

3R-3

## アーキテクチャ設計支援システムにおける並列処理とパイプライン処理

竹沢寿幸 上野聰 白井克彦

(早稲田大学理工学部)

### 1. はじめに

知識ベースに基づくアーキテクチャ設計支援システムの研究を続けている<sup>1)2)</sup>。回路の方式設計や機能設計を支援する上でまず問題となるのは、仕様の入力形式である。従来、設計者は自然言語により仕様記述を行い設計を進めていたと考えられる。しかし、CADシステムを開発するにあたりユーザ・インターフェイスとして自然言語を利用するには有効であろうが、仕様の入力として自然言語を直接利用するのは曖昧性を多く含むため必ずしも適切ではないであろう。

我々は、対象を汎用の回路ではなく特に信号処理に限定し、仕様の入力形式としてはアルゴリズムと入出力条件などの制約条件を選んだ。仕様記述とフロー解析については文献3)に譲り、本報告では並列処理やパイプライン処理の扱いを検討し、全体の処理概要とアーキテクチャを考慮した回路設計について述べる。

### 2. 処理概要と全体の制御

信号処理専用回路に限定しても多くの設計手法や設計例が考えられ、しかも、ユーザの制約に合わせて設計を行う必要がある。設計手法の要件として、ユーザの実行したい信号処理アルゴリズムは種類が多く、その上、制約条件が多種多様であるため、設計に関するストラテジの選択制御が適切に行われる事が重要である。そこに、知識工学の手法が利用できると考えられる。そこで、様々な設計手法を知識ベースに蓄えておき、その選択制御をプロダクション・システムで行うこととした。特に、フロー解析の結果をもとに、ストレージ割り当て、演算器割り当て、バス合成などの一連の手順を行う場合や、パイプラインのような高速化アーキテクチャを採用する場合、ユーザとインタラクティブに設計する、ユーザを中心とな

って設計する場合などの制御が重要である。回路の設計がうまくいかなかった場合の処理としては、プロダクション・ルールによる自動的なバックトラック機能が用意されている。なお、メタ・ルール的な制御も可能な枠組みとなっている。回路の表現方法は、プロダクション・システムがフレーム形式のデータとマッチングをとることが出来るため、フレーム表現とした。

### 3. フロー解析結果からの回路合成

まず、通常の回路設計手法を説明する。ストレージの割り当ては、変数のライフ・タイム解析をもとに、基本的には同時に生きていらない変数を同じストレージに割り当てる事ができるというものである。演算器の割り当ても、ストレージの場合と同様に、基本的には同時に動作しない演算を同じ演算器に割り当てる事ができると考えられるが、実際には演算の種類によって併合出来るものと出来ないものが有り得る。また、ユーザの制約、要求に余裕があれば、さらに併合することも有り得る。データ転送路についても同様にソースとデスティネーションを解析することによりバスを合成することが出来る。

本設計システムの場合、方式、機能設計を対象としているため、ALUとしては、複数の機能を行うものとのみ仮定する。従って、あるALUでは加算と減算を行うといった機能をまず決定するものとする。実際には、これはインストラクション・レベルに相当し、設計をさらに進めるにはレジスタ・トランスマッパー・レベルにおとす必要がある。

実際には、データバス系のみではなく、制御回路が必要である。制御回路としては、同期をとる機能とマイクロ・プログラムを仮定する。条件分岐やループの制御は、条件判断の分岐の制

御のみを行うものを仮定する。配列の初期化やバイオペラインの制御もこれに準ずるものとする。

回路の性能評価方法としては、アルゴリズムのデータフローがどの様なハードウェアにマッピングされているかをもとに、ユーザの制約を満足するかどうかを評価する。

#### 4. 例

本システムの実験例として、PARCOR格子型フィルタを取りあげ、いくつかの制約条件のもとで回路のアーキテクチャ設計を行った。仕様はフィルタのアルゴリズムと入出力条件として8ビット・シリアル、サンプリング周波数 12.5kHz、フィルタの段数12段、1フレーム当たりのサンプリング・ポイントを 128ポイントとした。データがサンプリング周期で与えられるすると、実時間で処理するためには  $80\ \mu\text{sec}$  以内で処理が終了する必要がある。

