

# 高位の仕様記述に基づく 4X-7 専用プロセッサ設計支援システム

池永 剛    竹沢 寿幸    白井 克彦  
早稲田大学

### 1. はじめに

高位の仕様記述に基づく専用マイクロプロセッサ設計支援システムの開発を行なっている [1]。このシステムは高級言語で書かれた仕様記述を入力とし、そのアルゴリズムを効率よく実行するマイクロプロセッサの設計支援を行なうことを目的とする。

プロセッサ設計は、命令セット、内部構成、動作仕様等多くの点を考えなければならないが、それにはプロセッサに関する高度な専門知識が要求される。しかしながら、特定用途向けのプロセッサ設計においては非専門家でも設計可能な支援システムが強く望まれる。そこで、本システムでは汎用性を持つプロセッサの詳細設計を支援するのではなく、プロセッサ上で実行されるアルゴリズムに焦点を絞り、特定アルゴリズムを実行するのに適するプロセッサの設計を支援する。よって、プロセッサの枠組みはテンプレートとして用意し、それに対して入力アルゴリズムを解析し、変更を加えるという形をとる。つまり、ユーザは実行したいアルゴリズムを高級言語 (Pascal) を用いて、通常のソフトウェアを書くように入力すればよい。このようにして、本システムはプロセッサの知識を持たない幅広いユーザに対する専用プロセッサ設計環境を与える。

本報告では、命令セットの最適設計とコード生成について基本的な枠組を述べると同時に、具対例を示す。

### 2. 命令セット設計

命令セット設計はプロセッサ設計において最重要項目であり、RISC、CISC 等様々な論争、評価 [2] がなされている。本システムではこれらの議論をふま次の設計指針を考える。命令セットは、最初、入力アルゴリズムを実行するのに必要最低限のものとし簡略化を図る (RISC)。その上でアルゴリズムのセマンティクスを解析し、有用で高頻度な高機能命令を取り入れる (CISC)。また、設計されたプロセッサに対する最適化コンパイラも自動合成することにより、ソフトウェアを含めた総合システムの実現をめざす。

### 3. 処理概要

本システムは、中間情報生成系、解析系、合成系、コード生成系の4つから構成される。(図1)

中間情報生成系は、入力アルゴリズムに対し字句・構文解析を行ない、種々の最適化を行なった後、中間情報 (3 番地文、フロー情報) を生成する。

解析系は、中間情報に対しライフタイム解析、アルゴリズム解析、RAM 解析、I/O ポート解析、頻度解析、命令の組合せ解析、データフロー解析等の解析を行う。前者4つは入力アルゴリズム実現に最低限どれだけの命令、ハードウェア要素が必要かの情報を調べ、後者3つは命令高機能化のための情報を調べる。

合成系は、解析系で得られた諸情報を基に命令セットを合成する。この際、ユーザは解析結果を基に命令セットを変更することができる。そして、合成された命令セットを基に機能ブロックを合成し、プロセッサのRT情報 (ハードウェア要素とその接続関係、状態遷移図、マイクロ命令のフォーマット) を出力する。

