

第十章、流水线结构的并行自适应递归滤波器

学过数字信号处理，就不可能不知道滤波器。经典的滤波器分两类：FIR 和 IIR。FIR 是有限长冲击响应滤波器，硬件电路是非递归的；而 IIR 是无限长冲击响应滤波器，硬件电路存在递归环路，值得注意的是，IIR 可以看成是一个 FIR 和递归环路的级联。

在第九章已讨论高速并行的 FIR 设计方法，而这一章的对象将是 IIR。IIR 不同于 FIR 之处在于其递归的环路，怎么在递归环路中进行流水化设计或者并行设计将是我们要解决的棘手问题，，注意，在《第二章、迭代边界》中，我们曾给出“环路边界不能通过插入流水线寄存器来改变”这一结论。从重定时的角度也很容易理解，环路中插入流水线寄存器是做不到的。但是，下面即将介绍的技术，将“很诡异地”实现环路的流水化或并行化，

要思考的问题是：环路的流水和并行是如何实现的，在实现环路流水和并行的同时，系统其他非递归部分发生了什么“巨大的”变化。注意，第三章所讨论的流水线是前馈割集上插入寄存器，而环路是不存在前馈割集的，那么这里所介绍的流水又将如何理解呢？

第一节、 流水线交织操作

交织，顾名思义是有规律的交错，比如 12121212，就是 1 和 2 两个数字的交织，又如 ABCABCABC，就是 A、B 和 C 三个字母的交织。下面将要讨论的技术，将使得环路中出现这种有规律的数据交错现象。

以一阶 IIR 滤波器为例，迭代公式如下

$$y(n) = a \cdot y(n-1) + b \cdot x(n) \tag{1}$$

其电路结构见图 1，

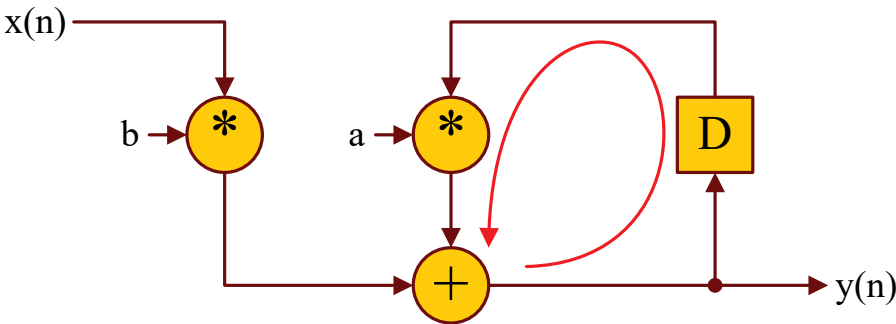


图 1 一阶 IIR 电路结构

如何才能增加图 1 环路（红线所示）中的延时个数，而又不破坏电路功能呢？其实最简单的办法就是 M 倍降速，比如要得到 3 级流水环路，进行 3 倍降速，新电路结构如下，

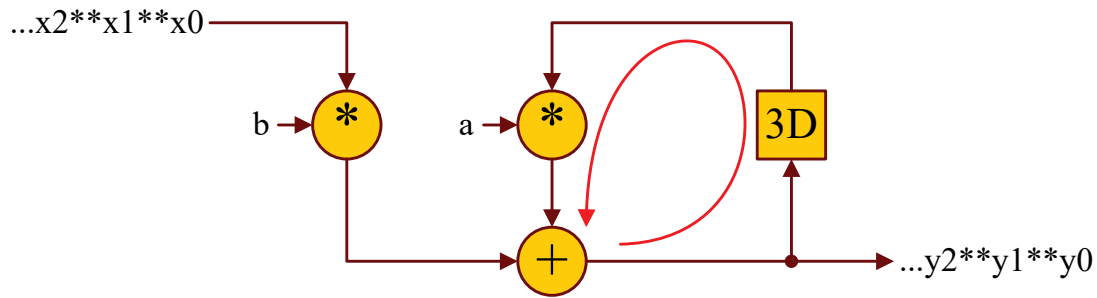


图 2 一阶 IIR 的 3 倍降速电路结构

不知大家是否还记得，什么叫  $M$  倍降速？ $M$  倍降速就是原 DFG 中的 1 个延时用  $M$  个延时取代（即，延时单元倍增到  $M$  倍），输入输出数据也随着“增采样”至  $M$  倍（\*表示无意义的占位，可以是 0 或其他）。从图 1 和图 2 的区别很容易理解  $M$  倍降速的含义。

