第16卷第6期　　半 导 体 学 报　　Vol. 16, No. 6
1995年6月　　CHINESE JOURNAL OF SEMICONDUCTORS　　June, 1995

数据流图的优化：时序重构和结合关系重构

王以峰　洪先龙　徐葭生

（清华大学　北京　100084）

摘要　数据流图(DFG)是数字系统行为描述的一种图形表示方式. 实验证明，DFG 内部结构对硬件结构实现有很大影响. 本文介绍的时序重构和结合关系重构被用来在不改变 DFG 外部功能的前提下对 DFG 的内部结构进行改造，使之更有可能在结构自动综合中构造出性能最优、造价最小的硬件结构. 文中首先定义两种重构技术的意义，讨论它们在 DFG 优化中的应用，然后提出一个新颖的优化算法.

CCACC：7410D，5210B，5260

1　引言

数字系统硬件结构自动综合技术越来越引起人们的重视. 结构自动综合(Architectural Synthesis)旨在实现数字系统的行为描述到实现其功能的寄存器传输级(Register Transfer Level)硬件结构的自动转换. 一般过程是：首先将用户提供的关于所设计数字系统行为的语言(如 VHDL)描述经编译转换成一种从结构设计角度分析更加简单明了的形式——数据流图(DFG)，在此基础上进行诸如操作调度、硬件资源分配、控制码提取等一系列过程，最终构造出用以实现数字系统功能的硬件结构.

将数字系统行为的语言描述翻译成 DFG 的过程通常是一个“直译”的过程，也就是说得到的 DFG 在功能和结构上完全对应于原来的语言描述. 语言描述中的变动，甚至于两个可调换语句位置的改变也会直接反映到 DFG 的结构上. 图 1 为两个语言描述(片断)及对应的 DFG. 两个描述片断的整体功能完全相同，但 DFG 结构上的相异却十分明显. 实践表明，最终的便件结构实现与 DFG 内部结构有着十分密切的关系. 代表同样功能而内部结构不同的两个 DFG 导致的最终硬件实现在性能、造价及其它方面可能会有相当大的差别. 以图 1 为例. (a)所示的 DFG 限定了三个加法操作必须串行进行，从而运行时这个局部至少要占三个操作周期的时间长度(假设每个加法操作占一个操作周期)；而(b)所示的 DFG 提供了操作

王以锋　男，1965年生，博士，主要从事高层次逻辑综合方面的研究工作
洪先龙　男，1940年生，教授，主要从事 IC CAD 方面的科研、教学工作
1993年10月27日收到本文

并行的可能性，使得行运时只占两个操作周期. 当然，后者需要两个加法器作为支持.

为了最终能获得最优的硬件结构设计，无非两种途径：一是要求用户书写的行为语言描述可保证产生最佳的 DFG 结构；另一个途径是硬件结构自动综合系统能自动实现 DFG 结构的优化. 显然真正可取的是后一途径，因为前者给用户利用结构自动综合技术增加了相当大的负担. 本文将就 DFG 结构自动优化问题展开一系列讨论，包括 DFG 结构的评判原则，进行 DFG 结构改造的两大重构技术：时序重构(Retiming)和结合关系重构(Association)以及如何有效引导 DFG 的结构改造朝优化结构的方向进行等问题.

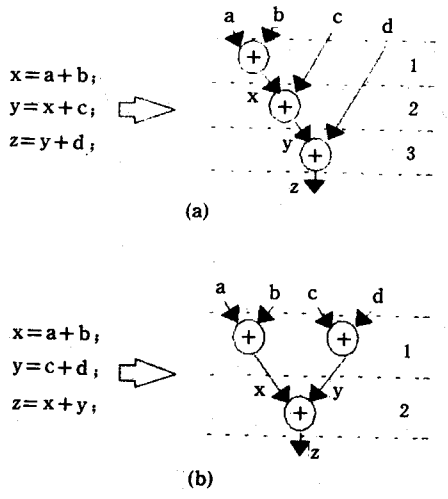

图 1　语言描述与对应的 DFG

2　DFG 及其结构评判原则

数据流图(DFG)是数字系统行为功能的一种图形表示方式，由结点，有向边和 I/O 端口组成.

$$DFG=(V,E,I/O)$$

DFG 中，结点(V)对应于数字系统的行为操作(如＋，＊等)，有向边(E)对应于操作间数据的传输，或称数据依赖关系. 数字系统的输入/出映射到 DFG 上为 DFG 的 I/O 端口. 在用 DFG 描述数字信号处理(DSP)算法时，我们为每个有向边 e 定义了一个延迟参量，记为 W(e)，其物理意义是描述 DSP 算法中常见的周期性延迟因子(Z^{-1}). W(e)为 e 代表的数据传输路径上 Z^{-1}因子的多少，显然 W(e)≥0. 图 2 为一个二阶滤波器的信号流图(a)和相应的数据流图(b).