コード生成系は、合成系で得られた命令セットを基にコード生成部を合成し、マイクロプログラムを出力する。

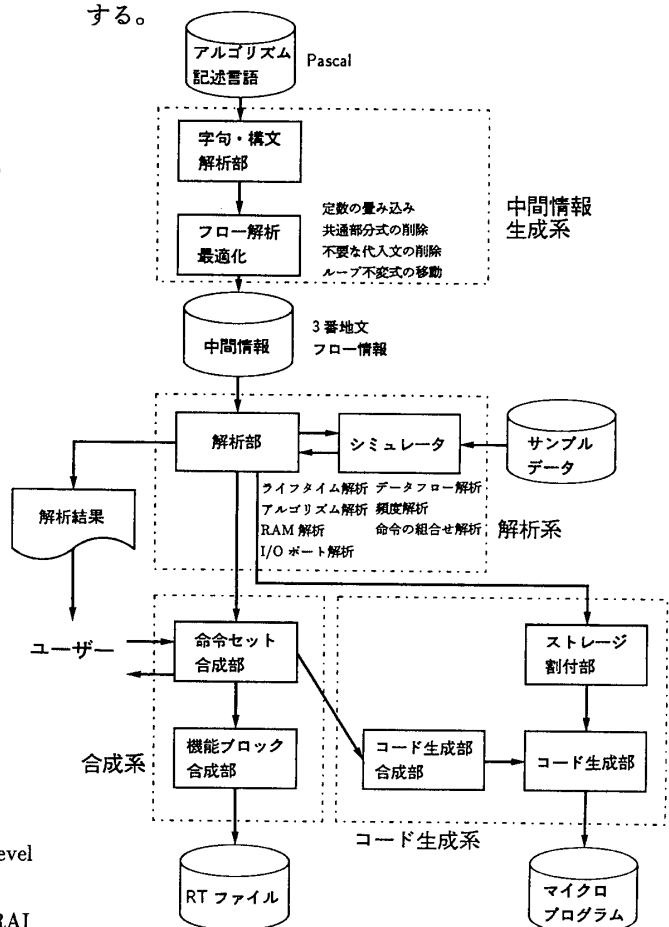


図1 処理フロー

Special Purpose Processor Design System Based on High Level Specification Description.

Takeshi IKENAGA, Toshiyuki TAKEZAWA, Katsuhiko SHIRAI  
Waseda University

### 4. 適用例

本システムを、代表的信号処理アルゴリズムである PARCOR 格子形フィルタに適用した例を示す。

仕様記述は、図2に示すようにほぼ Pascal の言語仕様に基づいて記述できる [3]。

これに対し命令セット合成を行なった結果を図3に示す。右下の OP1 は、頻度解析の結果、高頻度に出現する命令列を複合化したものである。この例では、信号処理特有の積差の命令列が高頻度で実行され、命令の組合せ解析によって抽出されたことを示している。

最後に決定された命令セットに対しコンパイラを合成し、マイクロプログラムを出力した結果を図4に示す。図4で、左列は中間情報である3番地文の列、中列、右列は生成したマイクロプログラムに対するそれぞれニーモニック、ビット列を示している。B0014 は13個の3番地文の列から構成されているが、複合命令 OP1 の導入によりマイクロ命令は11ステップとなっている。頻度解析により B0014 は1524回参照されているので、約3000ステップの短縮がはかられている。

### 5. むすび

高位の仕様記述に基づく専用プロセッサ設計支援システムについて報告した。本システムを用いて、ユーザはソフトウェアを記述するのと同じ感覚で専用プロ

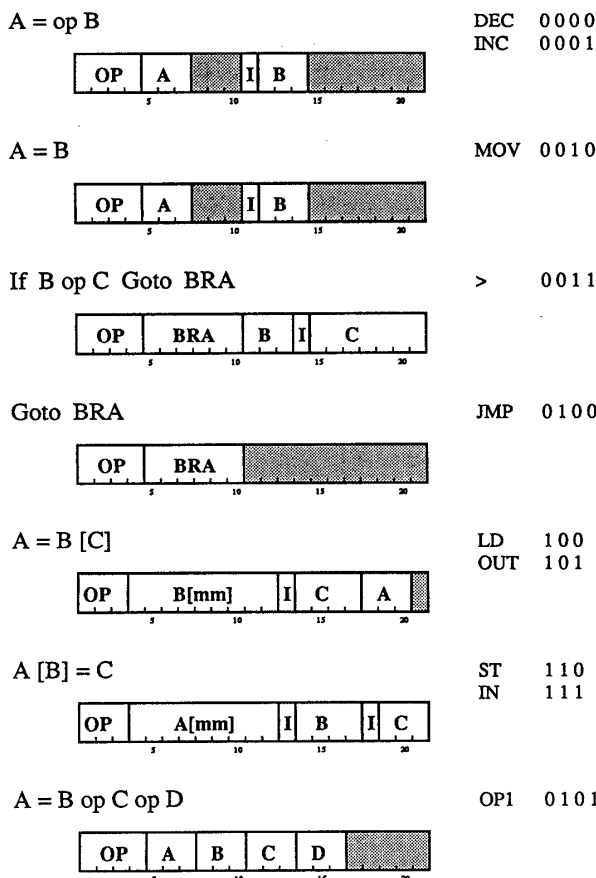


図3 命令セットフォーマット

セッサの設計が可能である。今後は、本システムで得られる RT 情報を既存の RT 言語とつなぐことにより LSI CAD 全体としてのシステムの実現と評価を試みたい。