图 1 的输入序列为  $\{...x_2x_1x_0\}$ ，图 2 的输入序列为  $\{...x_2**x_1**x_0\}$ ，其实也可看成数据在降速，原先一个周期一个有效数据，现在是三个周期一个有效数据。注意，第三章的前馈割集流水线技术不会导致数据降速，所以  $M$  倍降速所带来的“环路流水线”与一般的流水线是不太一样的。从迭代边界的角度，图 1 的迭代边界为  $T_a + T_m$ ，图 2 是  $(T_a + T_m)/3$ ，所以理论上图 2 电路时钟频率是图 1 电路的 3 倍。虽如此，但图 2 电路是三个周期处理一个有效数据，所以在吞吐率上并没带来任何改善。

仔细思考图 2 电路，其实每三个周期中有两个数据周期并没有被利用，白白浪费了。如果有三路独立的滤波数据，那么就能同时复合在同一电路（图 2）中，也就是

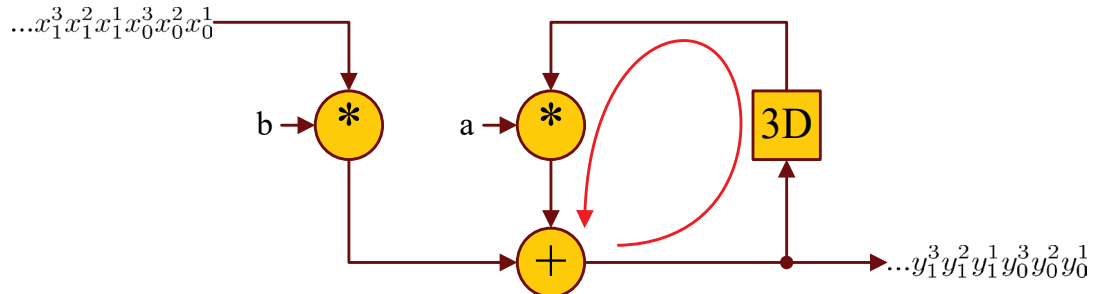


图 3 3 路独立数据的复合滤波

输入  $x_j^i$  表示第  $i$  路的第  $j$  个数据点，输出也类似。IIR 滤波器往往可以分解为多个 1 阶和 2 阶单元的级联或并联，这些单元的输入可以看成是多路独立的数据流，因此可以复合在同一电路结构中，见例 1 和例 2。