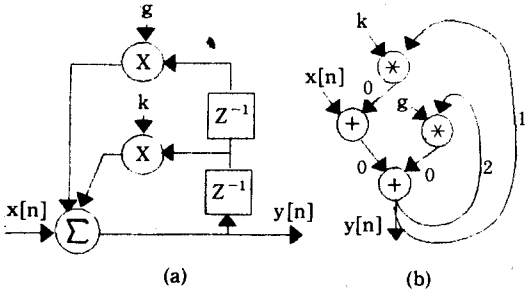

图 2　DSP 算法的信号流图(a)和 DFG(b)

DFG 与描述 DSP 算法的信号流图十分接近，只是将 Z^{-1}因子作为数据传输边上的参量以便于处理. 为清楚起见，本文引用的 DSP 算法的例子有时用信号流图的形式来标示结构上的改变，但须记住真正的处理是针对 DFG 进行的.

一般说来，一个 DFG 所代表的数据处理功能体现在输入-输出关系上，而其行为特征则反映在 DFG 之间并不存在一一对应关系，同一个数据处理功能可以由多个具有不同结

构的 DFG 来描述. 我们把代表的功能相同而内部结构不同的 DFG 称为"同形异构"DFG. 这里的"形"特指在 DFG I/O 端口上呈现的数据处理功能.

不同结构的 DFG 将导致不同的硬件实现. 我们试图用 DFG 最终造就的硬件结构的性能(速度)和造价(资源用量)来评判 DFG 内部结构的优劣. 显然,最终硬件所能达到的数据处理速度(非流水结构下完整处理一个(组)输入数据所用时间 EL 或流水结构下输入数据间隔时间 IL)最快,所用硬件资源(功能单元、寄存器、多路器、连线等)最少,则称 DFG 最佳.

必须注意,我们所进行的 DFG 结构优化过程发生在实际进行硬件结构综合之前(否则也就失去了 DFG 结构优化的意义). 换言之,此时我们还无法得知最终硬件实现的速度、造价等信息. 幸运的是,DFG 与最终硬件实现之间有一定的对应性,通过仔细分析 DFG 的结构可对未来硬件实现的一些指标作出估计. [1,2]中有通过分析 DFG,对硬件资源需求量进行较精确估计的详细讨论,这里不再赘述,只简单分析一下 DFG 结构与最终硬件实现的速度之间的关系.

[定义 1]DFG=(V,E,I/O),设存在一组结点 $v_1,v_2\cdots v_n\in V$,满足 $e_{i,i+1}=<v_i,v_{i+1}>\in E(i=1\cdot\cdot n-1)$,则称该组结点构成了 DFG 中一条数据传输路径,记为 $p=[v_1,v_2\cdots v_n]$. 如 $<v_n,v_1>\in E$ 则称构成了环路. 路径 P 的长度为 p 中结点代表的操作的运行时间总和,即 $|p|=\sum_{i=1}^{n}\mathrm{Delay}(v_i)$;路径 p 的延迟参量 W(e)定义为 p 中有向边的延迟参量之和,即 $W(p)=\sum_{i=1}^{n}W(e_{i,i+1})$,对于环路,$W(p)=\sum_{i=1}^{n-1}W(e_{i,i+1})+W(e_{n,1})$.

由定义 1,我们可以写出求 DFG 最终硬件实现的 EL 和 IL 的方程如下:

$$EL=\underset{p}{MAX}\{|p|W(p)=o\} \tag{1}$$

$$IL=\underset{p}{MAX}\{[|p|/W(p)]|p\text{ 为环路}\} \tag{2}$$

容易理解,IL 只受限于环路而非最长路径.

