脉动阵列的自动综合 —FP 方法

夏心杰　孙永强　胡振江

(上海交通大学计算机系， 200030)

AUTOMATIC SYNTHESIS OF SYSTOLIC ARRAYS—FP APPROACH

Xia Xinjie, Sun Yongqiang and Hu Zhenjiang

(*Department of Computer Science, Shanghai Jiao Tong University, 200030*)

Abstract　In this paper, an automatic synthesis system FP/B of systolic array is presented, in which FP algebra and rewriting system are essentially used. Some concurrent functional forms are proposed, the expansion solutions of a class of linear recursive equations can be expressed by them directly. Algorithms which can be transformed into them may have efficient and regular computing structures, so parallelism and pipelinability hidden in the original algorithms are well developed. Based on the FP/B algebra we've given the formal definition of systolic arrays and constructed a systolic rewriting system with properties of termitation and correctness. FP/B user programs can be automatically rewriten into equivalent optimized systolic expressions, which can be directly mapped into VLSI architectures according to the geometric meanings of FP/B functions and concurrent functional forms. Finally, a typical instance is given to show the synthesis process.

摘要　本文将 FP 代数，重写理论与脉动阵列 (Systolic Arrays) 的设计结合起来，研究了脉动阵列的形式化设计和自动综合的问题. 文章中提出的 FP/B 并发计算型，不但可表示某一类 FP/B 递归方程的展开式解，而且可以用来等价地对算法进行重新描述，从而开发了计算的并行性和流水线性，获得一个规整高效的计算结构. 文章形式地用 FP/B 定义了脉动式，并根据 FP/B 代数，建立了具有终止性和保持正确性的脉动阵列重写系统，它能将用户 FP/B 程序自动转换为等价的脉动式，再根据 FP/B 并发计算型及一些函数的几何语义可较为直接地获得一个脉动阵列的硬件描述. 文末给出一个例子加以说明.

1989 年 10 月 27 日收到， 1990 年 8 月 15 日定稿. 作者 **夏心杰**， 32 岁， 1989 年在上海交通大学获博士学位，现在加拿大工作，目前从事脉动阵列机程序语言方面的研究工作. **孙永强**， 教授，博士导师，主要从事计算理论、新型语言方面的研究工作. **胡振江**， 1990 年在上海交通大学获硕士学位，现为博士生，目前主要从事脉动阵列机函数式语言方面的研究工作.

§ 1. 引 言

VLSI 技术的发展使得单个芯片能实现很强的功能 [8,13]，但复杂度的提高使得设计的工作量陡然增加，设计的正确性难以保证. 因此， VLSI 的形式化设计理论和 VLSI 设计自动化技术已成为 VLSI 设计中非常活跃的研究领域.

H. T.Kung[5] 提出的脉动阵列 (Systolic Array) 是 VLSI 设计中很有希望的研究方向，它结构规整统一，时序控制简单且有很高的计算并行性. 近年来，人们已为众多含有大量运算的问题设计出了好的脉动算法 (Systolic Algorithm) 并制成了非常有效的芯片. 但是，从一般性的算法描述综合出对应的脉动算法的描述的系统化方法尚在研究之中 [3]. 例如，数学方法 [14]，差分方程的进程映射方法 [10,11]，数据流图变换方法 [7] 等. 但是上述方法中，综合过程的非形式化使得我们很难进行正确性验证，同时综合过程的非构造性和启发式的分析方法阻碍了综合过程自动化的实现. 在脉动阵列的综合过程中，形式化不仅体现在描述问题的形式化上，而且体现在处理问题的形式化上.

我们在 Backus[1]FP 的基础上，根据脉动阵列自动综合的需要进行了扩充，形成 FP/B. 我们的中心工作是研究如何从用户写的描写并行算法的 FP/B 程序，自动地综合出保持其语义的脉动阵列，从而获得一个高效的硬件实现. 我们提出了开发 FP/B 上并发计算型的思想. 对于一类递归方程描述的算法，我们不但要展开求解，而且要把它所含的并发计算型提取出来，并通过并发计算型间的等价变换获得更规整、更高效的并发计算结构. 我们给出了 FP/B 上脉动阵列的形式定义，建立了具有正确性和终止性的脉动阵列重写系统，使得设计的正确性融于设计过程中. 文末给出了脉动阵列自动综合的实现算法及典型的例子.