例1、 如图 4 的 2 个 1 阶 IIR 级联结构，构造一个 2 级流水的一阶 IIR 单元，并实现 2 路数据的复合。

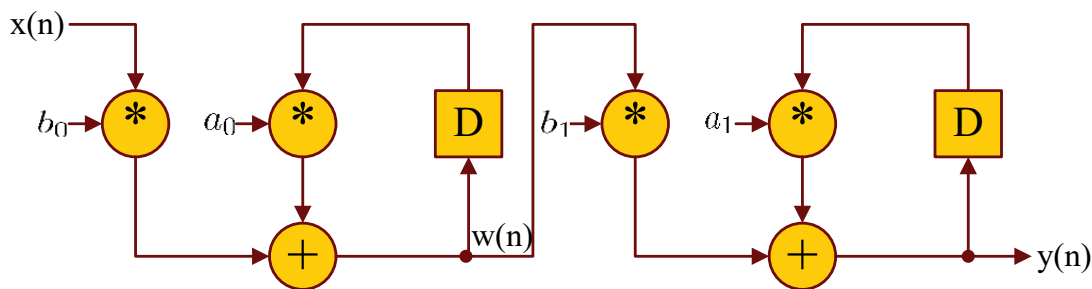


图 4 2 个一阶 IIR 单元的级联

只需在恰当的周期，将数据  $w_i$  选择作为输入即可实现 2 个一阶 IIR 单元的复合。复合电路的数据交织情况见表格 1，从中可以可知，在第-1 周期，第一个 IIR 单元的初始值  $w(-1)$  被置入 R0 寄存器，在第 0 周期，第二个 IIR 单元的初始值  $y(-1)$  被置入 R0 单元，偶数号周期，即第 0/2/4/6...周期输入 x 序列，而奇数号周期，即第 1/3/5/7...周期输入 w 序列，采样奇数号周期的输出即为 y 序列。

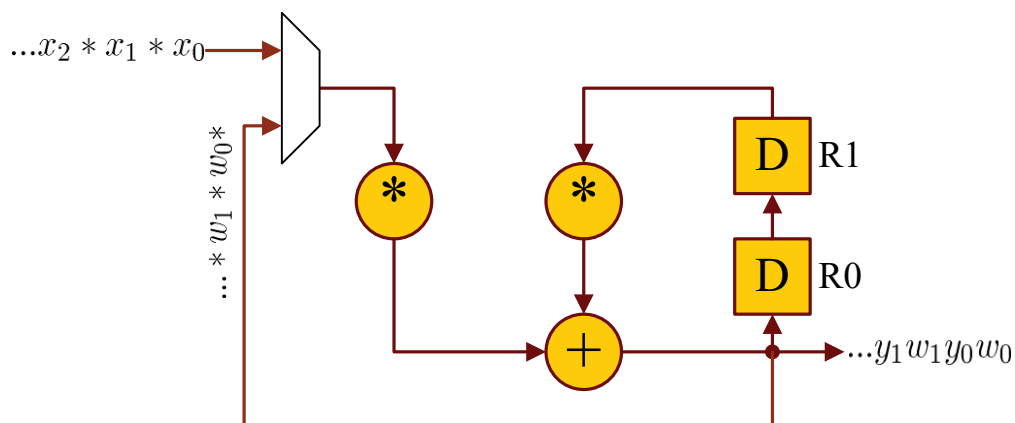


图 5 2 个一阶 IIR 单元复合 2 级流水环路结构

表格 1 2 个一阶 IIR 单元复合的数据交织表

-1	0	1	2	3	4	5	6	7	周期
	x(0)	w(0)	x(1)	w(1)	x(2)	w(2)	x(3)	w(3)	输入
w(-1)	y(-1)	w(0)	y(0)	w(1)	y(1)	w(2)	y(2)	w(3)	R0
	w(-1)	y(-1)	w(0)	y(0)	w(1)	y(1)	w(2)	y(2)	R1
		w(0)	y(0)	w(1)	y(1)	w(2)	y(2)	w(3)	输出