DFG 结构的评估函数(sef)构造如下:

$$sef(DFG)=\alpha_1\times\delta(IL-\widetilde{IL})+\alpha_1\times\delta(EL-\widetilde{EL})+\sum_r\beta r\delta(Mr-\widetilde{Mr})+\gamma\sum_{op\in V}1/TF(op) \tag{3}$$

说明如下:

①$\delta(x)=\begin{cases}x & x>o\\ o & x\leqslant o\end{cases}$

②$\widetilde{EL},\widetilde{IL},\widetilde{Mr}$为用户的规定值(未规定视为 0). Mr 为资源 r 的需求量估计值.

③α、β、γ 为系数. 为保证用户提出的要求优先得到满足,相应项的系数应足够大,使得用户要求未得到满足时 sef 中其它项几乎不起作用.

④TF(op)为结点操作 op 的调度自由域长度[1],TF(op)=ALAP(op)−ASAP(op)+1.

⑤sef 函数值越小,DFG 结构越优.

式(3)是一个通式，在具体情形下可以简化. 如在非流水结构模式，用户要求了速度的情形下，sef 可简化成：

$$sef(DFG)=\alpha\times\delta(EL-\widetilde{EL})+\sum_{r}\beta_r\times Mr+\gamma\sum_{op}1/TF(op)$$

我们取 $\alpha=1000$，$\beta_r=\cos T(r)$(资源价格)，$\gamma=0.8$

3　DFG 结构转换

DFG 结构优化的任务是从“同形异构”的 DFG 中择出 sef 最小的一个用于后序的硬件实现. 为实现这个目标，首先应在不改变 DFG 代表的功能的前提下实现 DFG 结构的转换，即由最初的 DFG(由用户的语言描述直接翻译而来)推出其它“同形异构”DFG. 这一工作是由时序重构和结合关系重构完成的.

3.1　时序重构

在 DSP 应用中，周期性延迟因子十分常见(如滤波器)，并对 DSP 算法的结构影响很大. 时序重构是通过合法变更周期性延迟因子的位置来获得“同形异构”的 DFG. 利用时序重构可以导致 DFG 结构发生巨大变化，但时序重构的最基本操作则是沿某一方向移动一个延迟因子. 必须保证移动是合法的，即不会改变 DFG 外部呈现的功能，原则如公式(4)

$$D(a\bigstar b)\rightleftharpoons D(a)\bigstar D(b)\qquad(4)$$

式中“★”为广义操作符，可以为＋、＊等真实操作，也可为数据传输中的一个分叉点. 图 3 为常见的合法移动的例子.

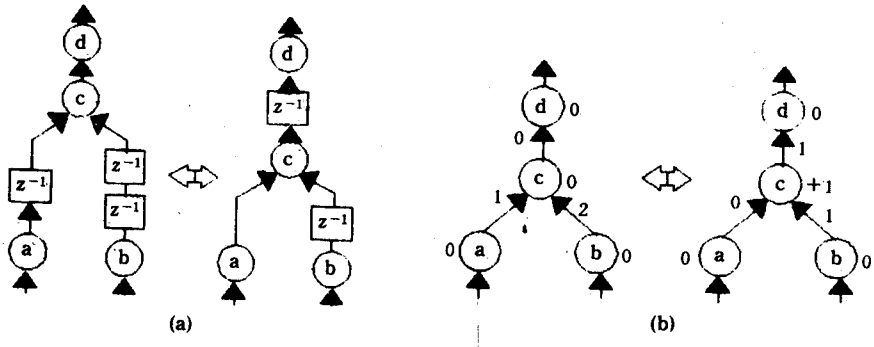

图 3　时序重构操作示意图

(a)信号流图形式　(b)数据流图形式

为了在重构过程中能够尽量少的信息有效记录时序重构的轨迹，以便快捷地得到最佳结构的 DFG，我们为 DFG 中每一个结点(包括分叉点)定义了一个时序重构参量，记为 r(v). 当有 Z^{-1}因子跨越 v 移动时，r(v)将作相应改变.

[定义 2]　DFG 结点的 r(v)初值为 0.

[定义 3]　当 Z^{-1}因子跨越结点 v 从其输入移向输出端，称为正向移动，r(v)＝r(v)＋1；反之称为逆向移动，r(v)＝r(v)－1.