§ 2. 函数式语言 FP/B

为了适合脉动阵列自动综合的需要，我们所设计的语言应具有： 1. 较高的抽象层次，面向用户描述算法； 2. 确定的几何解释，便于自动映射到其硬件实现； 3. 良好的代数性质，便于进行重写和程序变换； 4. 显式或隐式表达并行性. 为此，我们进行如下扩充：

(1) 引进时间概念：我们在原数据对象集中增加流对象，流是一个包含时序概念的对象，记为 $< s_1, s_2, \cdots, s_n, \cdots >^s$，其中每个分量 s_i 具有相同类型. 在流上我们定义了若干基本函数，如延时函数 z : $z :< s_i, s_{i+1}, \cdots, s_n, \cdots >^s \stackrel{def}{\equiv} < s_{i-1}, s_i, \cdots, s_n, \cdots >^s$，其它如分组函数 !，扩展函数 $\wedge$，超时函数 z^{-1} 等. 对于 FP/B 上其它函数 f 有：

$$f :< s_1, s_2, \cdots, s_n, \cdots >^s \stackrel{def}{\equiv} < f : s_1, f : s_2, \cdots, f : s_n, \cdots >^s$$

(2) 定义了完善的用户定义设施： FP/B 允许用户自定义若干一阶或高阶函数，同时参数提升的引入方便了用户书写程序. 下面是用 FP/B 描述的一个排序算法，它将一个序列 $< x_1, x_2, \cdots, x_n >$ 进行升序排列：

$$CDEF \quad C \circ [x, y] = gt \circ [x, y] \rightarrow [y, x]; id \circ [x, y] \tag{2.1}$$

$$DEF \quad Insert \circ [x] = x$$

$$Insert \circ [x|[y|w]] = null \circ w \to C \circ [x,y];$$

$$[1 \circ C \circ [x,y] | Insert \circ [2 \circ C \circ [x,y]|w]] \tag{2.2}$$

$$MDEF \quad Sort \circ [x|w] = null \circ w \to [x]; Insert \circ [x|Sort \circ w] \tag{2.3}$$

$CDEF$ 用来定义单元操作，它表示该操作不能再分解，它的抽象程度决定了综合结果的抽象程度． $MDEF$ 定义这段程序的主函数．上述定义中，函数 C 将两个数进行比较，小的放在前面，函数 Insert 将一元素插入已排好序的序列内， Sort 是主函数．上述描述中； $[x|y] \overset{def}{\equiv} apndl \circ [x,y]$, $[u\|v] \overset{def}{\equiv} apndr \circ [u,v]$.

本节中未加说明的符号，请参阅文献 [1].

§ 3. 并发计算型的开发

3.1 并发计算型

对于 $[f_1, f_2, \cdots, f_n]$ 中每个 f_i 可以并行地执行，对于 $f_1 \circ f_2 \circ \cdots \circ f_n$ 中，每个 f_i 间能以流水方式高效地计算．因此， FP/B 表达式中蕴含着很大的计算并发性．

所谓并发计算型就是能够确定地描述某种并发计算方式的高阶函数，我们根据 [12] 作了某些扩充，下面介绍几种典型的并发计算型：

(1) @ $\quad @ f: < x_1, x_2, \cdots, x_n > = < f : x_1, f : x_2, \cdots, f : x_n >$

(2) $/_L$ $\quad /_L \ f :< x_1, \cdots, x_n > = \begin{cases} x_1 & n = 1 \\ f :< f :< \cdots < f :< x_1, x_2 >, \cdots >>, x_n > & n \geq 2 \end{cases}$

(3) $/_R$ $\quad /_R \ f :< x_1, \cdots, x_n > = \begin{cases} x_1 & n = 1 \\ f :< x_1, f :< \cdots, f :< x_{n-1}, x_n >, \cdots >> & n \geq 2 \end{cases}$

