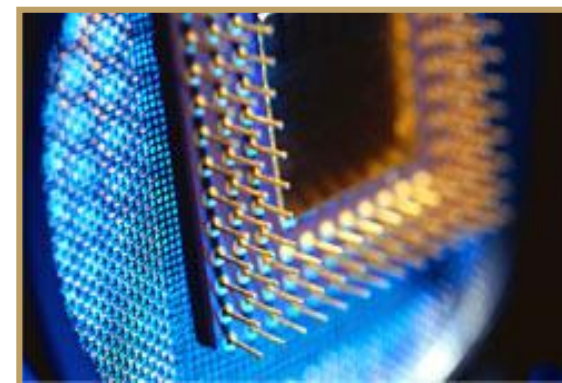
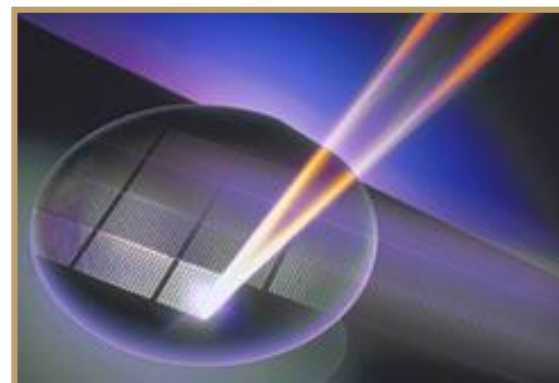
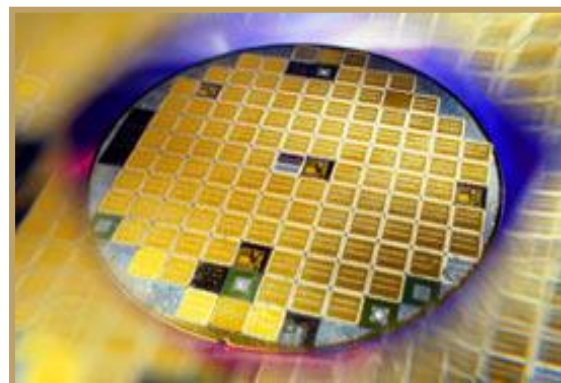




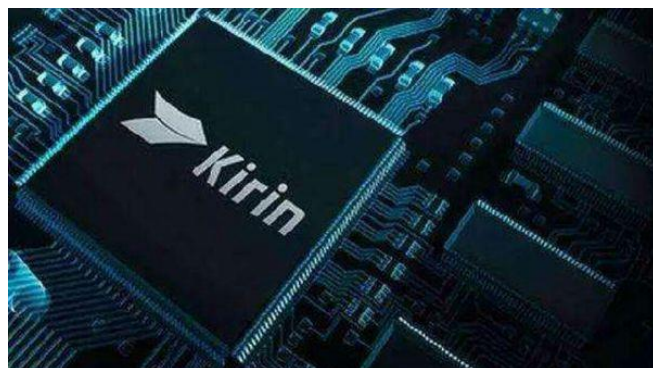
《VLSI数字通信原理与设计》课程

主讲人 贺光辉

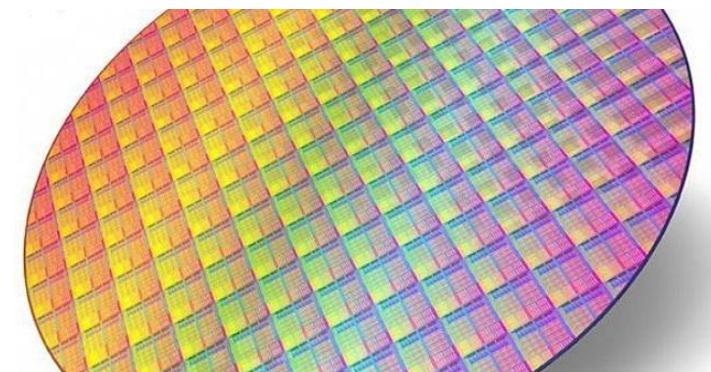
第六章：折叠



芯片的面积越来越大



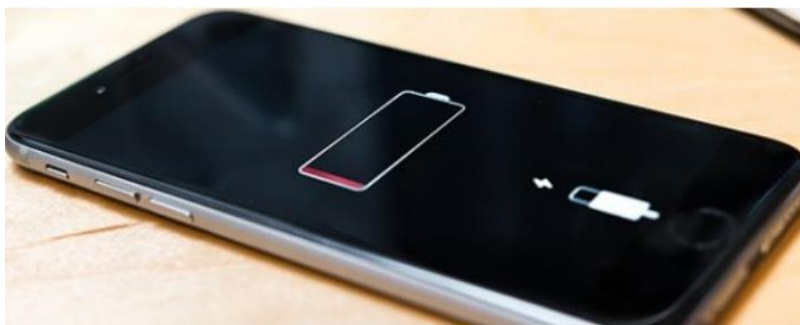
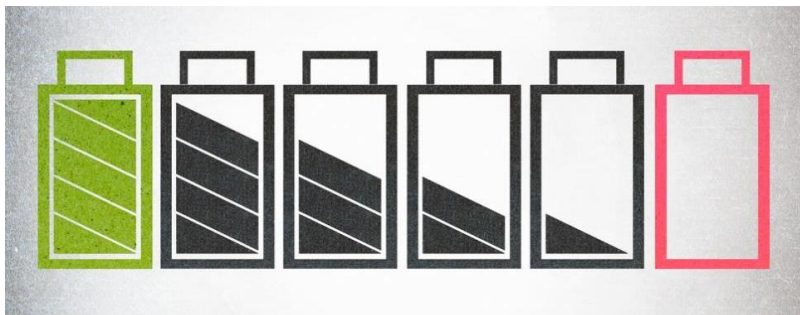
	麒麟970	骁龙835	苹果A10
晶体管数 (TSMC 10nm)	55亿	31亿	33亿



芯片面积越大
成本越高!

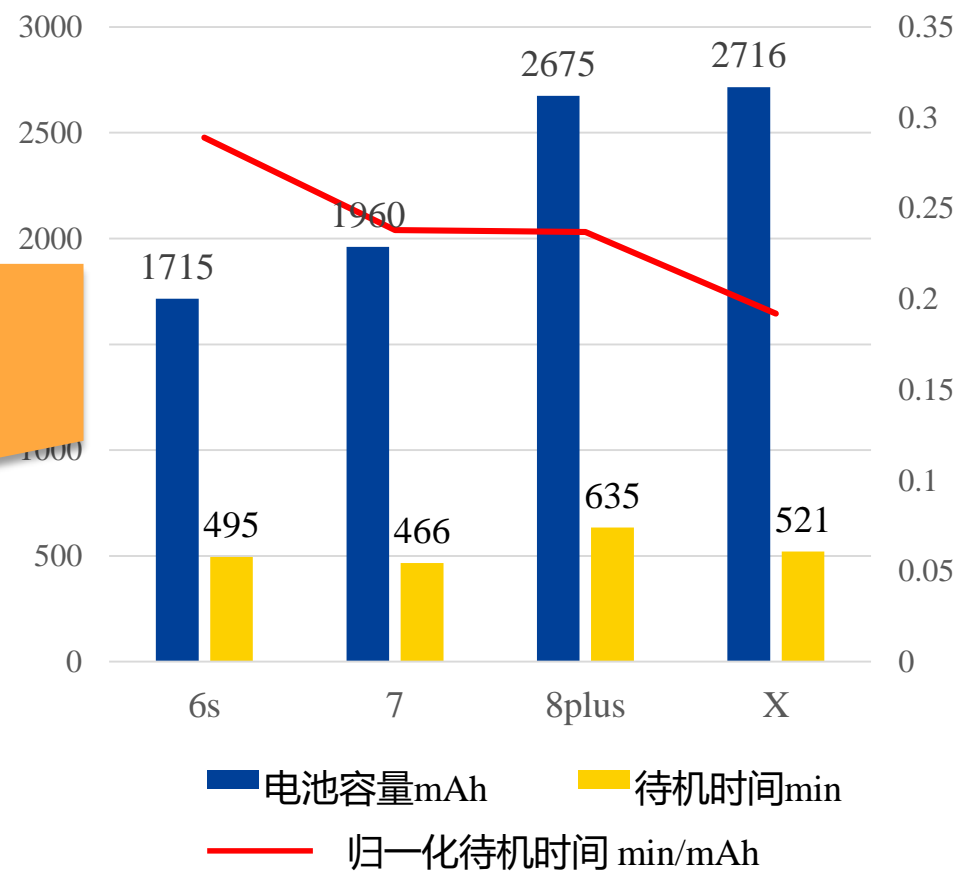
平均每块芯片(147mm²)生产成本22美元

移动设备耗电大



低功耗

iPhone手机电池调研



■ 不可兼得

- 采用流水线提高速度，但会增大面积和功耗

■ 有什么技术能降低面积和功耗？





目录

01 折叠的基本概念

02 折叠变换

03 寄存器最小化技术

04 折叠架构的寄存器最小化

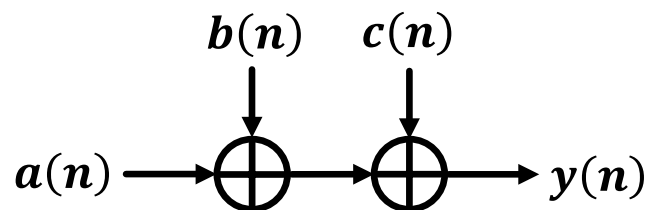
05 本章总结



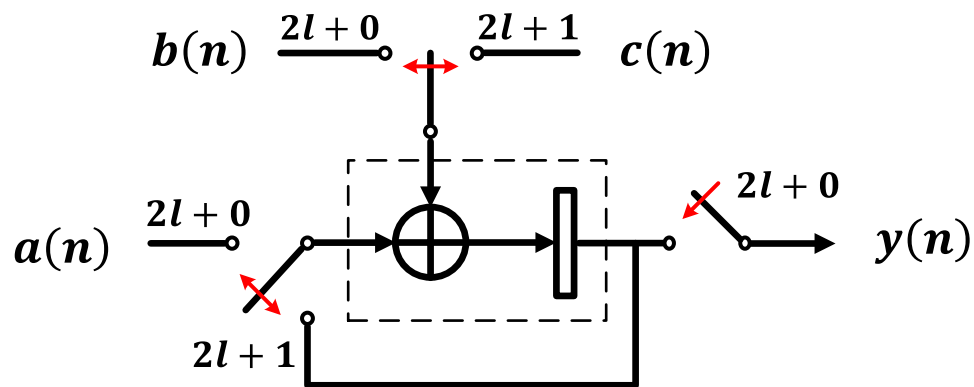
01. 折叠的基本概念——折叠是什么 (1)