我们用$\overline{W(e)}$标示 DFG 有向边 e 延迟参量的初值(对应原始语言描述),那么可以很容易推出 W(e)与 r(v)的关系方程:

设 $u \xrightarrow{e} v$

$$W(e)=\overline{W(e)}+r(u)-r(v) \tag{5}$$

可以证明,只要满足

$$\forall e\in E, W(e)\geq 0$$

DFG 所代表的功能将不会改变.

在时序重构过程中,只有 r(v)发生变化,W(e)的变化则蕴含在 r(v)变化之中.容易理解 r(v)比 W(e)更能有效地标示时序重构的轨迹.

既然时序重构的操作对象是周期延迟因子,显然,时序重构只对表示 DSP 算法的 DFG 结构改造有效.

3.2 结合关系重构

图 1 中两个 DFG(局部)结构差异颇大,但实现的功能却是完全一样的.稍加分析即可行出结论:这是算术运算结合律作用的结果.

$$(((a+b)+c)+d)\equiv((a+b)+(c+d))$$

利用运算结合律对某些操作的操作数作些调换,产生"同形异构"DFG 的过程称之为结合关系重构.表 1 中列出了常用的结合律变换规则.

表 1 结合律规则

	表达式 1	表达式 2	表达式 3
1	a+(b+c)	(a+c)+b	(a+b)+c
2	a*(b*c)	(a*c)*b	(a*b)*c
3	a+(b−c)	(a−c)+b	(a+b)−c
4	a*(b/c)	(a/c)*b	(a*b)/c
5	a−(b−c)	(a+c)−b	(a−b)+c
6	a/(b/c)	(a*c)/b	(a/b)*c
7	a−(b+c)	(a−c)−b	(a−b)−c
8	a/(b*c)	(a/c)/b	(a/b)/c

表 1 中同一行的三个表达式恒等.从结构角度看,三个表达式中各有一个操作数只经过一级运算(如表达式 1 的'a'),另外两个需经过两级运算.有趣的是,结合律有时会使被作用的操作类型发生改变(表 1 中的 5、6、7、8 规则).

结合律作用的最基本形式中包含两个操作和三个操作数.在利用结合律对 DFG 进行重构时,每次选取两个结点,判断重构可能性,按结合律相关规则进行结构改造.重构可能性由以下两点来判定:(设两结点分别为 u 和 v).

①u 和 v 代表的操作的操作类型符合结合律规则.典型的操作类型是+,−和*,/.

②$u \xrightarrow{e} v$(或 $v \xrightarrow{e} u$),$e\in E$ 且 $W(e)=0$.

利用结合律,一种形式的表达式可转换其它两种形式之一.为了便于记录变换轨迹,我们将图 4 中沿实箭头方向的变换称为正向变换,反之称为逆向变换,分别以+1,−1 标示之.

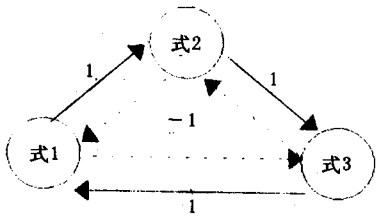

图 4　正逆变换定义

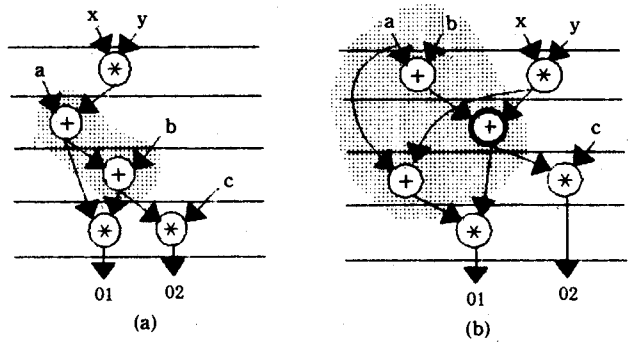

图 5　需增加新结点的结合关系重构

实际进行 DFG 结合关系重构时，常常会遇到这样的情况：u，v 符合重构条件，但存在 u $\xrightarrow{e'}$ v′ $e' \in E$ 且 $W(e')=0$. 这时重构结果通常要增加新的操作结点，新增加的结点的操作类型由结合律规则决定. 如图 5(a)中由两个加法操作构成的 DFG 局部(阴影部分)，经结合关系重构变换成(b)中阴影下的结构，增加了一个加法操作结点(粗线框所示).