(4) Δ_L $\quad \Delta_L f :< x_1, \cdots, x_{n-1}, x_n > = < f^{n-1} : x_1, \cdots, f : x_{n-1}, x_n >$

(5) Δ_R $\quad \Delta_R f :< x_1, x_2, \cdots, x_n > = < x_1, f : x_2, \cdots, f^{n-1} : x_n >$

(6) π_L $\quad \pi_L f :< x, k > = < f^{k-1} : x, \cdots, f : x, x >$

(7) π_R $\quad \pi_R f :< k, x > = < x, f : x, \cdots, f^{k-1} : x >$

(8) $\%_L$ $\quad \%_L f :< x, y_1, \cdots, y_n > = < a_1, \cdots, a_n, b, > \quad n \geq 1$

其中： $a_i = 1 \circ f :< 2 \circ f :< \cdots 2 \circ f :< x, y_1 >, \cdots >, y_i > \quad 1 \leq i \leq n,$

$b = 2 \circ f :< 2 \circ f :< \cdots 2 \circ f :< x, y_1 >, \cdots >, y_n >$

(9) $\%_R$ $\quad \%_R f :< x_1, \cdots, x_n, y > = < a, b_1, \cdots, b_n > \quad n \geq 1$

其中： $a = 1 \circ f :< x_1, 1 \circ f :< \cdots, 1 \circ f :< x_n, y > \cdots >>$

$b_i = 2 \circ f :< x_i, 1 \circ f :< \cdots, 1 \circ f :< x_n, y >, \cdots >> \quad 1 \leq i \leq n$

上面的高阶函数都与一定的计算结构对应 (见图 3.1).

3.2 序列上递归方程的展开

由于序列具有构造上的递归特性，因此，在 FP/B 上使用递归方程表达算法十分自然．为了开发蕴含在递归方程中的计算并发性，就必须将递归方程展开，下面的一些定理说明上述研究的并发计算型正是某一类方程的解．

定义 3.1: 给定函数 f, 对于函数 g 和 h, 如果存在函数 U_{Rf}, 使得对任意 x 有: $f \circ [h \circ x, U_{Rf}] = g \circ x$, 则称 f 关于 g,h 有右拟单位 U_{Rf}. (类似可定义左拟单位 U_{Lf}).

定理 3.2: 设 F 定义为： $F \circ [x|y] = null \circ y \to g \circ x; f \circ [h \circ x, F \circ y]$ 且 f 关于 g,h 有右拟单位 U_{Rf}, 设 $f' \circ [u,v] = f \circ [h \circ u, v]$ 则 $F \circ [x|y] = /_R f' \circ [x|y\|U_{Rf}]$.

