

## 仕様記述言語 LOTOS からのハードウェア設計について\*

2H-2

花屋雅貴、稲富 猛、西垣 正勝、佐藤 文明、水野 忠則†

静岡大学 理工学研究科‡

## 1 はじめに

近年、プロトコルの開発において仕様記述言語が用いられるようになってきている。しかしながら、その仕様をテストしたりする環境は完全に整っているとは言い難い。

また、ハードウェア開発環境においては、設計及び検証の能率化、様々なユーザレベルでハードウェアが理解できるように、ハードウェア記述言語 (Hardware Description Language: HDL) が用いられるようになってきた。

我々は仕様記述言語として ISO 標準の LOTOS を [1]、HDL として IEEE 標準の VHDL [2] を選び LOTOS から VHDL への自動変換を目的とする。それによりプロトコルのハードウェアによる実装の際、システムの動作確認及び開発コストの削減などにおいて有益であると考える。

本稿では、LOTOS から VHDL への変換のためのアルゴリズムについて提案する。

## 2 LOTOS と VHDL

## 2.1 LOTOS の概念

LOTOS は基本的に次の考え方に基づいて言語設定が行なわれている。

「システムは、そのシステムのやりとりする外部から観測可能なイベント間の時間順序を規定することで仕様化可能である。」

ここでイベントとは

1. 瞬時発生である
2. それ以上細分化できない
3. 他のイベントとの時間的重なりがない、つまり、ある時間に起こるイベントは唯一つに限定される。

ものであり、これをイベントのアトミック性という。この性質により LOTOS 記述は各プロセスのイベント木 (イベント列を木で表したもの) の関係により表せる。

また LOTOS はシステムをプロセスという実行単位で階層化して表現できる。

\*A hardware design method based on LOTOS description

†Hanaya Masataka, Inatomi Takeshi, Fumiaki Sato, Nishigaki Masakatsu, Tadanori Mizuno

‡Shizuoka University

## 2.2 LOTOS の同期

LOTOS における重要な意味を持つ概念として、イベントの同期というものがある。LOTOS の同期とは、通信する二つ以上のプロセスが、同時に、あるイベントの発生に関与するということである。つまり二つ以上の全てのプロセスにおいて同期すべきイベントが生起可能な場合において、はじめてそのイベントが起こりうる。またプロセス間の同期 (相互作用) はゲートにおいて起こる。

## 2.3 VHDL の特徴

VHDL はハードウェア記述言語の一つである。VHDL の特徴は、

- 階層的にハードウェアを記述できる
- 様々な抽象レベルで記述できる
- シミュレーションによるテストが可能である

などが挙げられる。具体的な記述としては、各ハードウェアコンポーネントの入出力ポートの指定を行うエンティティ宣言と、各コンポーネントの動作を規定するアーキテクチャ宣言からなる。階層化されたハードウェアを記述するには、アーキテクチャ宣言中で使用するコンポーネントを指定し、ポートマップ宣言でそれぞれの接続情報を記述する。

## 3 変換手順

我々の変換アルゴリズムでは従来のもの [3] と異なり、LOTOS の階層構造をそのままハードウェアの構造にできることが特徴である。

変換手順としては、(1) LOTOS 記述された仕様の分割、(2) VHDL コードの生成、の二つに分かれる。

(1) では、プロセス呼び出しによるモジュール分割と LOTOS オペレータによるモジュール分割を行う。そして、分割された各モジュール及びモジュール間の関係の情報を取り出す。(2) では (1) で取り出した情報に従い VHDL コードを生成する。まず、分割された各モジュールに対応するコンポーネントのエンティティ (インタフェイス) 宣言を行う。次に、各コンポーネントの動作をアーキテクチャ宣言で記述する。このとき上位階層では使用するコンポーネントの指定し、それらの接続関係を明示