■ 举例：如何设计 $y(n) = a(n) + b(n) + c(n)$

最简单思路，直接累加



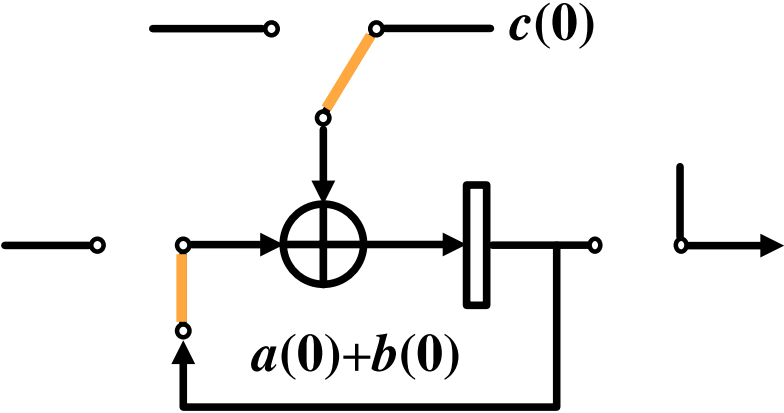
通过如下电路将两个加法器缩减为一个加法器



如何减小面积

01. 折叠的基本概念——折叠是什么（2）

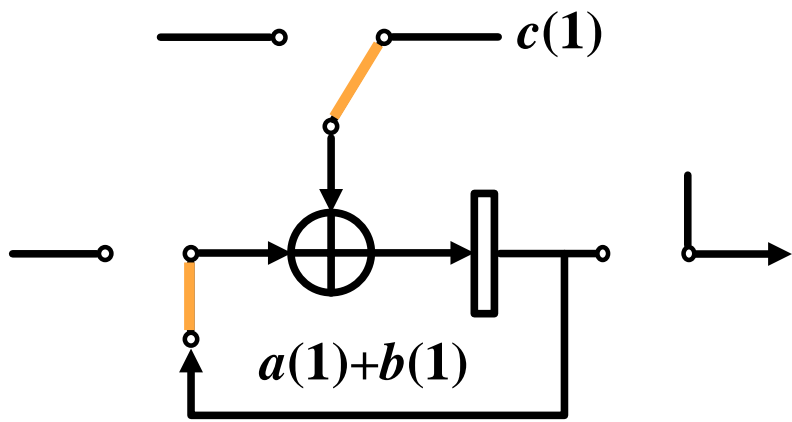
■ 工作过程



时刻	左侧输入	上侧输入	输出 $y(n)$
周期0	$a(0)$	$b(0)$	0
周期1	$a(0) + b(0)$	$c(0)$	-

01. 折叠的基本概念——折叠是什么（2）

工作过程



每2周期产生1次输出

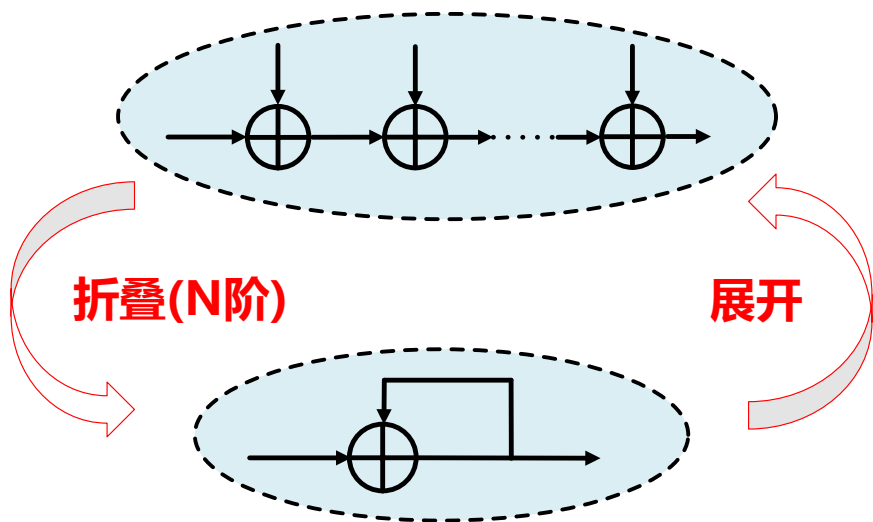
时刻	左侧输入	上侧输入	输出 $y(n)$
周期0	$a(0)$	$b(0)$	0
周期1	$a(0) + b(0)$	$c(0)$	-
周期2	$a(1)$	$b(1)$	$a(0) + b(0) + c(0)$
周期3	$a(1) + b(1)$	$c(1)$	-
周期4	$a(2)$	$b(2)$	$a(1) + b(1) + c(1)$
周期5	$a(2) + b(2)$	$c(2)$	-
.....			