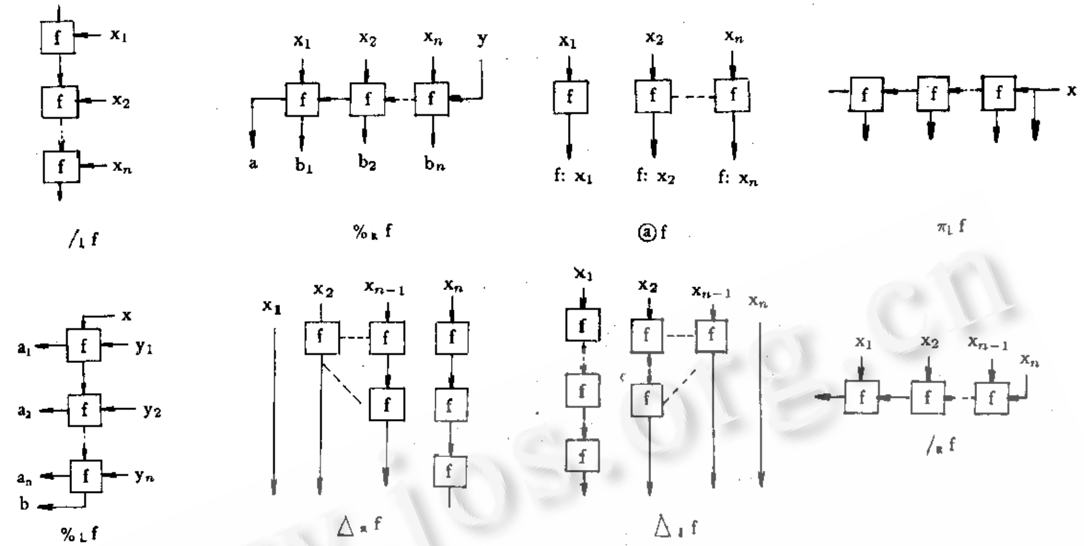


图 3.1 典型并发计算型的几何解释

定理 3.3: 设 F 定义为： $F\circ[x|[y|z]=null\circ z\rightarrow f\circ[x,y];[1\circ f\circ[x,y]|F\circ[2\circ f\circ[x,y]|z]]$, 则 F= $\%_L f$.

限于篇幅，其它定理略去.

§ 4. 脉动表达式的定义

设 $\varepsilon_{\mathrm{FP/B}}$ 表示 FP/B 上表达式的集合. 设 Exp $\in\varepsilon_{\mathrm{FP/B}}$, 定义 Prim (Exp) 表 Exp 中含有的初始函数 (不包括初始泛函) 的集合， Prim_C 表示 FP/B 中基本可运算函数集 (如 +, * 等，不包括走线函数如 trans, apndl 等及 z).

定义 4.1: 组线表达式集合 $L=\{l|(l\in\varepsilon_{\mathrm{FP/B}})\wedge(Prim(l)\cap(Prim\_C\cup\{z^{-1}\})=\phi)\}$. 任意 $l\in L$, 称 L 为组线表达式.

定义 4.2: c 称为细胞函数 $\Leftrightarrow c\in\varepsilon_{\mathrm{FP/B}}$ 而且 c 中不含并发计算型. 全部细胞函数的集合定义为 CELL.

定义 4.3: Sys 为脉动表达式集合定义为： Sys $\subset\varepsilon_{\mathrm{FP/B}}$ 且满足以下构造规则：

(1) 若 $f\in$ CELL, 则 $z\circ f\in$ Sys;

(2) 若 $f\in$ Sys, $l_1,l_2\in L$, 则 $f\circ l_1, l_2\circ f\in$ Sys;

(3) 若 $f,g\in$ Sys, 则 $f\circ g$, $[f,g]\in$ Sys.

定义 4.4: 若 $f\in$ Sys, 则@ f, $/_L f$, $/_R f$, $\Delta_L f$, $\Delta_R f$, $\pi_L f$, $\pi_R f$, $\%_L f$, $\%_R f\in$ Sys.

定义 4.5: $E\in\varepsilon_{\mathrm{FP/B}}$ 称为可脉动化 $\Leftrightarrow\exists l_1,l_2\in L$, $E^*\in$ Sys, 使得： $E^*=l_1\circ E\circ l_2$ 成立. E^* 称为 E 的一个脉动式.

定义 4.6: 设 $E_1,E_2\in\varepsilon_{\mathrm{FP/B}}$, $<E_1,E_2>$ 称为脉动等价对 $\Leftrightarrow$

(1) E_1 可脉动化 $\Leftrightarrow E_2$ 可脉动化；

(2) $\forall E^*\in$ Sys, E^* 是 E_1 的脉动式 $\Leftrightarrow E^*$ 是 E_2 的脉动式.

命题 4.7: 设 $E_1, E_2 \in \varepsilon_{\mathrm{FP/B}}$, 则 $< E_1 \circ E_2,\ E_1 \circ f \circ E_2 >$ 都是脉动等价对，其中 $f \in \{z^{-1}, \Delta_L z,\ \Delta_R z,\ \Delta_L z^{-1},\ \Delta_R z^{-1}\}$.