例2、 如图 5 的 2 个 1 阶 IIR 并联结构，构造一个 2 级流水的一阶 IIR 单元，并实现 2 路数据的复合。

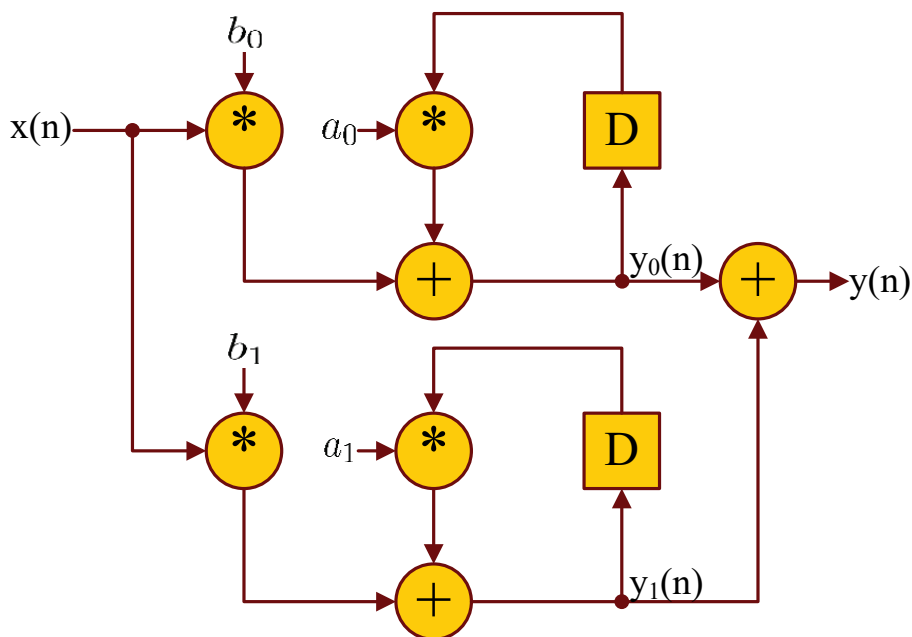


图 6 2 个一阶 IIR 单元的并联

从表格 2 可知，第-1 周期，将第一个 IIR 单元的初值  $y_0(-1)$  置入 R0，在第 0 周期将第二个 IIR 单元的初值置入 R0，对于 R2 之前的选择器，在偶数号周期锁存其前置加法器的结果，在奇数号周期锁存另一路结果，最终电路见图 7。

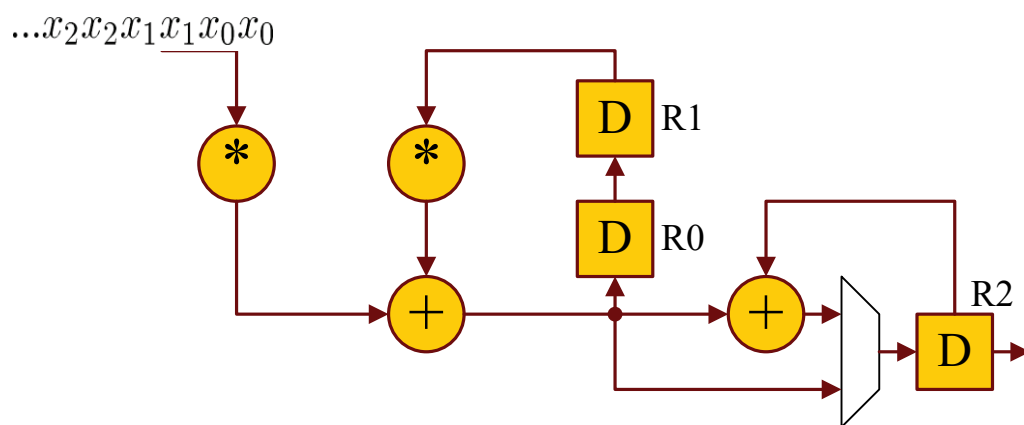


图 7 2 个一阶 IIR 单元复合 2 级流水环路结构

表格 2 2 个一阶 IIR 单元复合的数据交织表

-1	0	1	2	3	4	周期
	x(0)	x(0)	x(1)	x(1)	x(2)	输入
y0(-1)	y1(-1)	y0(0)	y1(0)	y0(1)	y1(1)	R0
	y0(-1)	y1(-1)	y0(0)	y1(0)	y0(1)	R1
		y0(0)	y0(0)+y1(0)	y0(1)	y0(1)+y1(1)	R3/输出