01. 折叠的基本概念——折叠的定义

■ 折叠(Folding): 展开的逆过程

把多个相同运算操作，通过**时分复用**在单个功能单元(如加法器和乘法器)上执行，达到**资源共享**

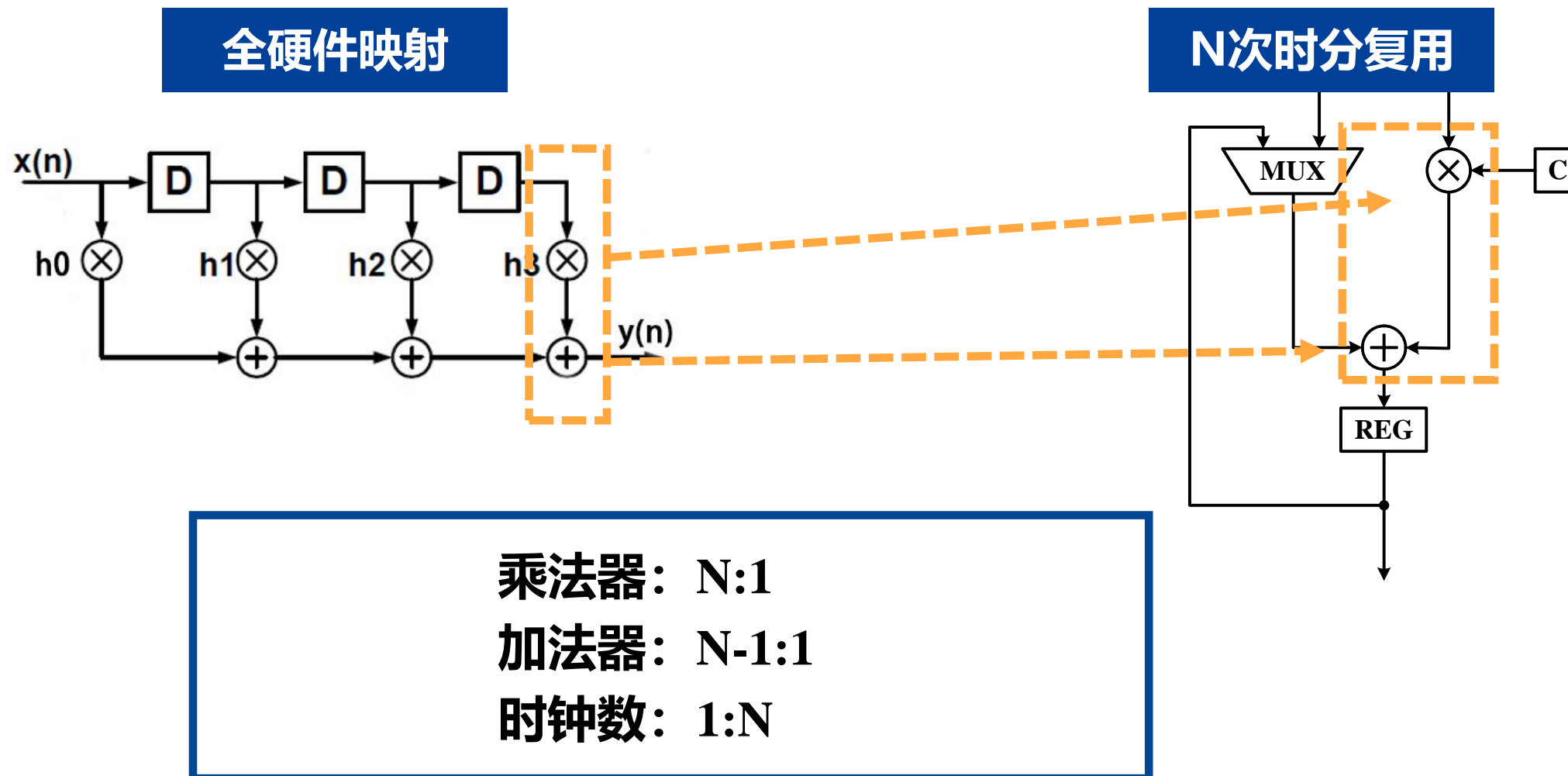
■ 目的：减少功能单元数目，从而减少面积



- 硬件开销
功能单元减少为 $1/N$
- 速度代价
处理时间增加为 N 倍

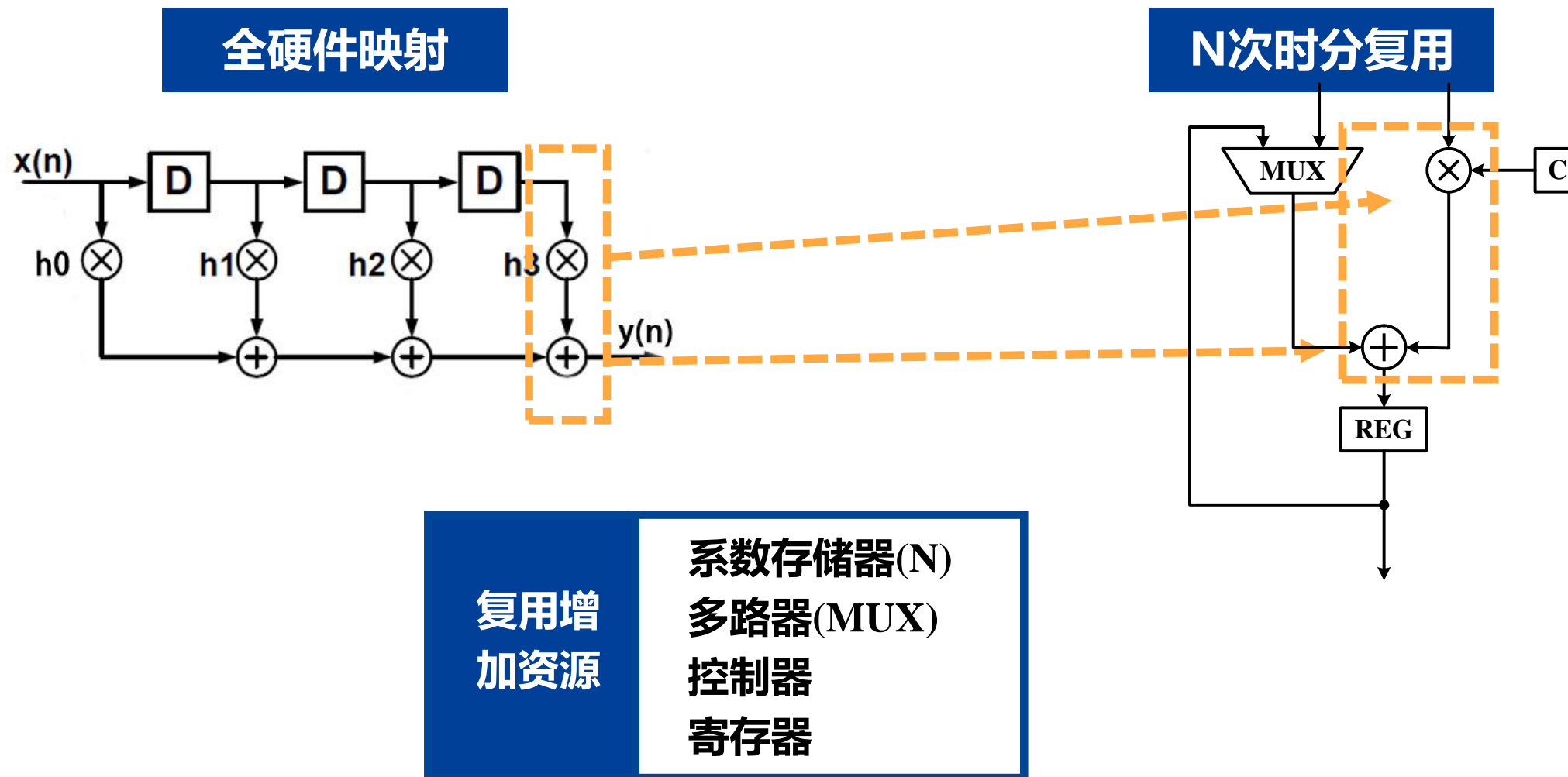
01. 折叠的基本概念——示例（1）

■ N阶折叠FIR滤波器： $y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$



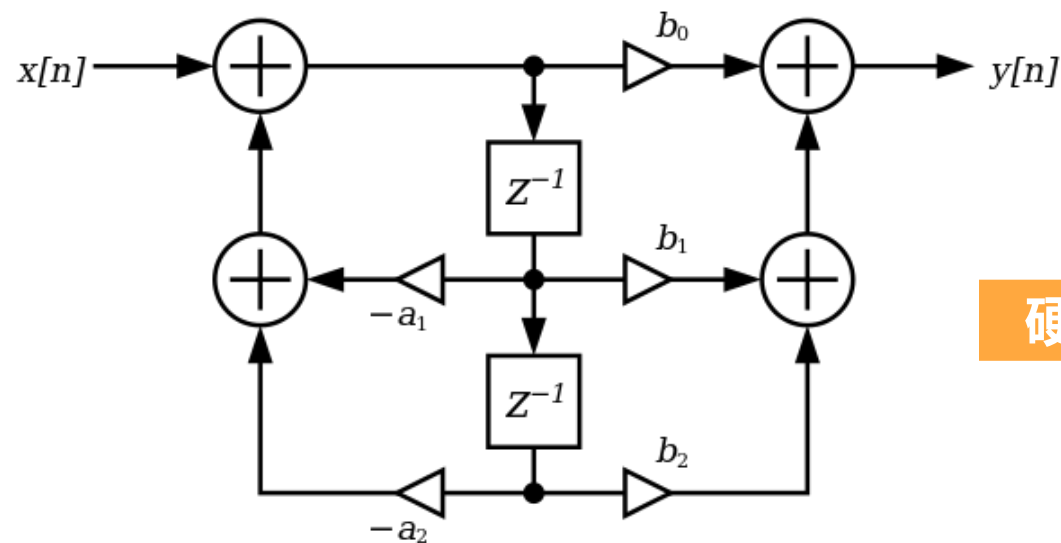
01. 折叠的基本概念——示例（1）

■ N阶折叠FIR滤波器： $y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$



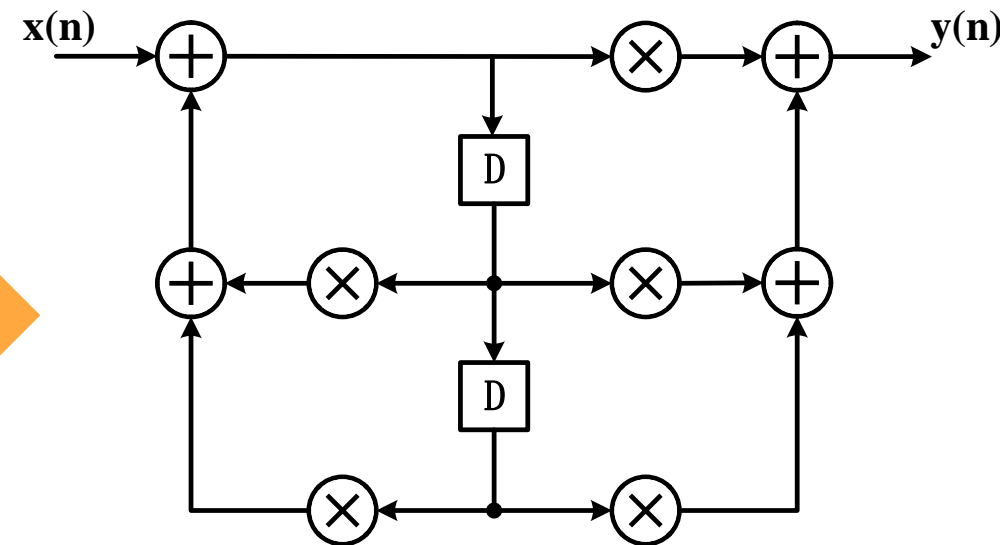
01. 折叠的基本概念——示例（2）

■ 微码折叠2阶节(Biquad)滤波器: $H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$



直接形式

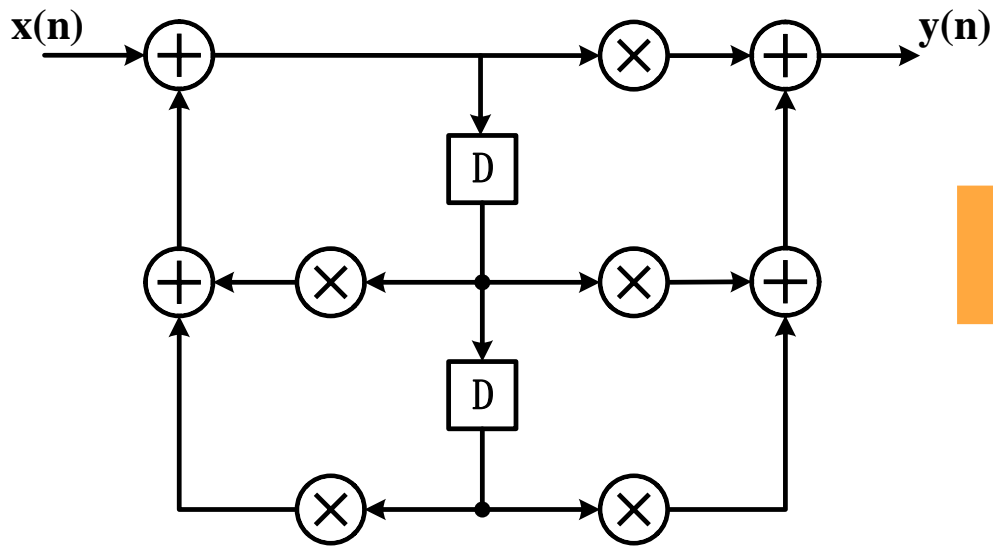
硬件映射



框图

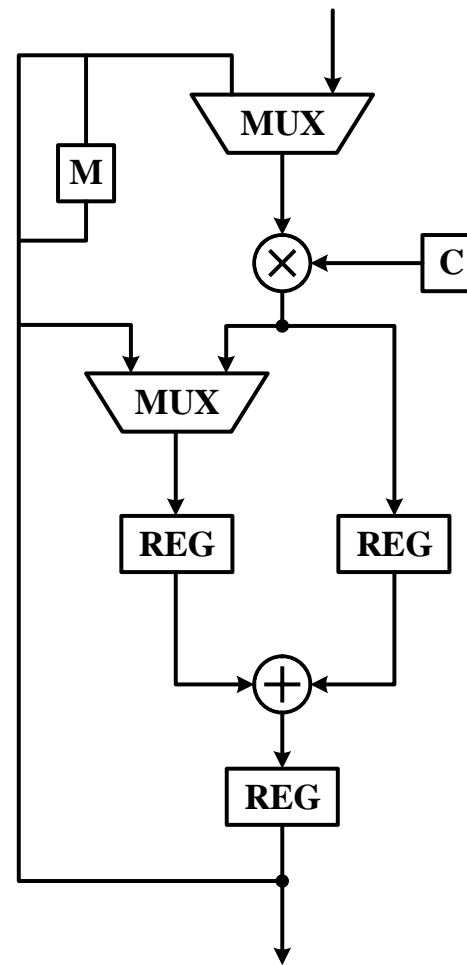
01. 折叠的基本概念——示例 (2)

■ **微码折叠2阶节(Biquad)滤波器：** $H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$



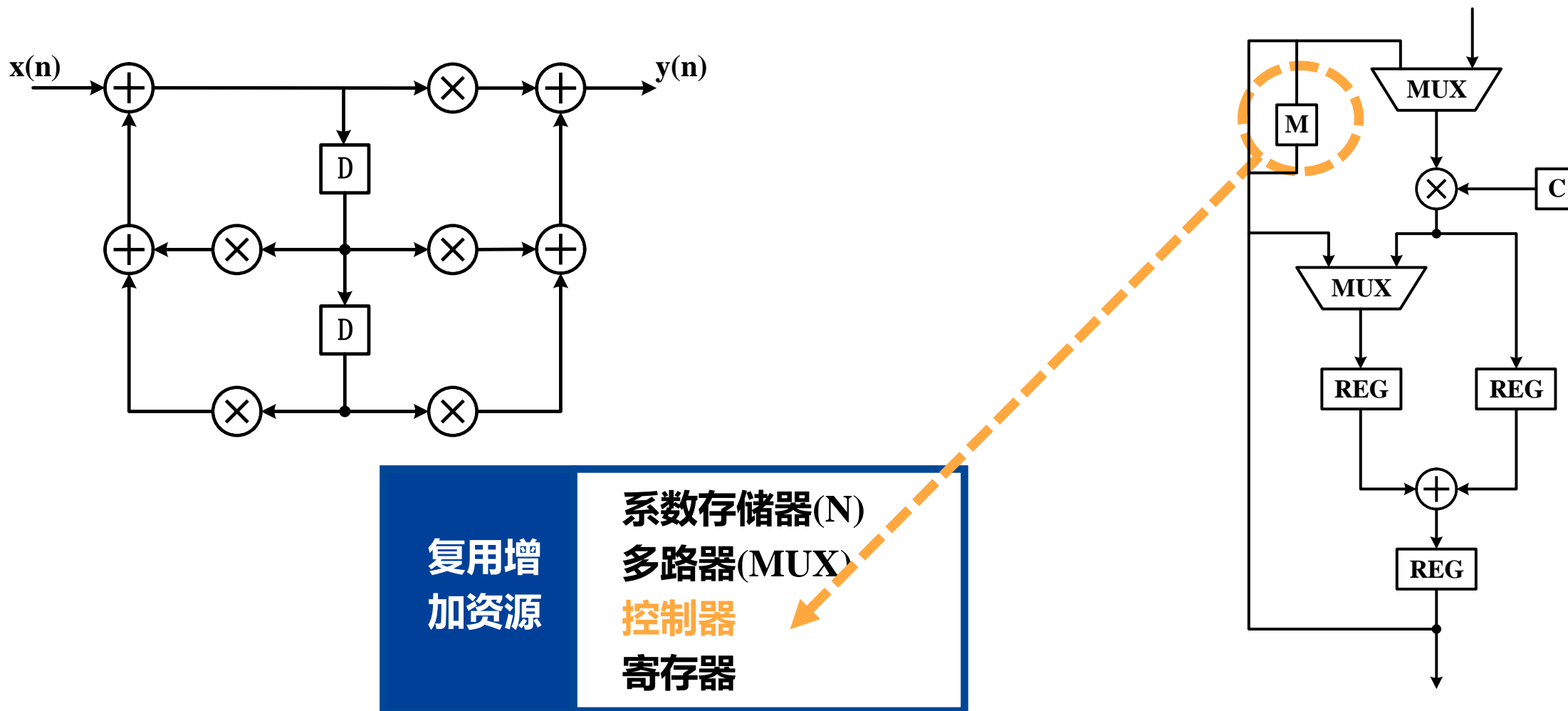
折叠

乘法器: 5:1
加法器: 4:1
时钟数: 1:5



01. 折叠的基本概念——示例 (2)

■ **微码折叠2阶节(Biquad)滤波器：** $H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$



01. 折叠的基本概念——N阶折叠说明

■ 硬件开销:

- 功能单元减少为 $1/N$
- 需要增加额外的寄存器, 多路器, 控制器等

■ 速度代价:

- 计算时间增加 N 倍, 迭代周期增加 N 倍, 产生迟滞 N 个时钟周期
- 输入样点需在 N 个时钟周期内保持有效

■ 通常DSP程序要复杂得多, 需要有系统的折叠技术理论



目录

01 折叠的基本概念

02 折叠变换

03 寄存器最小化技术

04 折叠架构的寄存器最小化

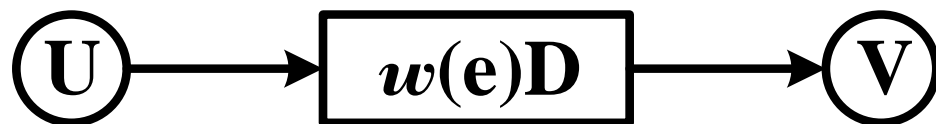
05 本章总结



02. 折叠变换 —— 定义

定义:

- 考虑节点U到V延迟为 $w(e)$ 的边 $U \rightarrow V$ ，其功能单元分别写为 H_u 和 H_v
- 折叠因子N：折叠到单个功能单元的运算数目
- 折叠序号u、v：节点U、V在硬件中调度执行的时间划分， $0 \sim N-1$
 - u、v是时间分割的顺序标号，节点U、V按此顺序被调度执行



02. 折叠变换 —— 折叠方程

考虑延迟为 $w(e)$ 的边 $U \rightarrow V$, 折叠算法调度 U 和 V 的第 l 次迭代分别在 $Nl+u$ 和 $Nl+v$ 时刻执行

- 若 H_u 包含 P_u 级流水线(1级表示无流水), 则 U 的输出在 $Nl+u+P_u$ 时刻产生
- 对边 $U \rightarrow V$, U 第 l 次迭代结果将被 V 第 $[l + w(e)]$ 次迭代使用
- 折叠后, U 在 $Nl+u+P_u$ 时刻产生输出, 在 $N[l + w(e)]+v$ 时刻被 V 使用

折叠方程: 表示中间结果必须存储的时间, 称折叠延迟

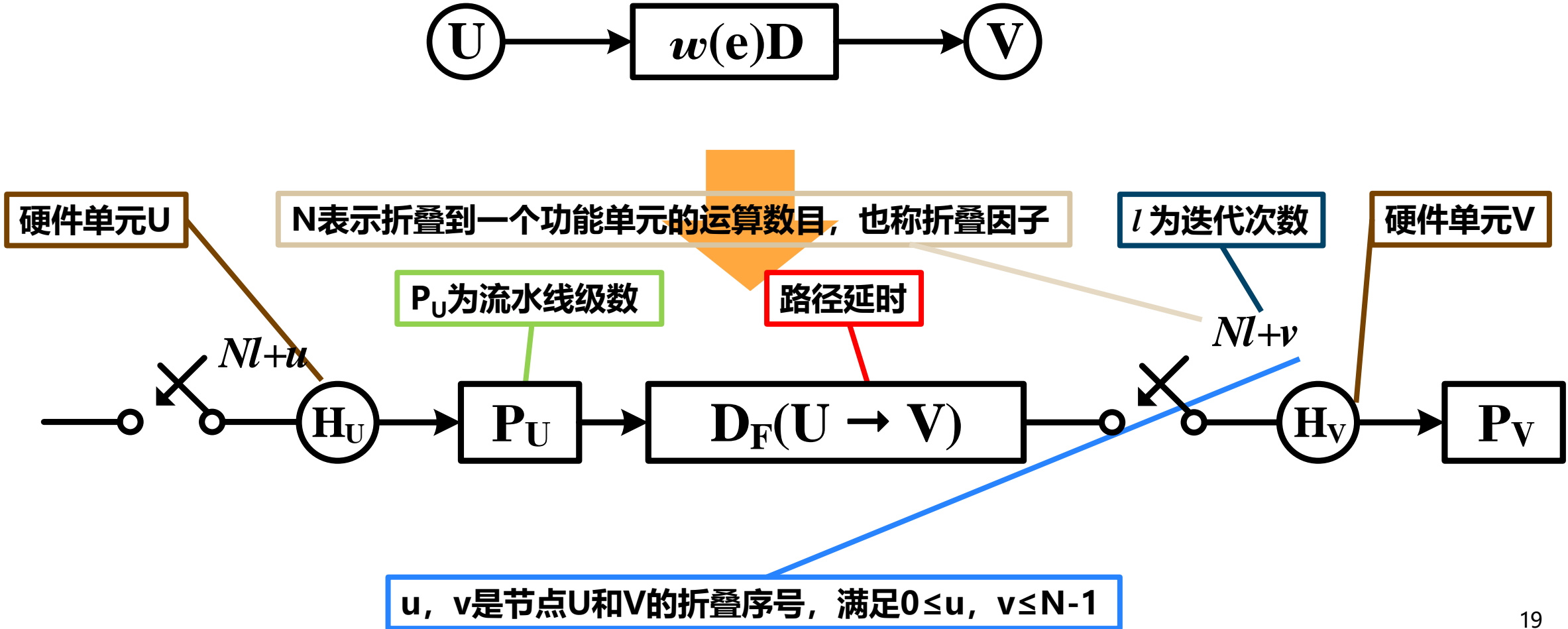
- $D_F(U \rightarrow V) = N[l + w(e)] + v - (Nl + u + P_u)$