4　DFG 结构的优化

时序重构和结合关系重构只是提供了构造“同形异构”DFG 的手段，真正实现 DFG 结构的优化还需要对结构重构过程进行有效引导，使其朝 DFG 结构优化方向进行. 试图构造出全部“同形异构”DFG，比较择出最佳 DFG 的方法一般是不可取的，因为“同形异构”DFG 的数量随 DFG 规模的增大而迅速增加. 这里我们提出一个目标引导下的迭代优化算法，有效引导和控制 DFG 结构和重构过程，使其迅速接近最佳的 DFG 结构. 所谓目标即 DFG 的评估函数值变小.

从前一节我们得知，无论时序重构还是结合关系重构，其基本变换都可以通过两个要素表示出来：对象和方向. 对象是指重构赖以实施的结点(时序重构的对象是一个结点. 结合关系重构为两个结点)；方向指的是重构为正或逆变换. 我们可用一个统一的形式表示这两种不同的重构变换：

重构类型	重构对象	重构方向

我们定义时序重构的类型为 1，结合关系重构为 2. 这样一个统一形式的表示称为重构变换记录. DFG 结构优化过程中某一时刻的 DFG 的结构支持的全部重构变换记录构成该时刻 DFG 的候选重构变换集合，记为 $CTS=\{ct_1, ct_2 \cdots ct_n\}$.

DFG 结构优化算法描述如下：

INPUT：原始数据流图 DFG_0；

OUTPUT：最佳结构的数据流图 DFG−opt；

begin：

/ * 初始化 * /

DFG−opt=DFG−tmp=DFG_0；

```
sef—val1=sef(DFG0);
invalid—count=0;
While(invalid—count<STOP—CRITEKION) do
begin
构造针对 DFG—tmp 的候选重构变换集 CTS;
sef—val2=∞;
for all ct ∈ CTS do
  begin
  New—DFG=Reconstruct(DFG—tmp,ct);
  if (sef(New—DFG)<sef—val1) then
    begin
    DFG—opt=DFG—tmp=New—DFG;
    sef—val1=sef(New—DFG);
    invalid—count=0;
    goto NEXT—ITERATION;
    end
  else if (sef(New—DFG)<sef—val1) then
    begin
    sef—val2=sef(New—DFG);
    cand—ct=ct;
    end;
  end for;
  DFG—tmp=Reconstruct(DFG—Tmp,cand—ct);
  invalid—count++;
NEXT—ITERATION:
  Continue;
end while;
end;
```

算法描述的是一个迭代优化过程. 迭代结束条件是连续 STOP—CRITERION 次迭代未能使数据流图的评估函数值进一步减小. 可以看出,迭代优化过程中优先接受 sef( )小的 DFG,但有时也接受 sef( )稍大的 DFG,以避免优化过程陷入局部最优解的低谷不能自拔. 这是模拟退火的特点,不过我们的算法目标性很强,使得解的收敛速度较模拟退火快得多.

5 DFG 结构优化的功效、实验与结论

DFG 结构优化已作为一部分集成在我们开发的 DSP 自动综合系统 DSPsyn 中. 实验表明,DFG 结构优化对降低最终硬件实现的费用和提高硬件实现的性能效果十分明显,因此是数字系统结构自动综合技术一个十分重要的组成部分. 下面用几个实例来说明 DFG 结构

优化所发挥的效用.

1 降低造价

一个数字系统最终硬件实现的造价来自所用资源(功能单元、寄存器单元、连线单元等)的费用. 通过 DFG 结构优化，可以增大每个操作的自由调度域(TF(op))，使得在后序调度过程中更有潜力均匀调度各类操作以减小功能单元的需求量.

图 5 为一由加、乘法操作结点组成的简单的 DFG. 假设加法和乘法操作各占一个操作周期，而限定 EL 为 4 个周期. 实现(a)所示的 DFG 至少需 2 个乘法器和 1 个加法器. 将(a)的 DFG 结构进行优化(阴影下的两个加法构成的局部进行结合关系重构)，虽然重构后 DFG 中增加了一个操作结点，但硬件实现时却节省了一个乘法器.

