

## 論理回路設計環境の構想

6Q-1

進藤 達也

川戸 信明

(株)富士通研究所

## 1. はじめに

現在、論理設計者1人にEWS1台という作業形態が現実のものとなろうとしており、従来は机上で行われていた、設計対象を仕様として具体化していく創造的作業も計算機の支援を受けながら行えることが望ましい。そこで、EWS上に考える場として試行錯誤的作業の行いやすい環境を構築する事が重要であると考え、以下に論理回路設計環境として、その構想を述べる。

## 2. システム概要

レジスタトランスフェレレベルの設計(機能設計)において、設計者の抽象的イメージの具体化を対話的操作を通して支援する環境を考える。

設計者はイメージを断片的に入力し、本システムはシミュレーションの実行を通して設計者に詳細化するべき項目を提示する。このプロセスを繰り返す事で仕様の詳細化を進める。設計者により入力された機能は最終的にはDDL〔1〕などの標準的言語として出力される。(図1)

以上の操作を設計者の思考を妨げることなく行えるように、各自にあった設計法がとれるよう入力手段に柔軟性を持たせ、最初は細かな事柄を気にせずメモ的な入力が可能なようにする。また、入力された機能に曖昧さが残っていても対話的に実行を進められる記号シミュレータを装備する。以下の章で詳細を述べる。

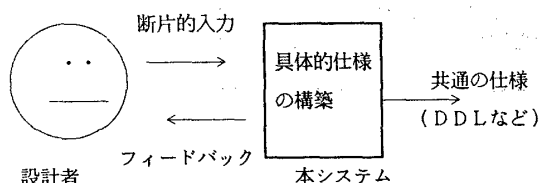


図1 論理回路設計環境

## 3. 機能記述用グラフィックエディタ

制御の詳細が決定しなくても入力できるような、機能記述的表現が可能なグラフィックエディタを用意する。対話的に設計を行うためには記述や変更のしやすさ、直感的な分かりやすさが重要となる。そこで言語による記述法は避け、状態遷移図と各状態毎にその動作内容を記述するデータ転送図を用いて仕様を表す。追加や削除の単位を1つの状態とする事により、実現したい機能を思いつくままに状態として登録できる。これらの機能がどのような条件で起動されるかは後で編集するようにし、断片的な入力を容易にしている。

## 4. 断片的入力とシミュレーション

設計者は頭の中で明確になっている部分から断片的に機能の入力を行うものとする。その為、シミュレーションの対象となるものは詳細化の途中にある回路であり、値が不定のデータを用いて演算せねばならなかったり、未定義の制御信号が現れたり、あるいは状態遷移において分岐先が未定義であるなどの事態にぶつかる事がある。この場合、記号的に処理できるものについては記号シミュレーション〔2〕を行い、不可能な場合はシミュレーションを一時停止して設計者の指示を仰ぐ。又、シミュレーションの実行につれて状態が遷移する様子や、モジュールの内部値の変化を刻々と表示するビジュアル・シミュレーション〔3〕の機能を持たせることにより、動作を分かり易く示す。

## 5. 任意の形状に設定可能なシンボル

データ転送図上に、部品として用いる既定義モジュールをシンボルとして記述する必要がある。この際に、シンボルの形状は任意の矩形に設定でき、また入出力端子の位置も固定されていない為、シンボルの辺上の任意の位置から信号線を引き出せる。又、ビジュアル・シミュレーションのために、モジュール内部の信号のうち何をシンボル上のどこに表示させるかは設計者が自由に設定できる。以上の操作は全て、マウスとポップアップメニューにより容易に行なえる。これにより、設計者は自分にとって分かりやすい図を用いて考えることができる。

## 6. インプリメンテーション

グラフィックエディタや記号シミュレータなど、設計者が直接触れる機能に重点をおいてプロトタイプを作成した。プロトタイプはSmalltalk-80上にインプリメントされ、図2に示すPartsBrowserを用いてエディットやシミュレーションを行う。上段の3つのビューにより階層的に示されたモジュールが操作の対象として下段の大きな2つのビューに表示される。下段のビューのうち左側が状態遷移図、右側がデータ転送図に対応する。状態遷移図において、状態名が白黒反転している状態が、現在選択されている状態を表している。データ転送図には、部品として使われるモジュールが書かれており、選択されている状態に対応する動作がモジュール間の配線として描かれる。この配線は、ある状態におけるデータの転送を意味するものであり、物理的結線と直接対応するわけではない。別の状態を選べば、現在描かれている配線は消え、新たに選んだ状態に対応する配線が表示される。

## 7. むすび

設計者が試行錯誤しながら機能仕様の詳細化を進めるための設計環境について考察し、プロトタイプを作成した。回路の自動合成技術が広く実用化された際には、その入力となる機能仕様を早期に決定する事が必要となり、ここで述べたようなシステムの意義は大きいと考える。今後の課題としては、設計者が面倒な細かい点を気にせず、より本質的な部分の設計に打ち込めるように、例外処理等に関する知識をシステムが持ち、これにより設計者から陽に示されていない情報を自動的に補って回路を構築する機能などの開発がある。

〔謝辞〕 日頃御指導を頂いている石井システム研究部長に感謝いたします。

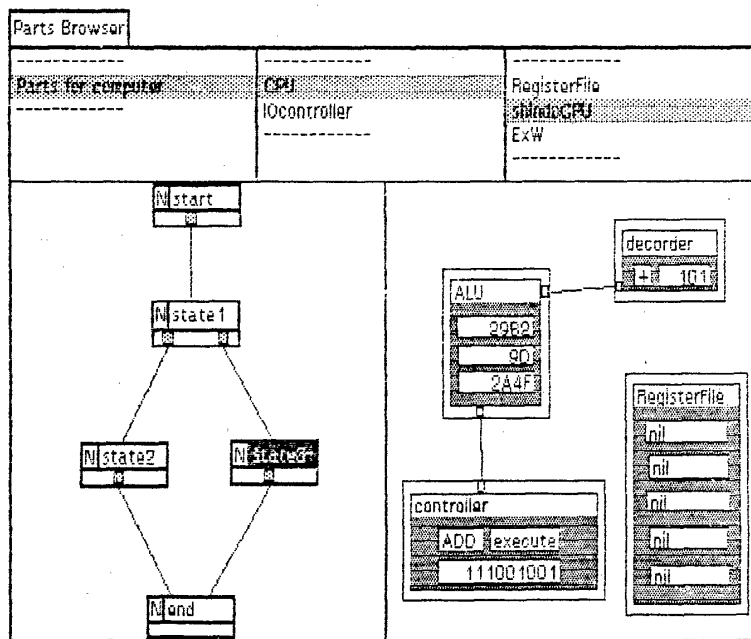


図2 インプリメント例

### 〔参考文献〕

- [1] 川戸, 斎藤, 上原: "ハードウェア設計言語DDLによる計算機支援システム", 情報処理, Vol. 21, No. 1, 1980.
- [2] 斎藤, 上原: "論理回路の記号シミュレーション", 情報処理, Vol. 26, No. 1, 1985.
- [3] 杉本, 阿部, 黒田, 加藤: "オブジェクト指向言語によるマイクロプログラム開発支援システム", 信学会, EC85-29, 1985.