$$D_F(U \rightarrow V) = Nw(e) - P_u + v - u$$

02. 折叠变换 —— 折叠方程

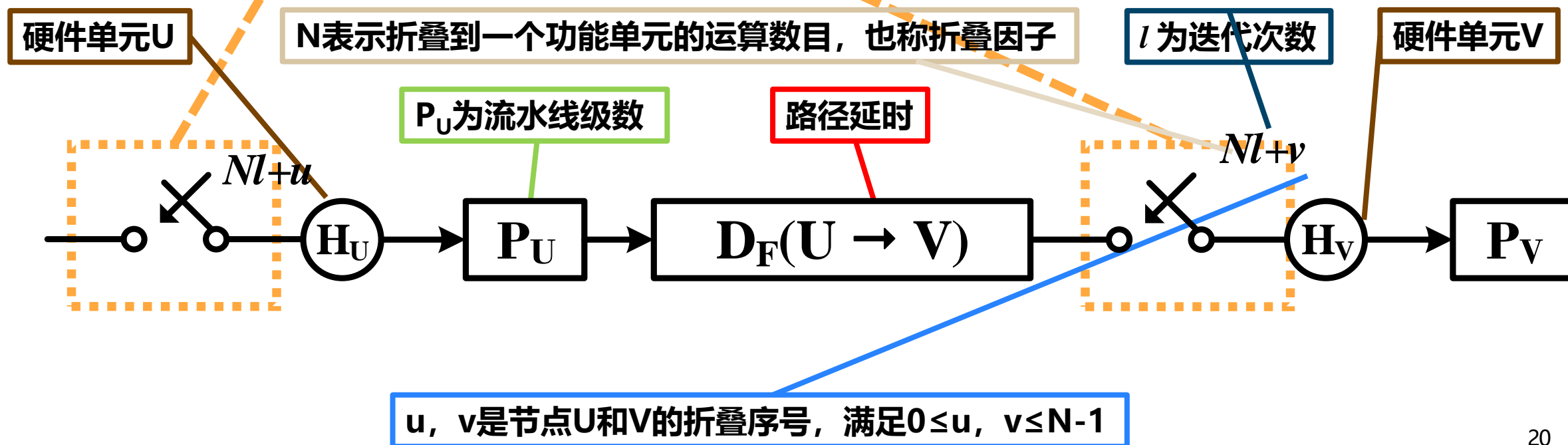
说明：原始DFG的边变换到折叠后DFG的边



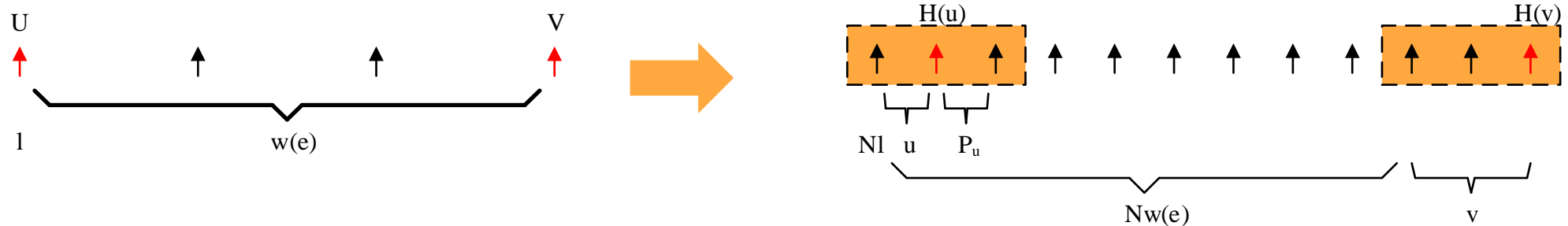
02. 折叠变换 —— 折叠方程

说明：原始DFG的边变换到折叠后DFG的边

- 考虑增加了两个受控开关
- 边的延迟变化(由折叠方程得到)： $D_F(U \rightarrow V) = Nw(e) - P_u + v - u$
- 边延迟的变化与迭代次数(编号 l)无关



02. 折叠变换 —— 折叠方程



$$D_F(U \xrightarrow{e} V) = [N(l + w(e)) + v] - [Nl + u + P_U] = Nw(e) - P_U + v - u$$

02. 折叠变换 —— 折叠集

定义：执行相同运算功能操作的有序集

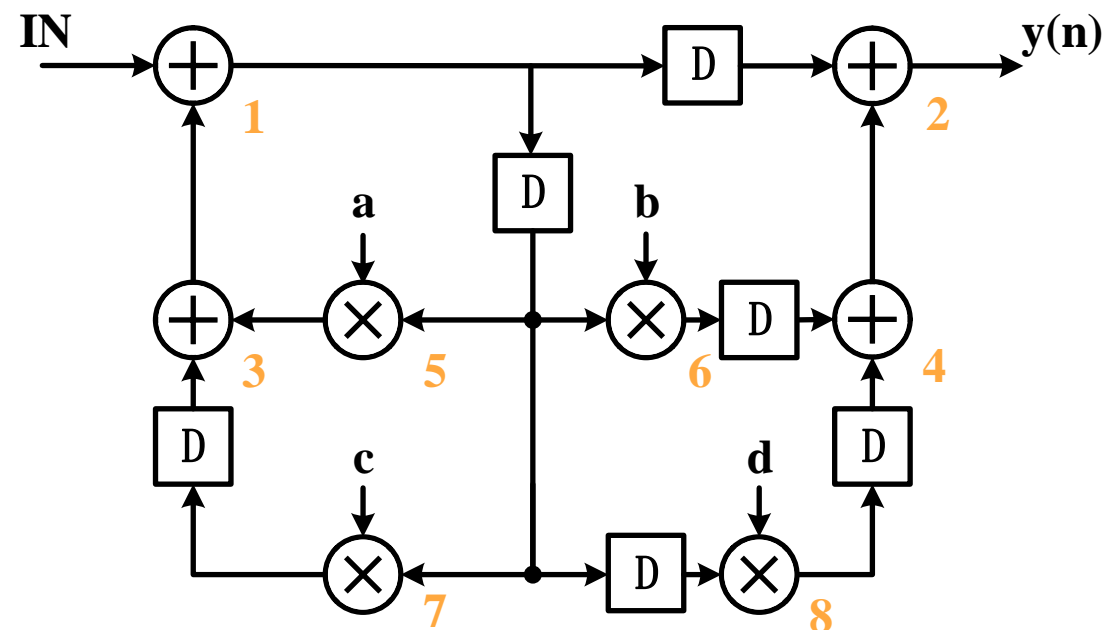
- 考虑包含 N 个元素，表示 N 个运算，其中的一些可能是空运算
- 折叠集中处于第 $j(0, 1 \dots N-1)$ 个位置的运算，由功能单元在第 j 个时间段执行
- 例子： $N=3$ 的折叠集 $S1=\{A1, 0, A2\}$
 - $A1$ 运算属于折叠集 $S1$ 的位置 0 ，其折叠序号为 0 ，符号为 $(S1|0)$ ，表示 $A1$ 运算在 $3l+0$ 时刻执行
 - $A2$ 运算属于折叠集 $S1$ 的位置 2 ，其折叠序号为 2 ，符号为 $(S1|2)$ ，表示 $A2$ 运算在 $3l+2$ 时刻执行
 - $S1$ 中位置 1 为空操作 $(S1|1=0)$ ，表示功能单元在 $3l+1$ 时刻没有用