一般 IIR 滤波器多是分解成 2 阶节，对于 2 阶 IIR 节也可类似处理，见例 3 的 2 个级联 2 阶

节的情况。

例3、 如图 5 的 2 个 2 阶 IIR 级联结构，构造一个 2 级流水的 2 阶 IIR 单元，并实现 2 路数据的复合。

2 阶 IIR 节的迭代方程为

$$y(n) = a_1y(n-1) + a_2y(n-2) + b_0x(n) + b_1x(n-1) + b_2x(n-2) \quad (2)$$

将 2 个 2 阶节复合在一起，首先需要 2 倍降速处理，最终电路图如下（基于直接 II 型的结构），

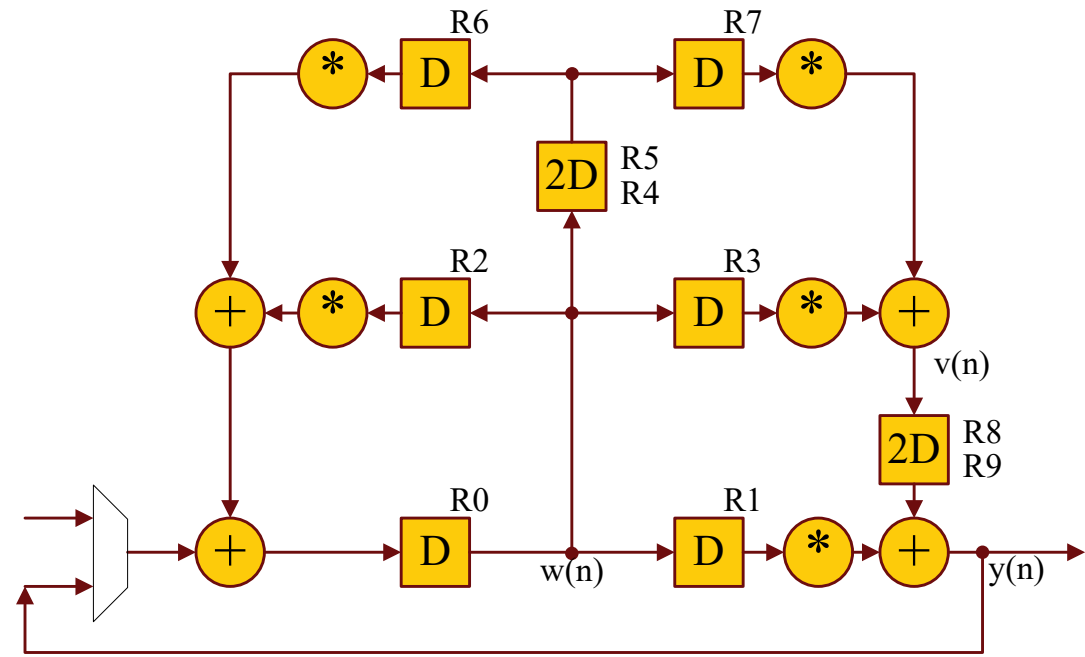


图 8 2 个 2 阶 IIR 单元复合 2 级流水环路结构

表格 3 2 个 2 阶 IIR 单元复合的数据交织表

-2	-1	0	1	2	3	4	周期
		x(0)		x(1)		x(2)	输入
	w0(-1)		w0(0)		w0(1)	w1(-1)	R0
		w0(-1)		w0(0)		w0(1)	R1
		w0(-1)		w0(0)		w0(1)	R2
		w0(-1)		w0(0)		w0(1)	R3
w0(-2)		w0(-1)		w0(0)	w1(-2)	w0(1)	R4
	w0(-2)		w0(-1)		w0(0)	w1(-2)	R5
		w0(-2)		w0(-1)		w0(0)	R6
		w0(-2)		w0(-1)		w0(0)	R7
			v0(-1)		v0(0)		R8
			v0(-1)			v0(0)	R9
			*			y0(0)	输出

5	6	7	8	9	10	11	周期
y0(0)	x(3)	y0(1)	x(4)	y0(2)	x(5)	y0(3)	输入

w (2)	w1 (0)	w0 (3)	w1 (1)	w0 (4)	w1 (2)	w0 (5)	R0
w1 (-1)	w0 (2)	w1 (0)	w0 (3)	w1 (1)	w0 (4)	w1 (2)	R1
w1 (-1)	w0 (2)	w1 (0)	w0 (3)	w1 (1)	w0 (4)	w1 (2)	R2
w1 (-1)	w0 (2)	w1 (0)	w0 (3)	w1 (1)	w0 (4)	w1 (2)	R3
w1 (-1)	w0 (2)	w1 (0)	w0 (3)	w1 (1)	w0 (4)	w1 (2)	R4
w0 (1)	w1 (-1)	w0 (2)	w1 (0)	w0 (3)	w1 (1)	w0 (4)	R5
w1 (-2)	w0 (1)	w1 (-1)	w0 (2)	w1 (0)	w0 (3)	w1 (1)	R6
w1 (-2)	w0 (1)	w1 (-1)	w0 (2)	w1 (0)	w0 (3)	w1 (1)	R7
v0 (1)	v1 (-1)	v0 (2)	v1 (0)	v0 (3)	v1 (1)	v0 (4)	R8
	v0 (1)	v1 (-1)	v0 (2)	v1 (0)	v0 (3)	v1 (1)	R9
	y0 (1)	*	y0 (2)	y1 (0)	y0 (3)	y1 (1)	输出