§ 5. 脉动阵列重写系统

脉动阵列重写系统 [2,4] 由以下几部分组成:

5.1 一般化简

一般化简是利用 FP/B 的代数性质进行静态运算，消除冗余运算. 例如:

[A1] $most \circ [x \| y] \to x$

[A2] $last \circ [x \| y] \to y$

[A3] $hd \circ apndl \to 1$

[A4] $tl \circ apndl \to 2$

[A5] $apndl \circ [1,2] \to apndl$

[A6] $most \circ \Delta_L z \to z \circ \Delta_L z \circ most$

5.2 计算格式的变换

为了开发 FP/B 上的并行计算，这就需要用并发计算型来重写原表达式，例如:

[C1] $! \circ [x, l] \to \pi_L z \circ [x, l]$

其中 $! : << x_1, x_2, \cdots >^s, k > \stackrel{def}{\equiv} << x_1, \cdots, x_k >, < x_2, \cdots, x_{k+1} >, \cdots, < x_i, \cdots, x_{i+k-1} >, \cdots >^s$

如果一个表达式含有几个并发计算型，为了得到更规整的结构，就需进行并发型间的重写，例如:

$[/_R - @]$ $\quad /_R f \circ [\ @\ g \circ x \| y] \to /_R (f \circ [g \circ hd, last]) \circ [x \| y]$

5.3 脉动式的变换

根据脉动式的定义，在脉动式中，每个细胞函数皆以 $z \circ f$ 的形式出现. 下述重写规则说明如何将外部的 z 移到细胞函数上，例如:

[D1] $f \notin Sys \Rightarrow z \circ /_R f \to /_R (z \circ f) \circ \Delta_R z^{-1} \circ [most \| z \circ last]$

[D2] $f \notin Sys \Rightarrow z \circ \%_L f \to [\Delta_L z \circ most \| last] \circ \%_L (z \circ f) \circ \Delta_L z^{-1} \circ [z \circ hd, tl]$

5.4 脉动式优化

脉动式优化的目的是提高其脉动时效，例如:

$[E_1]$ $z \circ z^{-1} \to id$

$[E_2]$ $\Delta_R z \circ \bar{C} \to \bar{C}$ （$\bar{C}$ 是常函数).

$[E_3]$ $\Delta_R z \circ \Delta_R z^{-1} \to id$

$[E_4]$ $f \circ h = id_\wedge f \in Sys \Rightarrow /_R (f \circ F \circ [1, g \circ 2]) \to f \circ /_R (F \circ [1, g \circ f \circ 2]) \circ [most \| h \circ last]$

除了上述四部分，还有等式定理 [1]. 可以证明，这个重写系统保持计算语义和终止性，限于篇幅，不再赘述.

§ 6. 脉动阵列重写系统的工作策略

重写系统的工作流程如下图所示:

用户 FP/B 程序 E.

↓

(展开求解，得非递归描述的表达式 E_1)

↓

子系统 I: 并发计算格式变换，得终止式 E_2

↓

确定细胞函数 $E_2(f_i)$, 令 $k=2$

↓

子系统 II: 将 $z \circ E_K$ 作脉动化变换,

得终止式 E_{k+1}, 令 $k=k+1$

↓

否

E_k 中的细胞函数 f_i 是否都以 $z \circ f_i$ 形式出现？

↓ 是

如果存在 E_{k+1}, 使 $< E_k, E_{k+1} >$ 是脉动等价对

(命题 4.7) 则令 $k=k+1$

↓

是

子系统 III: 优化 E_k, 得终止式 E_{k+1}, 令 $k=k+1$

↓

是否 $z^{-1} \in P(E_k)$?

↓ 否

E_k 为最终的脉动式

§ 7. 举例 — 排序算法的脉动阵列综合

用户的 FP/B 描述见 § 2.