02. 折叠变换 —— Biquad滤波器

8个节点（橙色标号）

- 加法器4个， $T_A=1ut$ ， $P_A=1$
- 乘法器4个， $T_M=2ut$ ， $P_M=2$

11条边，6个寄存器

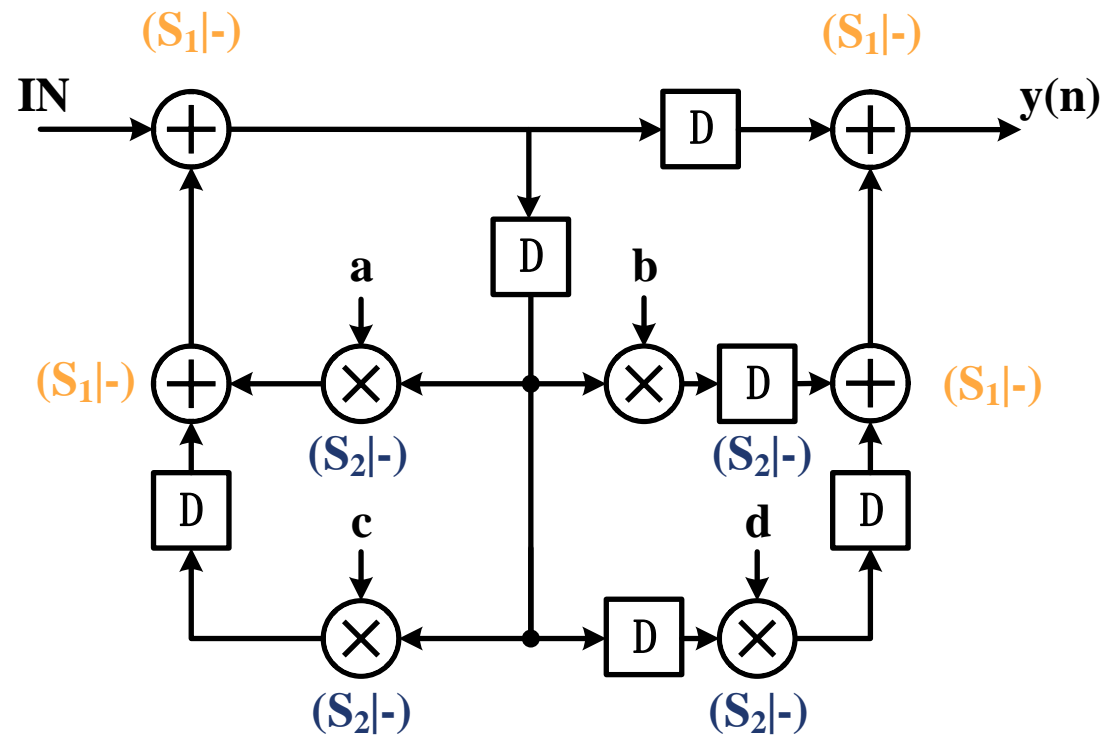


02. 折叠变换 —— Biquad滤波器

分配与指定

- 分配一个加法器单元, S_1
- 分配一个乘法器单元, S_2

分配比较容易



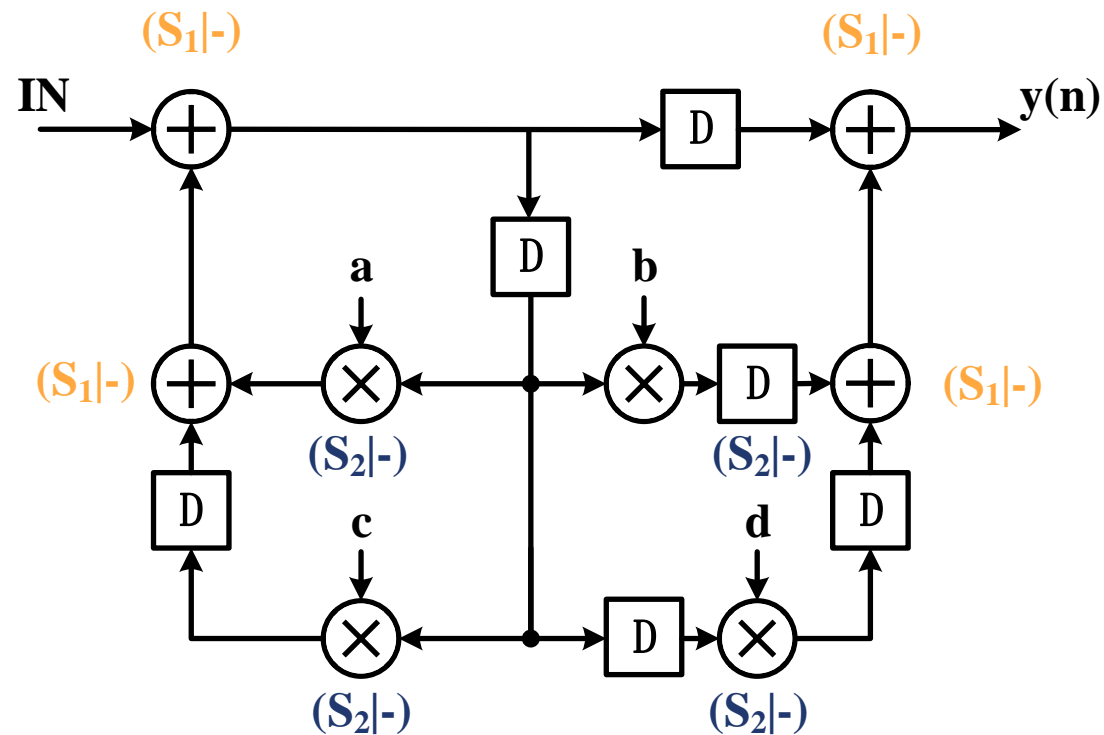
02. 折叠变换 —— Biquad 滤波器

调度

- 折叠因子 = 4



- 折叠序号为 0,1,2,3



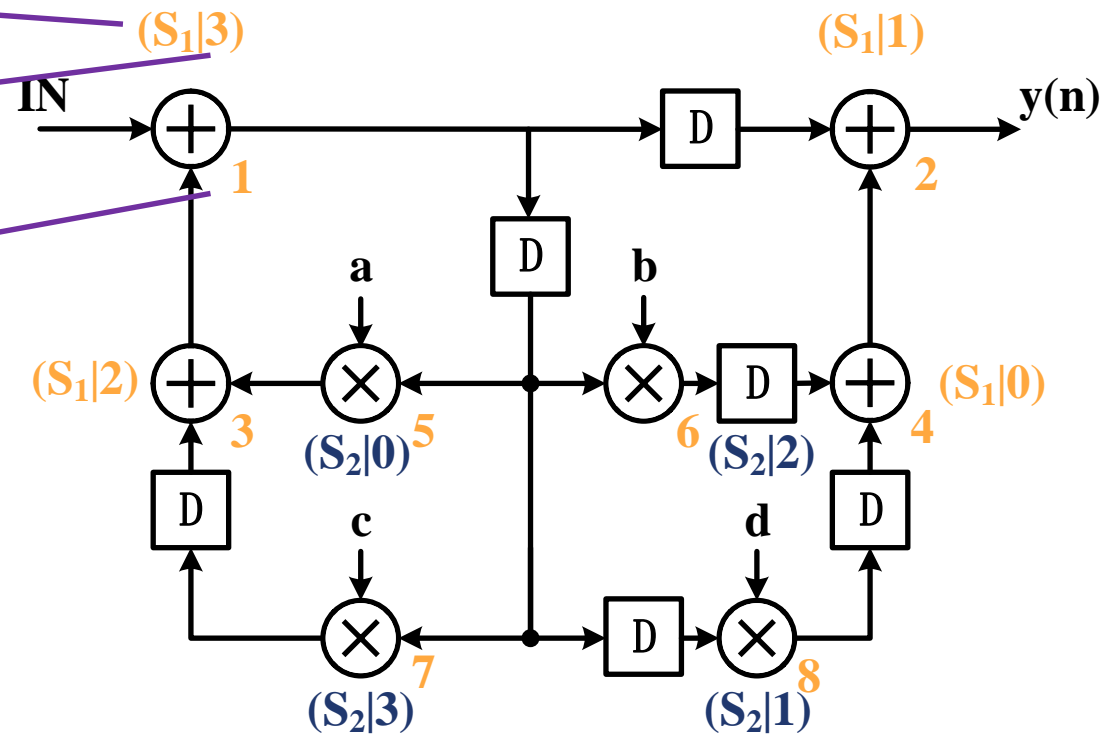
02. 折叠变换 —— Biquad滤波器

计算单元1

折叠序号

节点号

如何得到的该折叠序号？
它是否正确？



02. 折叠变换 —— Biquad滤波器

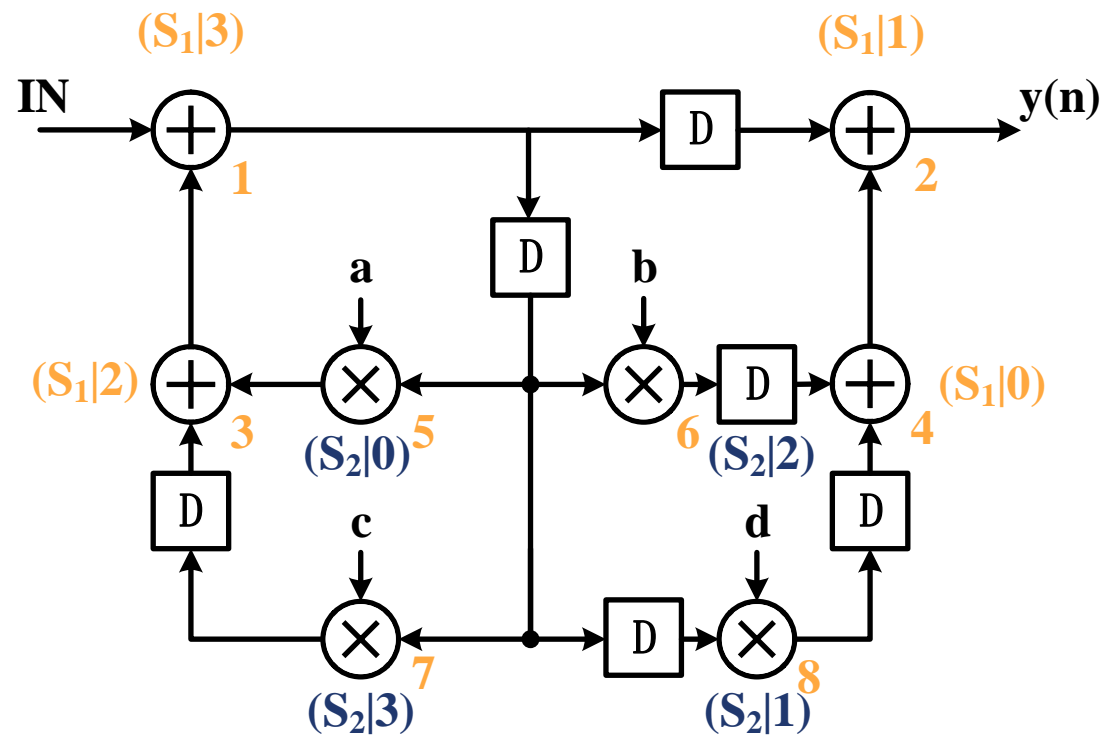
以 $N=4$ 为折叠因子进行折叠

- $N=4$ 意味着

- 折叠后每个功能单元的时钟周期可为 1 ut , 执行DSP程序中4次运算
- 折叠后的硬件迭代周期为 4 ut , 即双2阶节滤波器的原DFG中每个节点采用折叠结构后每 4 ut 执行一次

- 指定两个折叠集：加法折叠集；乘法折叠集

- $S1=\{4, 2, 3, 1\}$, 加法运算, 由同一个硬件加法器执行
- $S2=\{5, 8, 6, 7\}$, 乘法运算, 由同一个硬件乘法器执行



02. 折叠变换 —— Biquad滤波器

列出11条边的折叠方程组:

$$D_F(U \rightarrow V) = Nw(e) - P_u + v - u$$

$$D_F(1 \rightarrow 2) = 4(1) - 1 + 1 - 3 = 1$$

$$D_F(1 \rightarrow 5) = 4(1) - 1 + 0 - 3 = 0$$

$$D_F(1 \rightarrow 6) = 4(1) - 1 + 2 - 3 = 2$$

$$D_F(1 \rightarrow 7) = 4(1) - 1 + 3 - 3 = 3$$

$$D_F(1 \rightarrow 8) = 4(2) - 1 + 1 - 3 = 5$$

$$D_F(3 \rightarrow 1) = 4(0) - 1 + 3 - 2 = 0$$

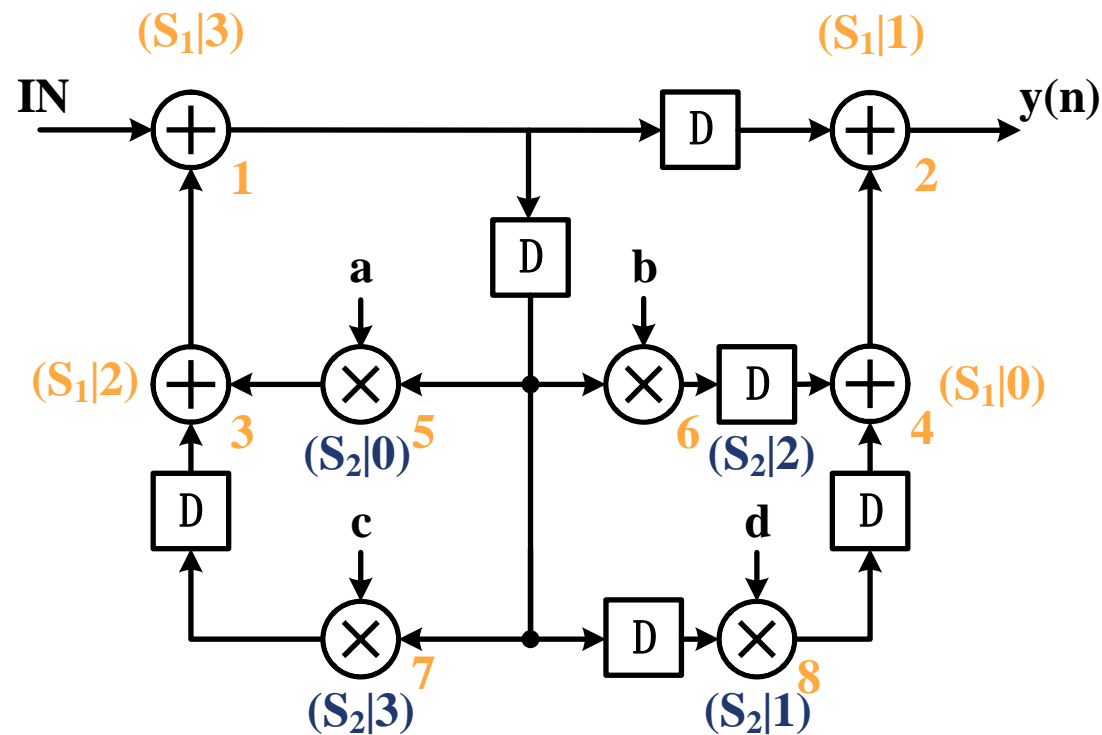
$$D_F(4 \rightarrow 2) = 4(0) - 1 + 1 - 0 = 0$$