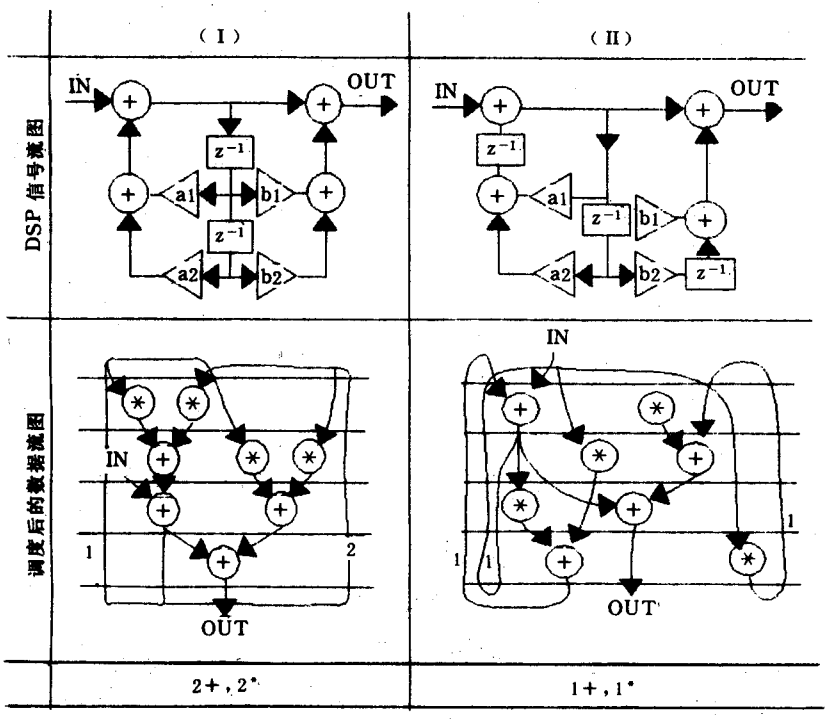

图 6　利用时序重构进行优化

图 6 为一 DSP 算法中常见的四阶 IIR 滤波器. 含 4 个乘法和 4 个加法. 仍假设 Delay(t)＝1^c，Delay(＊)＝1^c，EL＝4^c. 对于原始结构的 IIR(I)，由于结构的约束，4 个乘法只能安排在前两个操作周期执行，而 4 个加法操作也不可能安排在第一个周期，因而实现该 IIR 至少需 2 个乘法器和两个加法器. 利用时序重构对原始 IIR 的结构进行优化得(Ⅱ)所示结构，不难证明，(Ⅰ)、(Ⅱ)的功能完全一致，但实现(Ⅱ)的 IIR 只需 1 个加法器和一个乘法器. 单从功能单元角度衡量，费用节省达 50%.

我们用时序重构和结合关系重构联合对一 5 阶波动滤波器[3]进行 DFG 结构优化，表 2 列出了不同 EL 约束下优化前后资源用量的比较. 表 2 中同时列出了实现相应优化所用的 CPU 时间.

优化前的 DFG 根本无法实现 EL＜17^c.

表 2 （假设 Delay(+)=1^c,Delay(*)=2^c）

EL	优化前	优化后	CPU 时间(秒)
15	NA	3+,3*	1.02
16	NA	3+,2*	0.95
17	3+,3*	2+,2*	0.83
18	3+,3*	2+,2*	0.72
19	2+,2*	2+,2*	0.73

2 减小 EL 或 IL,提高硬件的速度

在非流水结构硬件实现方式下,硬件的速度取决于 DFG 中存在的延迟参量 W(p)为 0 的最长数据传输路径(称关键路径).通过优化 DFG 结构,或者将一些操作结点从关键路径移开(结合关系重构)或将 Z^{-1}因子移入关键路径实现现其分割,都可以缩短关键路径长度,提高硬件的速度.仍以 5 阶波动滤波器为例,(3+,3*)的资源用量在优化前只能达到 EL=17^c 的速度,优化后提高到 EL=15^c.