7.1 消除递归，开发并发计算型

由定理 3.3: $Insert = \%_L C$ (4)

又因为 $Insert \circ apndl \circ [x, \bar{\phi}] = [x]$ 所以 $U_{R(Insert \circ apndl)} = \bar{\phi}$

而 $f' \circ [u, v] = Insert \circ apndl \circ [id \circ u, v] = Insert \circ apndl \circ [u, v]$

由定理 3.2: $Sort \circ [x|w] = /_R(Insert \circ apndl) \circ [[x|w]\|\bar{\phi}]$ (5)

(4) 代入 (5) 得: $Sort \circ [x|w] = /_R(\%_L C \circ apndl) \circ [[x|w]\|\bar{\phi}]$ (6)

7.2 计算格式变换

由于 (6) 式已是规整结构，不需此变换.

7.3 脉动式变换

$z \circ Sort \circ [x|w] = z \circ /_R(\%_L C \circ apndl) \circ [[x|w]\|\bar{\phi}]$

$\xrightarrow{[D_1]} /_R(z \circ \%_L C \circ apndl) \circ \Delta_R z^{-1} \circ [most\|z \circ last] \circ [[x|w]\|\bar{\phi}]$

$\xrightarrow{[A_1, A_2, E_2]} /_R(z \circ \%_L C \circ apndl) \circ \Delta_R z^{-1} \circ [[x|w]\|\bar{\phi}]$

$\xrightarrow{[D2]} /_R(\Delta_R z^{-1} \circ [most\|z \circ last] \circ \%_L(z \circ C) \circ [hd|\Delta_R z \circ tl] \circ apndl) \circ \Delta_R z^{-1} \circ [[x|w]\|\bar{\phi}]$

$$\xrightarrow[{[f\circ h,g\circ h]}]{[f,g]\circ h} /_R(\Delta_R z^{-1}\circ[most\|z\circ last]\circ\%_L(z\circ C)\circ[hd\circ apndl|\Delta_R z\circ tl\circ apndl])\circ\Delta_R z^{-1}\circ[[x|w]\|\bar{\phi}]$$

$$\xrightarrow{[A_3,A_4]}\Delta_R z^{-1}\circ/_R([most\|z\circ last]\circ\%_L(z\circ C)\circ apndl\circ[1,\Delta_R z\circ 2])\circ\Delta_R z^{-1}\circ[[x|w]\|\bar{\phi}]$$

$$\xrightarrow{[E_4]}\Delta_R z^{-1}\circ/_R([most\|z\circ last]\circ\%_L(z\circ C)\circ apndl\circ[1,\Delta_R z\circ\Delta_R z^{-1}\circ 2])$$
$$\circ[most\|\Delta_R z\circ last]\circ\Delta_R z^{-1}\circ[[x|w]\|\bar{\phi}]$$

$$\xrightarrow{[E_3,E_5]}\Delta_R z^{-1}\circ/_R([most\|z\circ last]\circ\%_L(z\circ C)\circ apndl)\circ[most\|\Delta_R z\circ last]\circ\Delta_R z^{-1}\circ[[x|w]\|\bar{\phi}]$$

7.4 消去 z^{-1}

上式与下式形成脉动等价对 (命题 4.7).

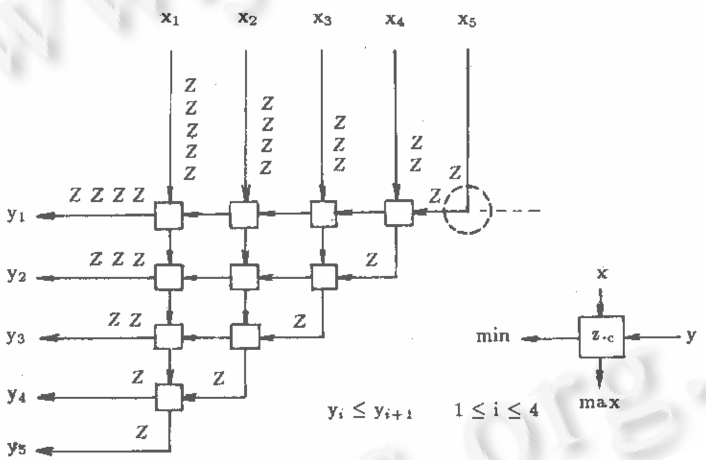
$$\Delta_L z\circ/_R([most\|z\circ last]\circ\%_L(z\circ C)\circ apndl)\circ[most\|\Delta_R z\circ last]\circ\Delta_L z\circ[[x|w]\|\bar{\phi}]$$