$$D_F(5 \rightarrow 3) = 4(0) - 2 + 2 - 0 = 0$$

$$D_F(6 \rightarrow 4) = 4(1) - 2 + 0 - 2 = 0$$

$$D_F(7 \rightarrow 3) = 4(1) - 2 + 2 - 3 = 1$$

$$D_F(8 \rightarrow 4) = 4(1) - 2 + 0 - 1 = 1$$



02. 折叠变换 —— Biquad滤波器

$D_F(U \rightarrow V) = Nw(e) - P_u + v - u$

• $D_F(1 \rightarrow 6) = 4(1) - 1 + 2 - 3 = 2$

合理的折叠

$D_F(U \rightarrow V) \geq 0$

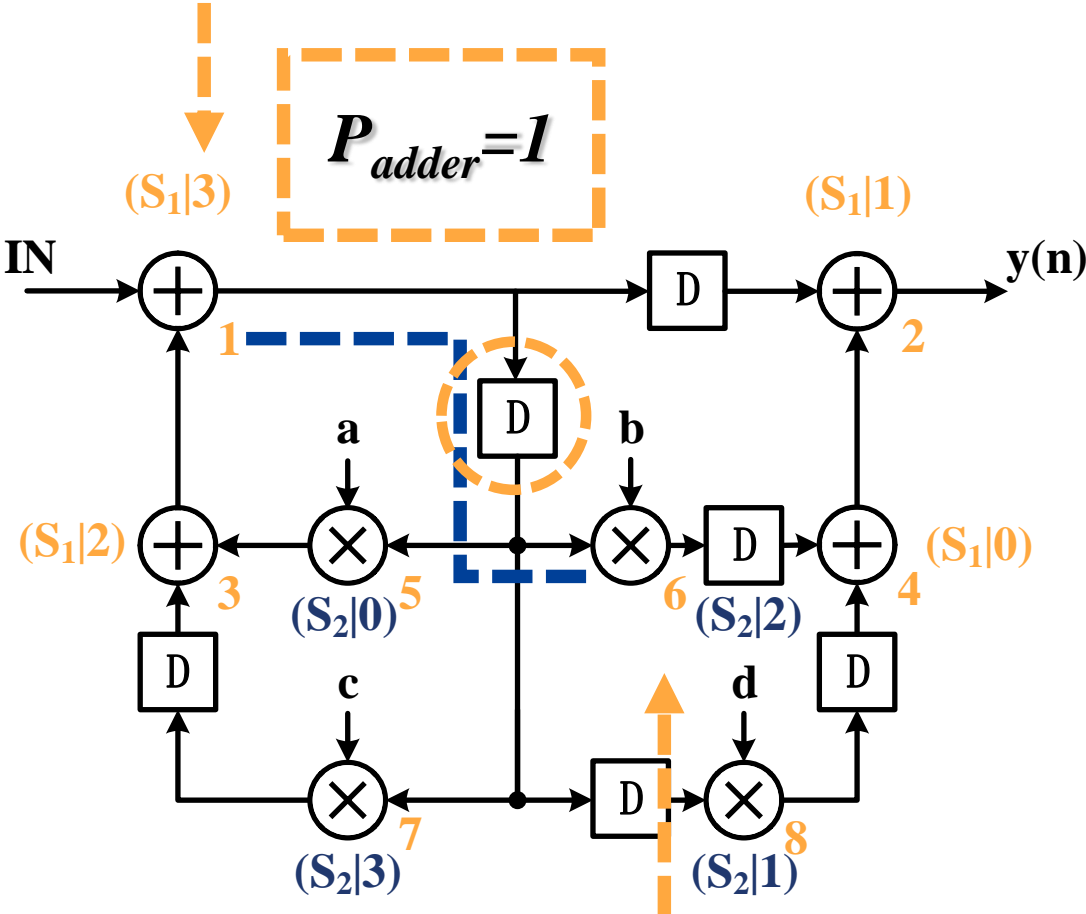
折叠因子=4

路径延迟=1

流水线级数=1

v折叠序号=2

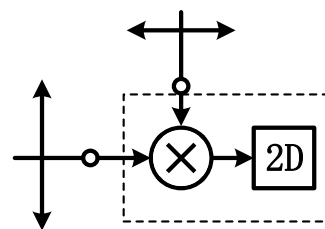
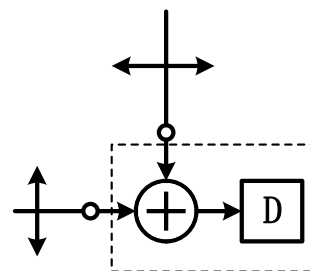
u折叠序号=3



02. 折叠变换 —— Biquad滤波器

折叠结构:

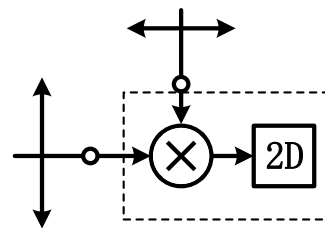
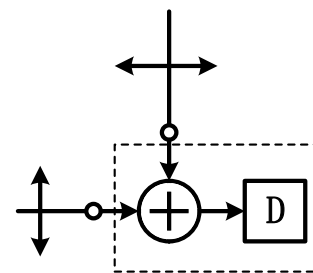
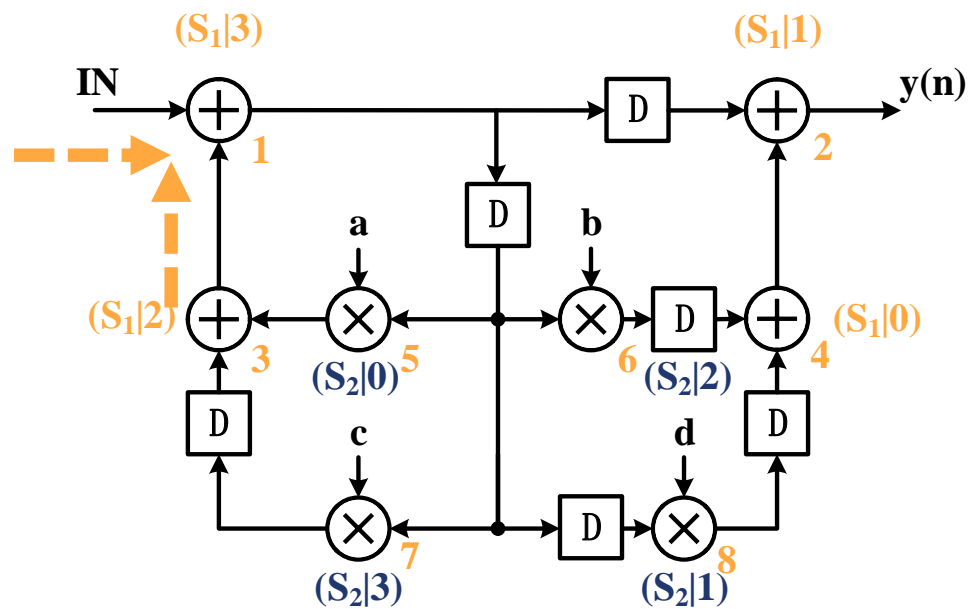
- 一个可实现的折叠系统DFG中所有边都必须满足 $D_F(U \rightarrow V) \geq 0$,
- 该方案所有边都满足 $D_F(U \rightarrow V) \geq 0$, 是可行的
- 画折叠DFG的功能单元、寄存器和多路开关
- 按折叠方程连接各边、标注开关的折叠序号



02. 折叠变换 —— Biquad滤波器

考察节点1 ($S_1|3$) 输入:

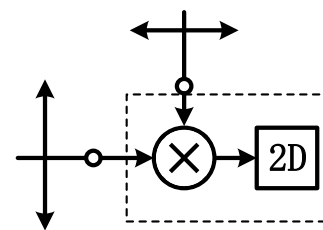
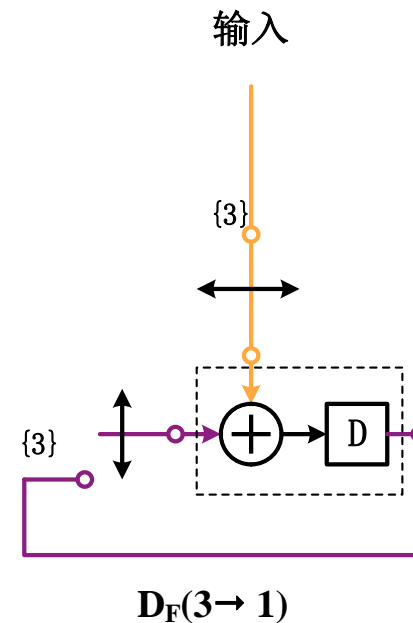
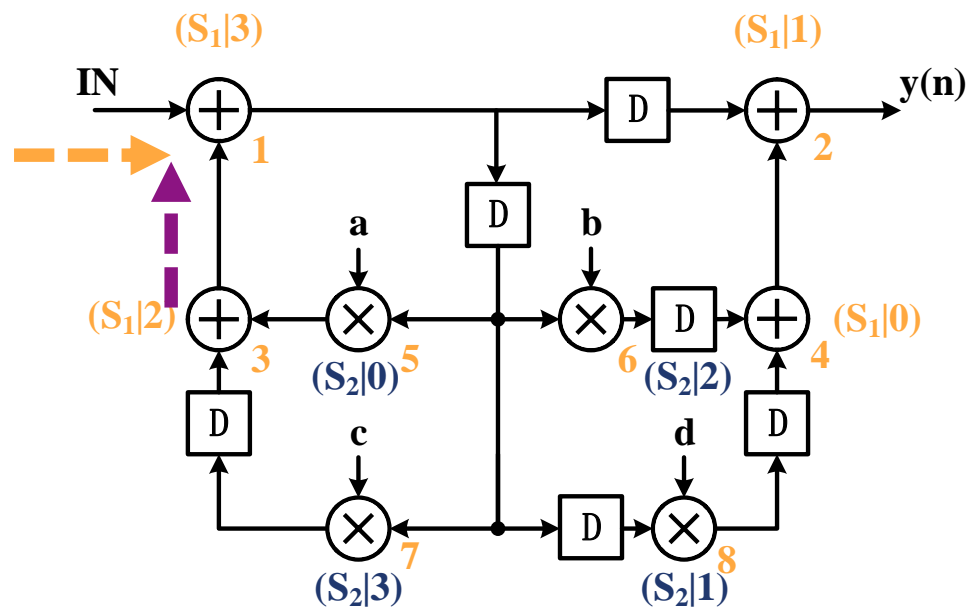
- 节点3: $D_F(3 \rightarrow 1) = 4(0) - 1 + 3 - 2 = 0$
- 输入端: IN



02. 折叠变换 —— Biquad滤波器

考察节点1 ($S_1|3$) 输入:

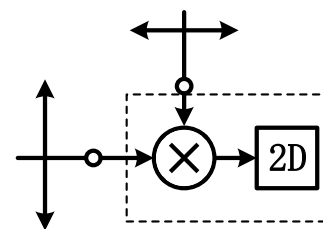
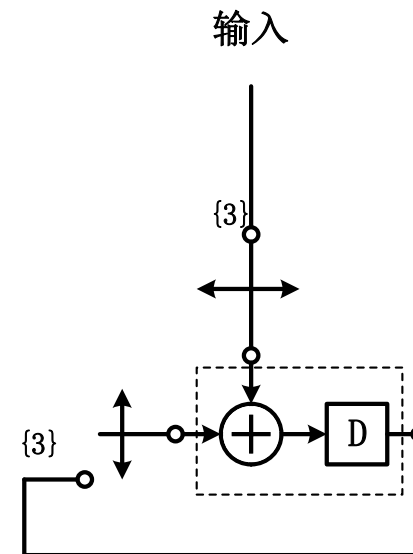
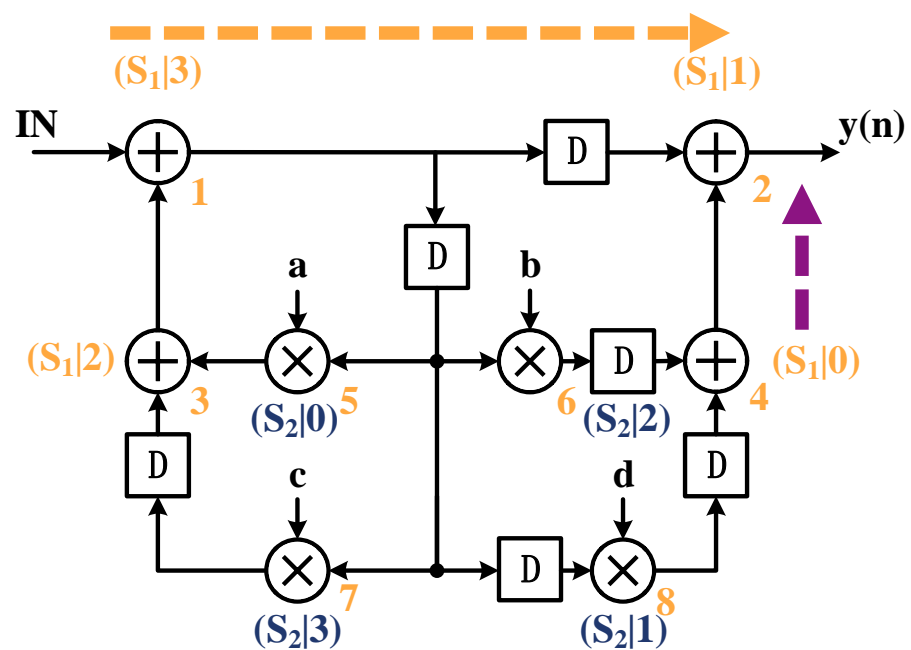
- 节点3: $D_F(3 \rightarrow 1) = 4(0) - 1 + 3 - 2 = 0$
- 输入端: IN



02. 折叠变换 —— Biquad滤波器

考察节点2 ($S_1|1$) 输入:

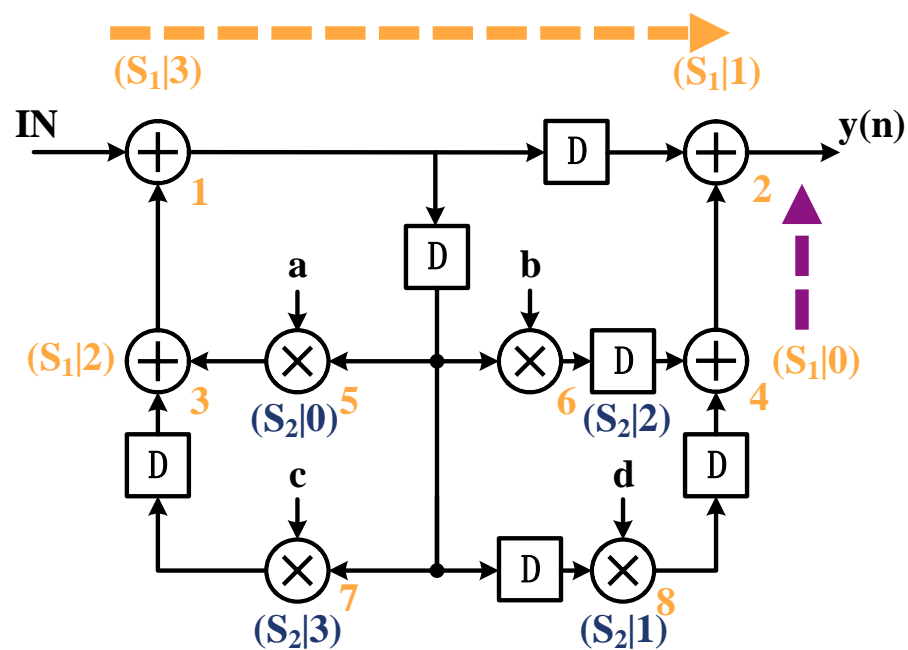
- 节点1: $D_F(1 \rightarrow 2) = 4(1) - 1 + 1 - 3 = 1$
- 节点4: $D_F(4 \rightarrow 2) = 4(0) - 1 + 1 - 0 = 0$