图 8 电路，为了把每个乘法单元做成 2 级流水节点，采用了重定时技术，从图中可以看出每个乘法器前面都有一个延时单元，将该延时整合到乘法器中即可得 2 级流水的乘法器。此外，为了保证复合的正确性，应该预先分析“数据交织”关系，如表 3 所示，从表中可知何时初始化 R0 和 R4 寄存器，何时将输出反馈回输入端等等，这样就可以构造恰当的控制逻辑，控制数据的正确复合。

**至此，做个小小结：** 以上讨论了基于 M 倍降速的递归流水线技术，通过 3 个例子讨论了 IIR 1 阶节复合和 2 阶节复合的具体做法，使用表格（excel）的方法列出数据交织的情况，可以帮助设计者更清晰了解复合的相关数据时序，从而构造正确的状态机。如果原来系统是稳定的，那么 M 倍降速之后仍然是稳定的（不要认为这是理所当然!!），此外要想 100% 发挥 M 倍降速硬件利用率，需要有（将）M 个独立输入通道复合在一起。

M 倍降速是非常不错的技术，简单易用！但是有一点令人不爽，就是只有存在 M 个独立通道复合时才能达到 100% 的硬件利用率；某些情况下，要求一定要 M 级流水，但又无 M 个独立通道需要处理，就会造成资源的浪费。为了弥补这个不足，发展了超前计算技术，但由超前计算导出的结构可能出现不稳定的情况，所以对新电路需要验证其稳定性。仍然以一阶和二阶 IIR 节为例来讨论。

对于一阶 IIR 节，很幸运得到的超前计算流水线是稳定的（前提是原系统是稳定的），但是二阶 IIR 节却不一定稳定。先来看看一阶 IIR 节的例子，原始迭代公式如下，

$$y(n) = a \cdot y(n-1) + b \cdot x(n) \quad (3)$$

所谓的超前计算是指，对公式(3)进行多次迭代，比如 2 步超前，如下

$$\begin{aligned}
y(n-1) &= a \cdot \boxed{y(n-2)} + b \cdot x(n-1) \\
&\quad \downarrow \\
y(n) &= a \cdot \boxed{y(n-1)} + b \cdot x(n) \\
&\quad \downarrow \\
y(n) &= a \cdot \{a \cdot [a \cdot y(n-3) + b \cdot x(n-2)] + b \cdot x(n-1)\} + b \cdot x(n) \\
&\quad \downarrow \\
y(n) &= a^3 \cdot y(n-3) + b \cdot x(n) + a \cdot b \cdot x(n-1) + a^2 \cdot b \cdot x(n-2) \\
&= a^3 \cdot y(n-3) + \sum_{i=0}^2 a^i \cdot b \cdot x(n-i)
\end{aligned}$$

图 9 一阶 IIR 节的 2 步超前

推而广之，M-1 步超前，就是用

$$y(n-i) = a \cdot y(n-i-1) + b \cdot x(n-i) \quad (4)$$

代入

$$y(n-i+1) = a \cdot y(n-i) + b \cdot x(n-i+1) \quad (5)$$

其中  $i = 1, 2, \dots, M-1$ ，最终得到的公式为

$$y(n) = a^M \cdot y(n-M) + \sum_{i=0}^{M-1} a^i \cdot b \cdot x(n-i) \quad (6)$$

将公式(6)转化为传递函数，有

$$H(e^{j\omega}) = \frac{\sum_{i=0}^{M-1} a^i \cdot b \cdot z^{-i}}{1 - a^M \cdot z^{-M}} \quad (7)$$