フロー解析の結果をもとに回路合成を行うと、図1のようになる。制約条件に余裕があるので、ストレージ、演算器、データ転送路の併合を行う設計手法を起動する。この場合演算の数が少ないので、ルールや構成要素の選択順序には依存しないといえる。

条件がさらに厳しい場合には、バイオペライン・アーキテクチャを選択せよというルールが与えられているため、その処理が起動される。このアルゴリズムは本来格子型のものであるので、それをフィルタの段数分カスケード状に並べることによって、バイオペラインが構成できる。このような構成をとることによってフィルタの段数分の演算が1度に可能となるので、この例では、処理時間は約12分の1となることが分かる。この場合の回路設計例を図2に示す。

このような高速化のアーキテクチャは、ユーザの要求が厳しい時に発動され、ユーザとインタラクティブに行うか、または、フィルタ等のテンプレートを用意し、バタンがマッチした時にそれを起動する。

#### 5. むすび

信号処理を対象としたアーキテクチャ設計支援システムについて報告した。実験システムは、東京大学大型計算機センターのUTILISP を用いて開発しており、早稲田大学リモート・データ・ステ

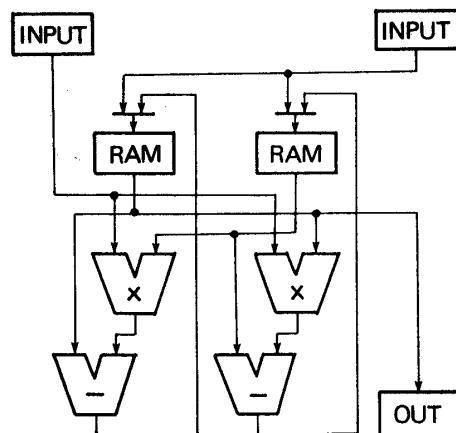


図1 PARCOR格子型フィルタ (データバス系主要部)

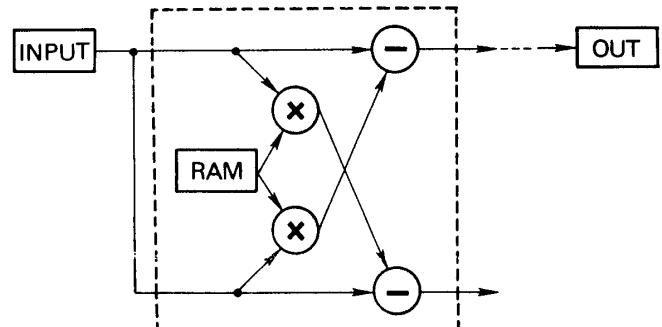


図2 バイオペライン構成例

ーションを利用して利用している。知識ベース中の知識が十分でない場合には、ユーザとインタラクティブに設計を進める必要があるので、マン・マシン・インターフェイスが重要である。今後は、フロー解析の結果や設計の途中経過を自動的に表示する機能を強化する必要がある。また、今回は主にデータバス系の演算器について検討を行った。ストレージやデータ転送路も演算速度や精度などの性能上重要なため、さらに検討を行う必要がある。

#### 【文 献】

- 1)竹沢、白井：“アーキテクチャ設計支援エキスパート・システムの構成と設計環境”，「VLSI CADへの知識工学の応用」シンポジウム, pp.1-10, (1986-1).
- 2)白井、竹沢：“VLSIアーキテクチャ設計支援における知識情報処理”，電気学会システム・制御研究会, SC-86-11, (1986-8).
- 3)上野、竹沢、白井：“回路自動設計のためのアルゴリズム記述とフロー解析”，情報処理学会第33回全国大会, 3R-4, (1986-10).