02. 折叠变换 —— Biquad滤波器

■ 考察节点2 ($S_1|1$) 输入:

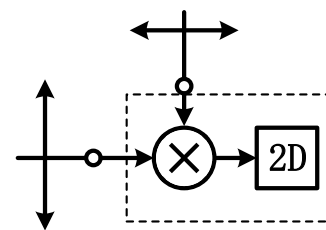
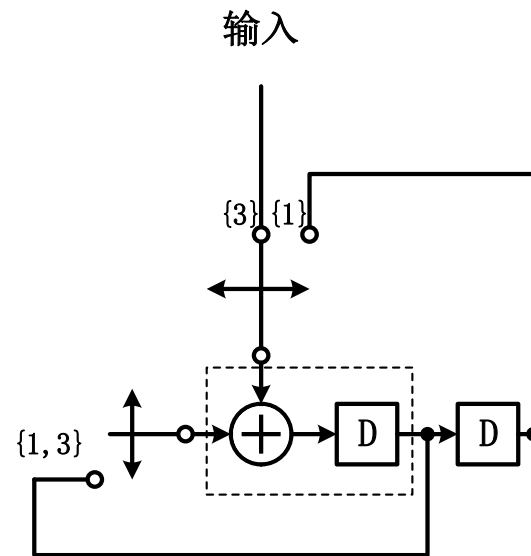
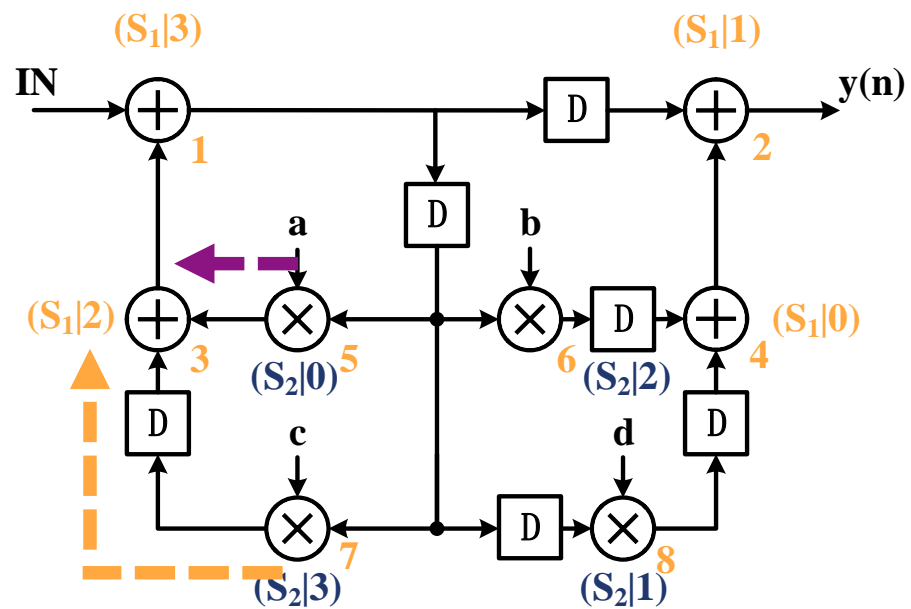
- 节点1: $D_F(1 \rightarrow 2) = 4(1) - 1 + 1 - 3 = 1$
- 节点4: $D_F(4 \rightarrow 2) = 4(0) - 1 + 1 - 0 = 0$



02. 折叠变换 —— Biquad滤波器

考察节点3 ($S_1|2$) 输入:

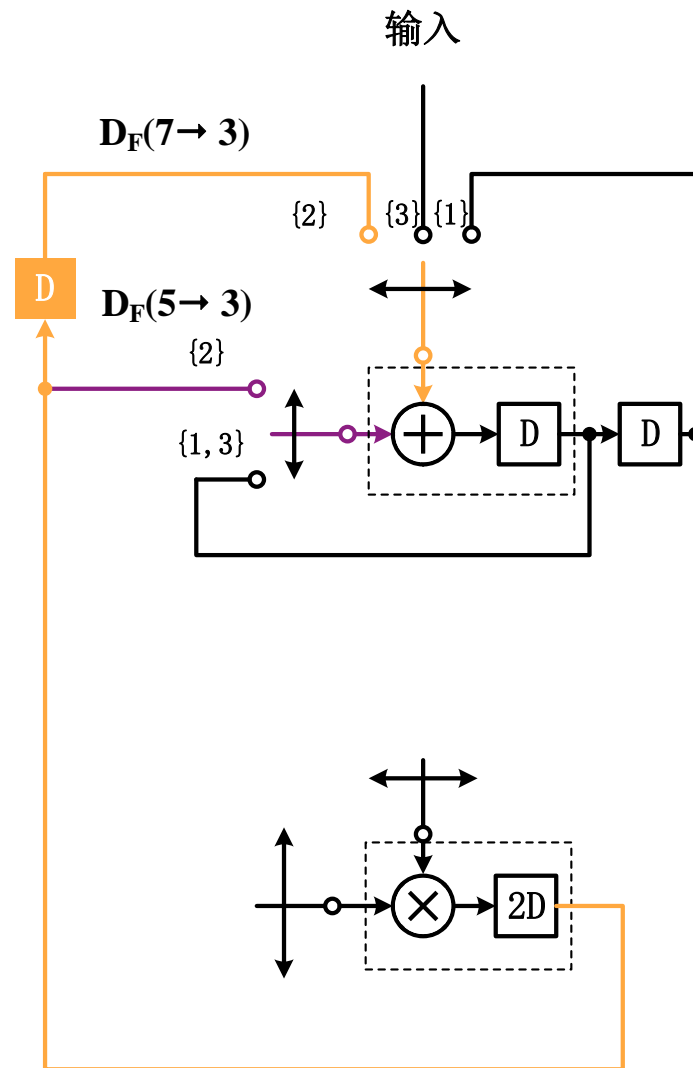
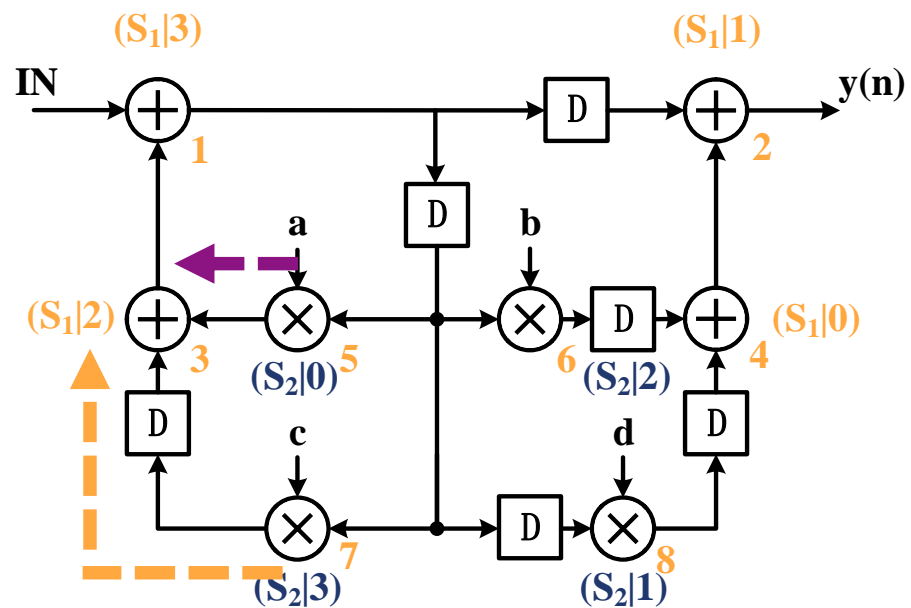
- 节点5: $D_F(5 \rightarrow 3) = 4(0) - 2 + 2 - 0 = 0$
- 节点7: $D_F(7 \rightarrow 3) = 4(1) - 2 + 2 - 3 = 1$



02. 折叠变换 —— Biquad滤波器

考察节点3 ($S_1|2$) 输入:

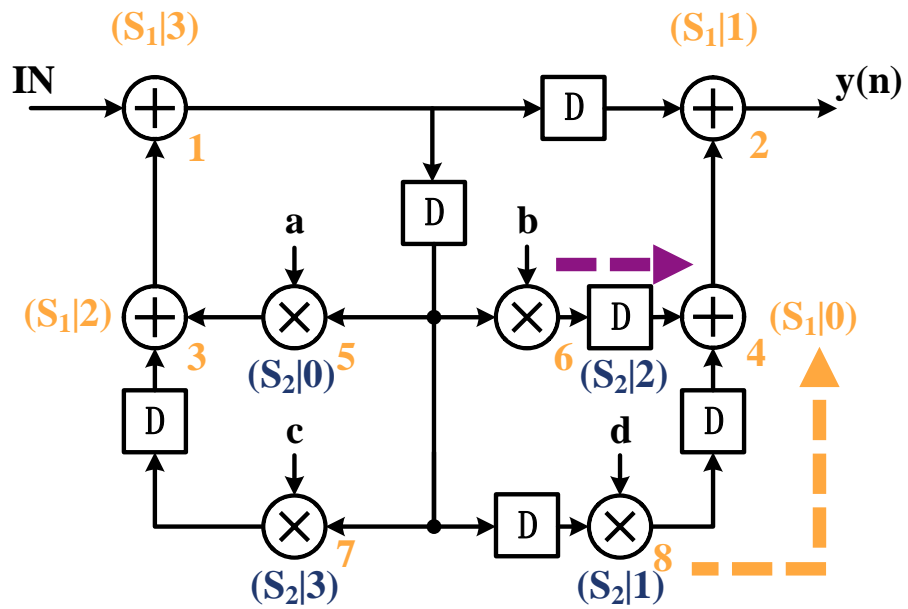
- 节点5: $D_F(5 \rightarrow 3) = 4(0) - 2 + 2 - 0 = 0$
- 节点7: $D_F(7 \rightarrow 3) = 4(1) - 2 + 2 - 3 = 1$



02. 折叠变换 —— Biquad 滤波器

考察节点4 ($S_1|0$) 输入:

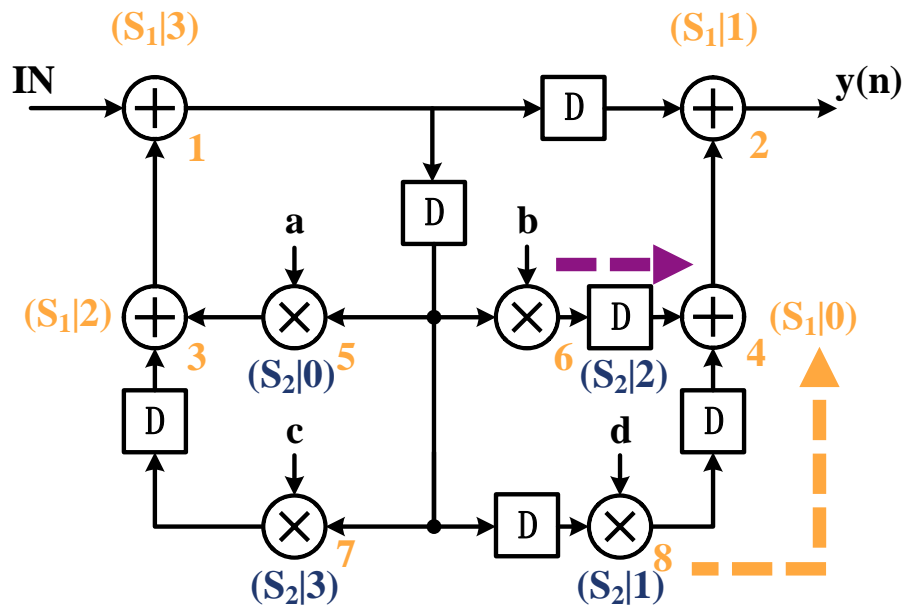
- **节点6: $D_F(6 \rightarrow 4) = 4(1) - 2 + 0 - 2 = 0$**
- **节点8: $D_F(8 \rightarrow 4) = 4(1) - 2 + 0 - 1 = 1$**