利用流水结构挖掘数字系统数据处理速度的潜力是一个十分有效且常用的手段.但流水线的 IL(代表数字系统数据处理速度)仍受 DFG 结构的约束.由公式 2.2,我们得知,利用 DFG 结构优化,要么缩短数据传输环路和长度,要么增加环路上 Z^{-1}因子个数以增大 W(p),,均可减小 IL.图 7 为一 Volterra 滤波器.我们在图中用阴影对限制 IL 的关键环中进

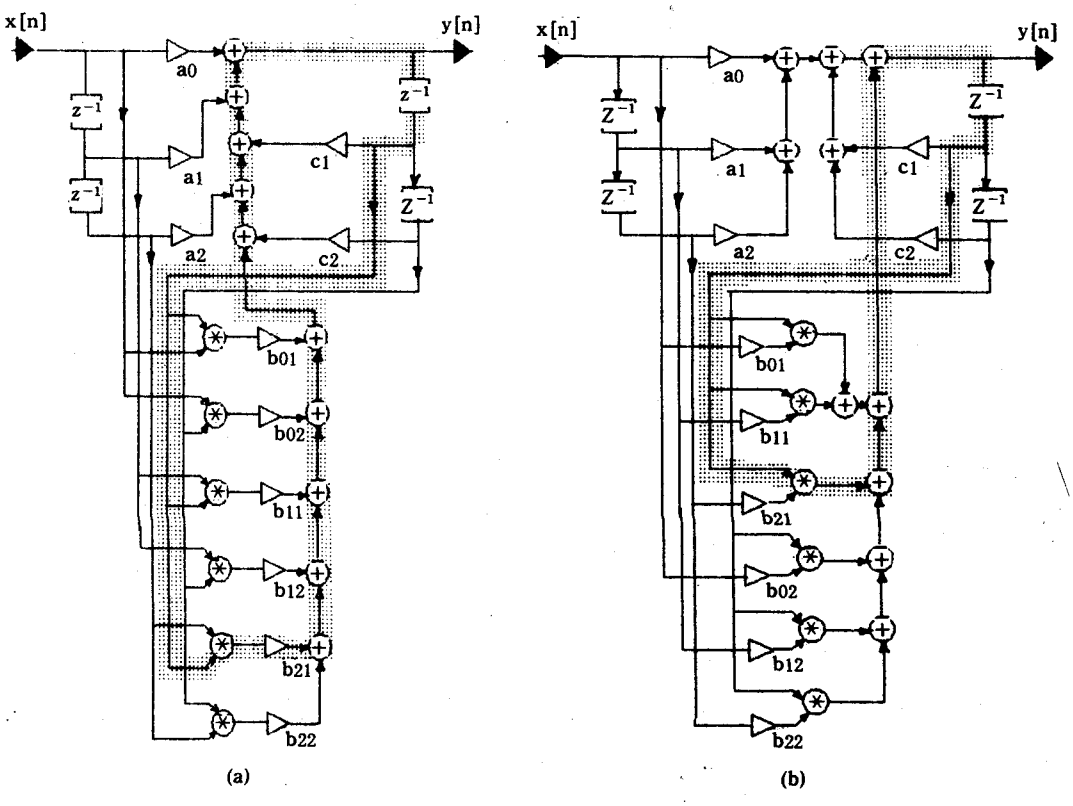

图 7

行了标示. 优化前无论多么充足的资源供给，IL 都不可能小于 12^c（设 Delay(+)=Delay(*)=1^c）；优化后 IL 降到 4^c，速度提高了两倍.

DSP 应用对性能要求较高，因而 DFG 优化对 DSP 结构自动综合而言更为必要.

参　考　文　献

[1] Yifeng Wang. Jiasheng Xu and Xianlong Hong, 3rd Int. Conf. on CAD & Graphics, 1993, pp. 430－437.

[2] Yifeng Wang, Xianlong Hong and Jiasheng Xu, Int. Conf. on CAPE, 1993, pp. 456－461.

[3] S. Y. Kung and H. J. Whitehouse, VLSI and Modern Signal Processing, Prentice Hall, 1985.

[4] V. J. Mathewa, IEEE Signal Processing Magzine, July, 1991, pp. 10－26.

[5] S. Malik, Trans. on CAD, 1991, **10**(1): 74－84.

DFG Optimization: Retiming and Association

Wang Yifeng, Hong Xianlong and Xu Jiasheng

(*Tsinghua University, Beijing* 100084)

Reveived 27 October 1993

Abstract Data flow graph (DFG) is a graphic representation of digital system behavior. Experiments have shown that the final hardware implementation of a digital system is on close terms with the given DFG structure. Retiming and association introduced in this paper, can be used to change DFG structure without altering its input-output relationships. In this paper, we first define the two transformation techniques and then discuss their application in DFG optimization. An effective algorithm is put forward to combine the two transformations and to guide the DFG optimization process to achieve the optimal DFG structure.

CCACC: 7410D, 5210B, 5260