$$\xrightarrow{[A_1,A_2,A_6]}\Delta_L z\circ/_R([most\|z\circ last]\circ\%_L(z\circ C)\circ apndl)\circ[z\circ\Delta_L z\circ[x|w]\|\Delta_R z\circ\bar{\phi}]$$

$$\xrightarrow{[E_2]}\Delta_L z\circ/_R([most\|z\circ last]\circ\%_L(z\circ C)\circ apndl)\circ[z\circ\Delta_L z\circ[x|w]\|\bar{\phi}] \tag{7}$$

7.5 硬件结构映射

最终得到的 (7) 式，根据 $/_R$, $\%_L$ 的几何解释，可方便地导出下述硬件结构.



§ 8. 结论和讨论

这项研究表明， FP 不但适用于硬件描述和 VLSI 设计，而且描述简洁方便，实现直接容易. 本文将计算机科学中的一些理论引进到脉动阵列的设计上，使得常见的算法都能用 FP/B 递归描述，经展开和重写，有效地求出其脉动式，再根据并发计算型的几何解释，自动设计出硬件结构. 这种形式化的方法，保证了设计的正确性.

研究同时表明， FP/B 单向数据作用的特点，使得它不能很好地描述某些非含数据环路不可的算法. 对于网络中长期驻留的数据单元也不易描述. 另外 FP/B 的递归方程在我们这个系统中可能无解，应进一步扩大可解的递归方程类. 我们正着手解决上述问题，对于含数据环路算法的描述等问题的解决方法请参阅 [15].

参考文献

[1] J. Backus, "Can Programming be Liberated from the Von Neumann Style? A Functional Style and Its Algebra of Programs", CACM, No. 3, 1978.

[2] F. Bellegarde, "Rewriting System on FP Expression to Reduce the Number of Sequence Yielded", Science of Computer Programming 6 (1986), North–Holland.

[3] J.A.B. Fortes, K.S. Fu, "Systematic Approaches to the Design of Algorithmically Specified Systolic Arrays", Purdue Univ. TR–EE84–39.

[4] G. Huet, "Equations and Rewrite Rules, a Survey", Tech. Report, Standford Univ. STAN–CS–80–785.

[5] H.T. Kung, "Let's Design Algorithms for VLSI Systems", Tech. Report, Carnegie-mellon Univ. TR–CMU–CS–79–151.

[6] S.Y. Kung, K.S. Arum, "Wavefront Array Processor: Language, Architecture, and Applications", IEEE trans. on computer Vol. 3–31, No. 11, 1982.

[7] C.E. Leiserson, J.B. Saxe, "Optimizing Synchronous Systems", Tech. Report. Carnegie–Mellon Univ. TR–CMU–CS–82–101.

[8] C. Mead, L. Conway, "Introduction to VLSI Systems" Addison-wesley, 1, 1980.

[9] G. Milne, "CIRCAL: A Calculas for Circuit Description", Univ. of Edinburge, Report CSR–122–82, 1982.

[10] D.I. Moldovan, J.A.B. Fortes, "Partitioning and Mapping Algorithms into Fixed Size Systolic Arrays", IEEE trans. on computer Vol. c–35, No. 1, 1986.

[11] P. Quinton, "Automatic Synthesis of Systolic Arrays from Uniform Recurrent Equations", Proc. IEEE, 1984.

[12] M. Sheeran, "uFP–An Algebraic VLSI Design Language" Oxford Univ. Ph. D thesis, TM–PRG–39, Nov. 1983.

[13] J.D. Ullman, "Computational Aspects of VLSI", Computer Science Press, 1983.

[14] U. Weiser, A.L. Davis, ""Mathematical Representation for VLSI Arrays", Tech. Report, Utah Univ. TR–UU–CS–80–111.

[15] 胡振江，孙永强， "并行算法的 FP 描述及其脉动化的判定"，软件学报， Vol. 3, No.3, 1992.