する。一方、最下層ではイベント列を考慮した状態遷移記述を行う。

また、並列オペレータと選択オペレータは特別なハードウェア構造として実現する。以下にこの二つのオペレータの意味とハードウェアの実現方法、及び、最下層のイベント列のハードウェア化について示す。

### 3.1 並列オペレータの実現

LOTOS のオペレータにプロセス間の並列性を表すものがある。完全同期並列では二つ以上のプロセスで全てのイベントが同期する。一般同期並列では二つ以上のプロセスである指定されたイベントに関してのみ同期する。非同期並列では二つ以上のプロセスでその各々がお互いに関与せず独立に動作する。同期並列オペレータは「同期ゲート」モジュールを作成することで図1のように構成できる。



図1: 同期並列

非同期並列は二つ以上の独立したハードウェアとみなす。ここで述べた「同期ゲート」の仕様は以下である。

- 複数の要求を同時に受け付ける。
- 同期がとれるかを判定する。
- 判定結果をプロセスに返す。
- 複数の要求が同期可能な場合はランダムに一つを選ぶ。

### 3.2 選択オペレータの実現

また、複数のイベント及びプロセスが選択的に実行可能であることを表現するために、LOTOS には選択オペレータが存在する。

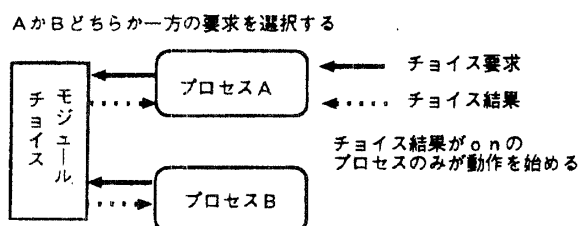


図2: 選択機能の実現

この選択オペレータの場合も「選択」ハードウェアを構成して図2のようなハードウェアとして実現する。こうすることで選択が階層化された場合でも改めてハードウェアを再構成する必要がなくなる。これは従来の HARPO [4]でも対応していない。

そしてプロセス呼び出しがされた場合においても、そのプロセスを一つのハードウェアモジュールとして実現し、LOTOS の階層構造をハードウェアに反映させる。

### 3.3 イベント記述の状態遷移記述

最初に述べたように LOTOS のシステムはイベント木として表せる。イベント木はイベント a のあとに b が起こるなどのイベント列を木構造で表現したものである。

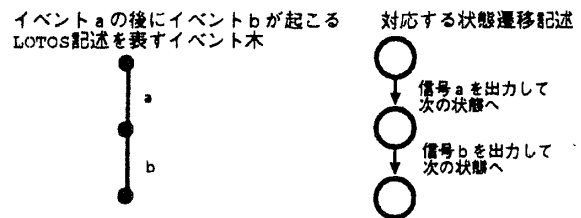


図3: イベント列と状態遷移記述

これは図3のように、ある状態で信号 a を出力し、次状態に移りそこで信号 b を出力するという状態遷移記述で表現できる。このような状態遷移記述は VHDL において記述可能である。

## 4 まとめ

LOTOS 記述の階層表現、選択表現をハードウェアで実現可能にするために「同期ゲート」及び「選択」ハードウェアを導入した。現在このアルゴリズムに基づき実装中である。今後はイベントデータとしてビット型以外のものを許すように拡張し、まだ扱っていない LOTOS オペレータに関して変換アルゴリズムを拡張していく。

## 参考文献

- [1] 高橋、神長著：仕様記述言語 LOTOS、カットシステム
- [2] Z・ナバビ著：VHDL の基礎、日経BP出版センター
- [3] 黄、安本、北道、東野、谷口：LOTOS 風言語で書かれた同期式順序回路の要求仕様記述と回路自動合成、マルチメディア通信と分散処理、66-9, (1994.7)
- [4] A.M.Lopes: HARPO, [gopher://sanson.dit.upm.es:70/00/pub/lotos/tools/HARPO](http://sanson.dit.upm.es:70/00/pub/lotos/tools/HARPO)