### 参考文献

- [1] 竹沢、上野、白井：特定用途向け回路アーキテクチャ設計支援システム、信学論 D、Vol.J71-D、(1988-10)
- [2] 宇都宮公訓：命令セットアーキテクチャの評価、情報処理、Vol.29、No.12 (1988)
- [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)

```

program parcorfilter (s_in, rk_in, e_out);
const m1 = 13; n = 128;
input s_in : signed bit 12; rk_in : signed bit 8;
output e_out : signed bit 12;
var i, j : signed bit 16;
    ft, gto : array [m1] of integer;
    s, e : array [n] of signed bit 12;
    rk : array [m1] of signed bit 8;

begin
    : 入力部

    for i := 0 to m1-1 do
        begin gto[i] := 0; ft[i] := 0 end;
    e[0] := s[0];
    for j := 1 to n-1 do
        begin
            ft[0] := s[j]; gto[0] := s[j-1];
            for i := 1 to m1-1 do
                begin
                    ft[i] := ft[i-1] - rk[i-1] * gto[i-1];
                    gto[i] := gto[i-1] - rk[i-1] * ft[i-1];
                end;
            e[j] := ft[m1-1]
        end;

    : 出力部

end.

```

図2 仕様記述 (PARCOR フィルタ)

```

B0012 :
21 : T0006 := S (J)          : LD WR0 S WR5          : 100000011010001010000
22 : FT (0) := T0006        : ST FT 0 WR0           : 11000000000001000000010
23 : I := -1                : MOV WR6 1             : 00101100001001000000000
24 : T0007 := J - 1         : DEC WR0 WR5           : 00000000000010100000000
25 : T0008 := S (T0007)     : LD WR0 S WR0          : 10000001101000000000000
26 : GTO (0) := T0008       : ST GTO 0 WR0          : 11000000110110000000000

B0013 :
27 : IF I > 12 GOTO B0015   : > WR6 12 B0015        : 00111010011101000111000

B0014 :
28 : T0010 := I - 1         : DEC WR0 WR6           : 00000000000011000000000
29 : T0011 := FT (T0010)    : LD WR1 FT WR0         : 10000000000000000000010
30 : T0013 := RK (T0010)    : LD WR2 RK WR0         : 10001100110100000000100
31 : T0015 := GTO (T0010)   : LD WR3 GTO WR0        : 10000000110100000001100
32 : T0016 := T0013 * T0015 : OP1 WR1 WR1 WR2 WR3   : 01010010010100110000000
33 : T0017 := T0011 - T0016 :                          :
34 : FT (I) := T0017        : ST FT WR6 WR1         : 11000000000000011000001
35 : T0023 := FT (T0010)    : LD WR0 FT WR0         : 10000000000000000000000
36 : T0024 := T0013 * T0023 : OP1 WR0 WR3 WR2 WR0   : 01010000110100000000000
37 : T0025 := T0015 - T0024 :                          :
38 : GTO (I) := T0025       : ST GTO WR6 WR0        : 11000000110100110000000
39 : I := I + 1             : INC WR6 WR6           : 00011100000110000000000
40 : GOTO B0013             : JMP B0013             : 01000110110000000000000

B0015 :
41 : T0027 := FT (12)       : LD WR0 FT 12          : 10000000000001110000000
42 : E (J) := T0027         : ST E WR5 WR0          : 11001010011100101000000
43 : J := J + 1             : INC WR5 WR5           : 00011010000010100000000
44 : GOTO B0011             : JMP B0011             : 01000101000000000000000

```

図4 マイクロプログラム