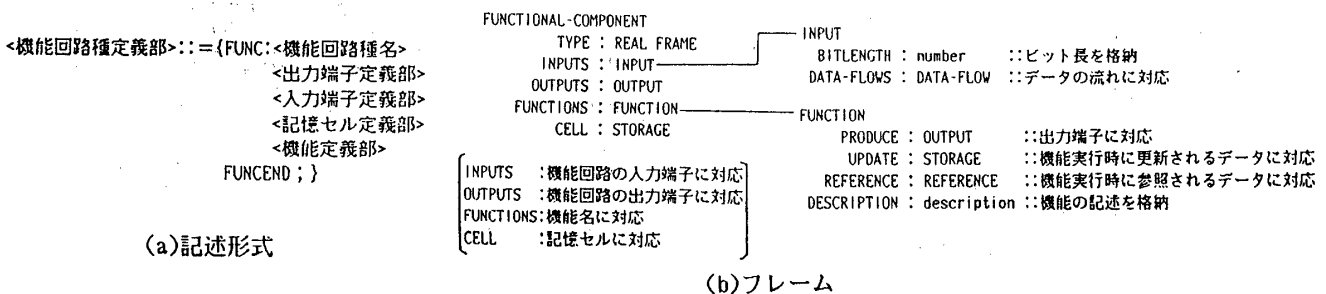


図4. SFL記述形式とフレーム(機能回路の例)

②"ANDXは演算器の結果をアキュムレータに送る",

③"メモリデータは、演算器に送られ、bit長が8である",

④"演算結果は、bit長が8である",

がある。ここで①は演算器に対するデータの出力、②はデータの出力、③、④は各bit長を示す。これにより図4に示した仕様対応フレームINPUTのスロット(BITLENGTH, DATA-FLOWS)が埋まり、演算器に対応するフレームFUNCTIONAL-COMPONENTの端子を示すスロット(INPUTS)が埋まる。

4.2 動作手順・機能(演算)定義入力

端子と同様に仕様記述で陽に現れていなくて、SFL記述で陽に表現されなくてはならないステージの動作の手順、機能回路の機能についてはユーザに問い合わせて入力することになっている。

動作の手順については、ステージに関する動作の記述が列挙されるのでユーザはそれらに対して前後関係を与えることで手順を示す。

演算器の機能(AND, ADD)の記述はここで入力され、フレームFUNCTIONのスロットDESCRIPTIONが埋まる。

5. むすび

仕様記述で与えられたオブジェクト間の関係に着目し、ルールを適用することによりプロセスの分類を行い、それに対応したフレームを用いることによって仕様記述からSFLによる動作記述への展開の方法を提案した。

この方法により、設計工程の最上位のレベルである仕様設計から展開を行うことにより、ハードウェア設計の工数の削減及び、工程途中での誤り混入の防止を図ることができ

る。しかし、今回のルールはSFLの要素を多分に意識したものであるため、仕様記述に制約が加わったかたちになっている。

今後は、ルールの整備、充実を図り、現段階でサポートしていない動作の順序、機能回路の機能に対応するルールの検討を進めるとともにより一般的な展開ルールの選出をおこなっていく。

参考文献:

- [1]中村、他『An RTL Behavioral Description Based CAD System with Synthesis Capability』IFIP, CHDL'85, p64-78, 1985
- [2]仲西、他『ハードウェア向き仕様記述システム』情報処理学会第32回全国大会講演論文集、1986
- [3]中村、他『ハードウェア動作記述言語SFL』情報処理学会第29回全国大会講演論文集、1984