02. 折叠变换 —— Biquad 滤波器

考察节点4 ($S_1|0$) 输入:

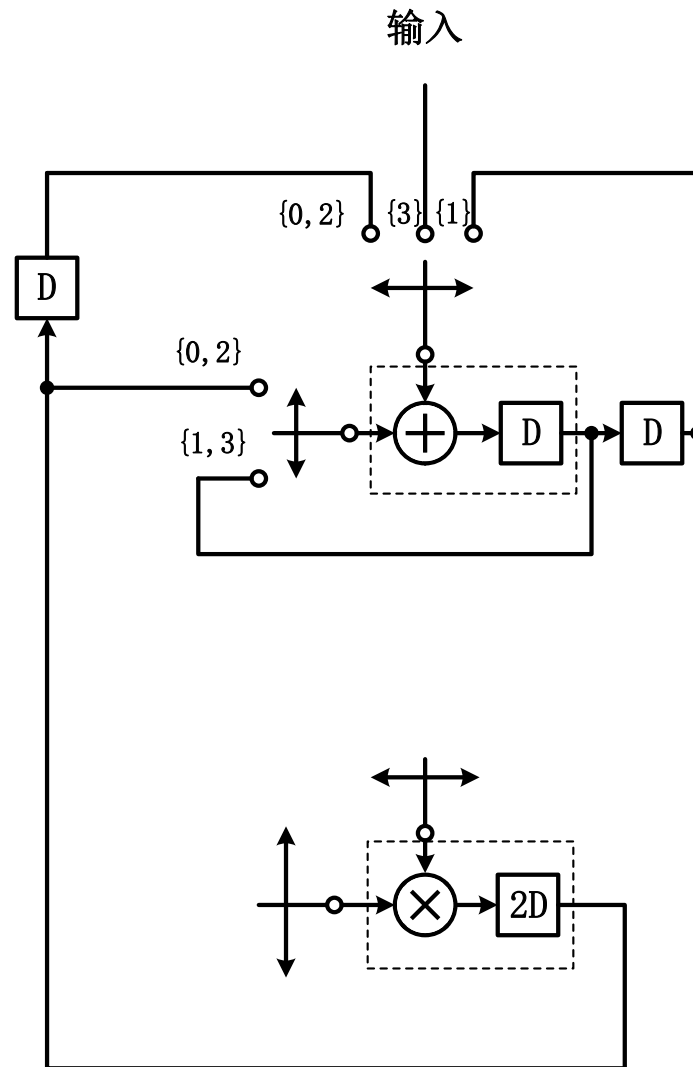
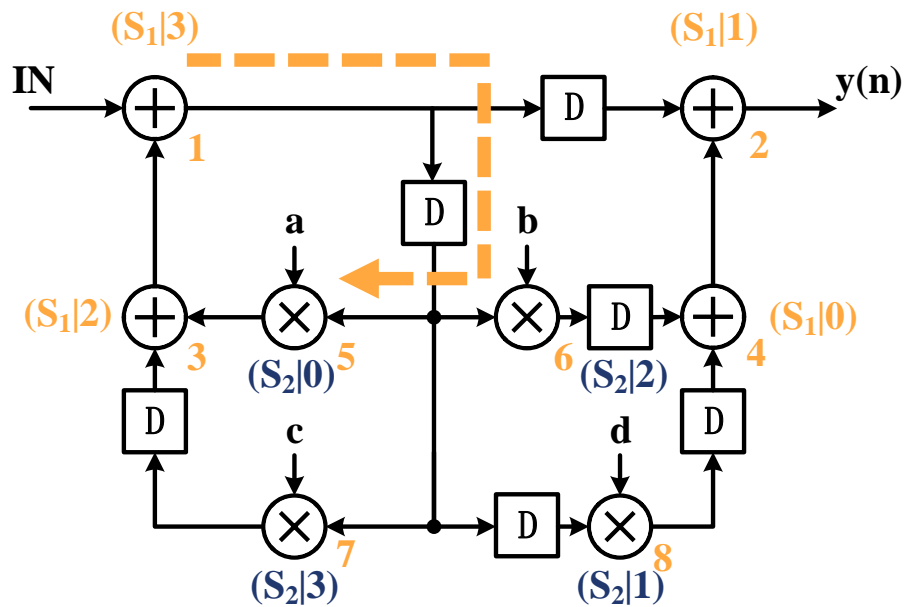
- **节点6: $D_F(6 \rightarrow 4) = 4(1) - 2 + 0 - 2 = 0$**
- **节点8: $D_F(8 \rightarrow 4) = 4(1) - 2 + 0 - 1 = 1$**



02. 折叠变换 —— Biquad滤波器

考察节点5 ($S_2|0$) 输入:

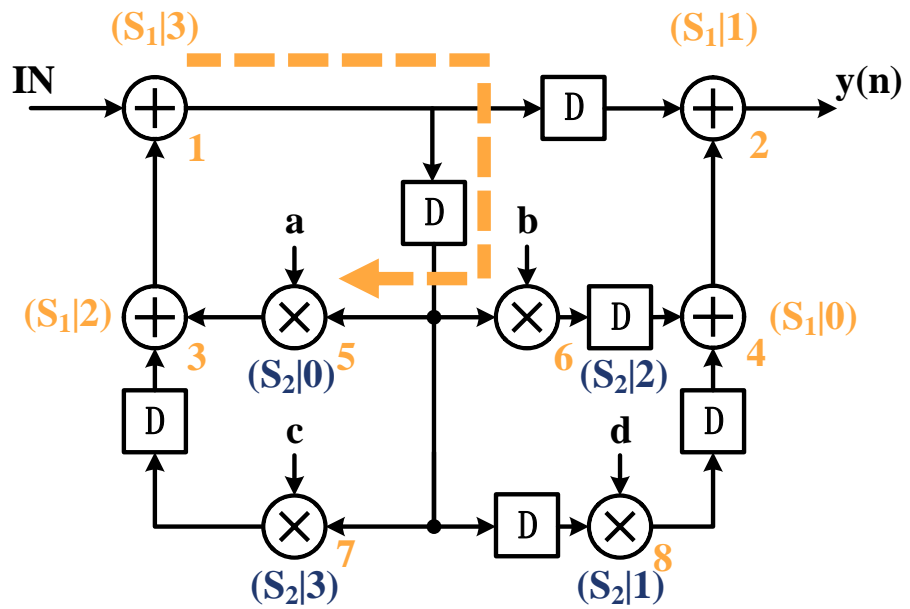
- **节点1: $D_F(1 \rightarrow 5) = 4(1) - 1 + 0 - 3 = 0$**
- **系数: a**



02. 折叠变换 —— Biquad 滤波器

考察节点5 ($S_2|0$) 输入:

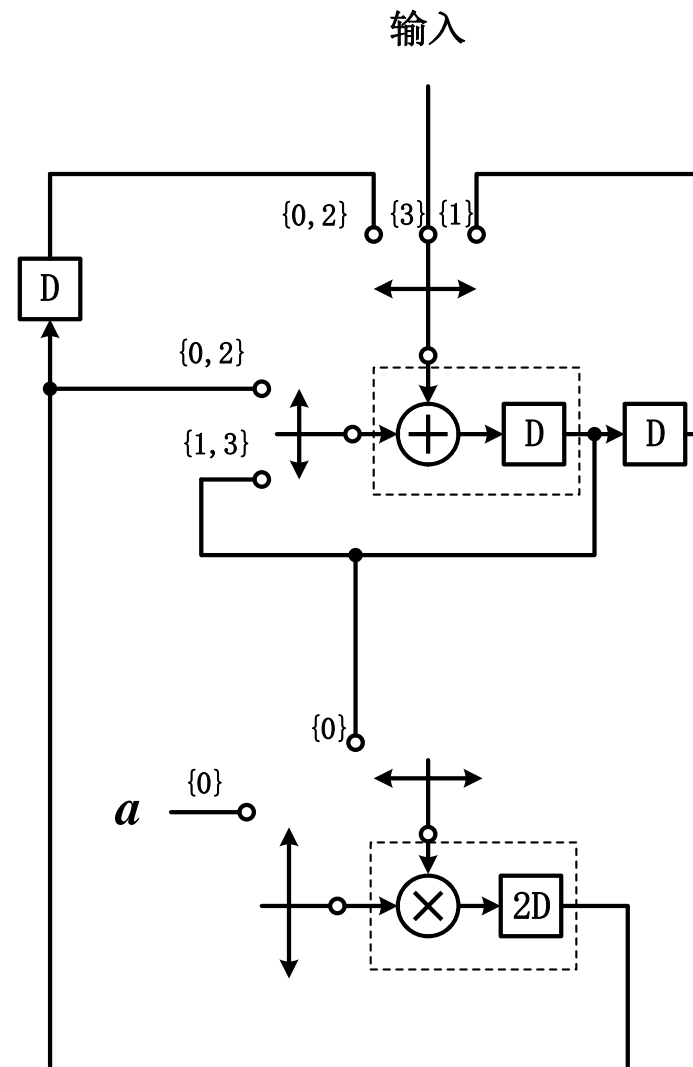
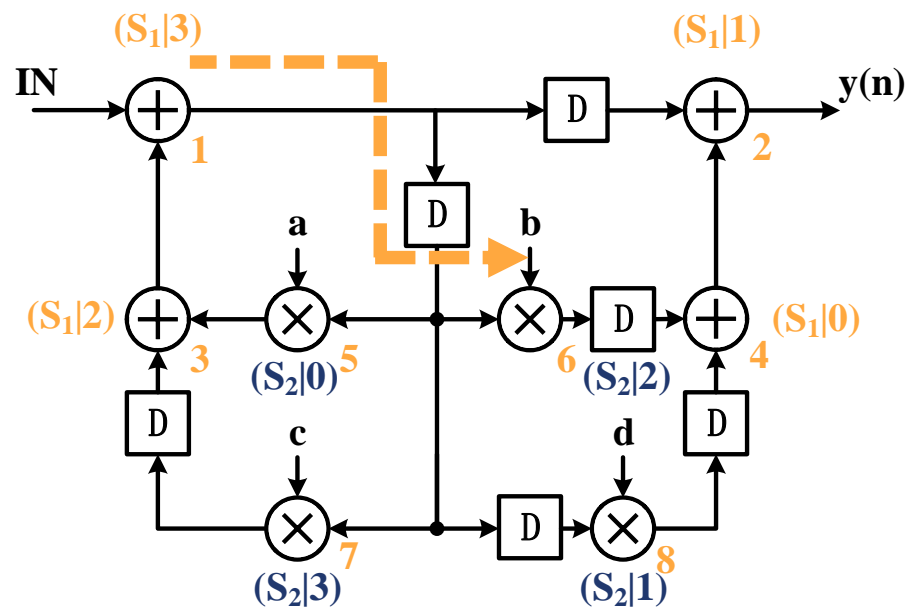
- **节点1: $D_F(1 \rightarrow 5) = 4(1) - 1 + 0 - 3 = 0$**
- **系数: a**



02. 折叠变换 —— Biquad滤波器

考察节点6 ($S_2|2$) 输入:

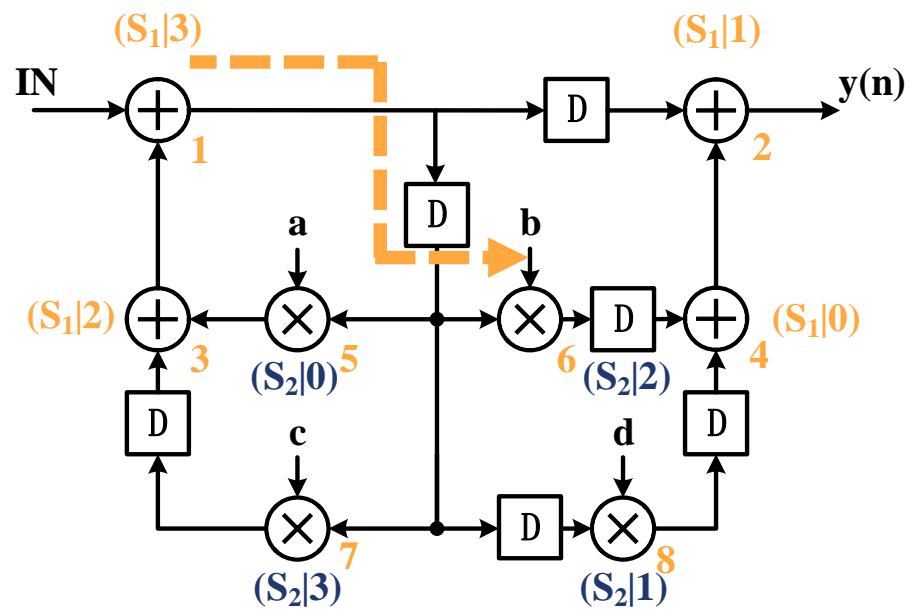
- 节点1: $D_F(1 \rightarrow 6) = 4(1) - 1 + 2 - 3 = 2$
- 系数: b



02. 折叠变换 —— Biquad滤波器

考察节点6 ($S_2|2$) 输入:

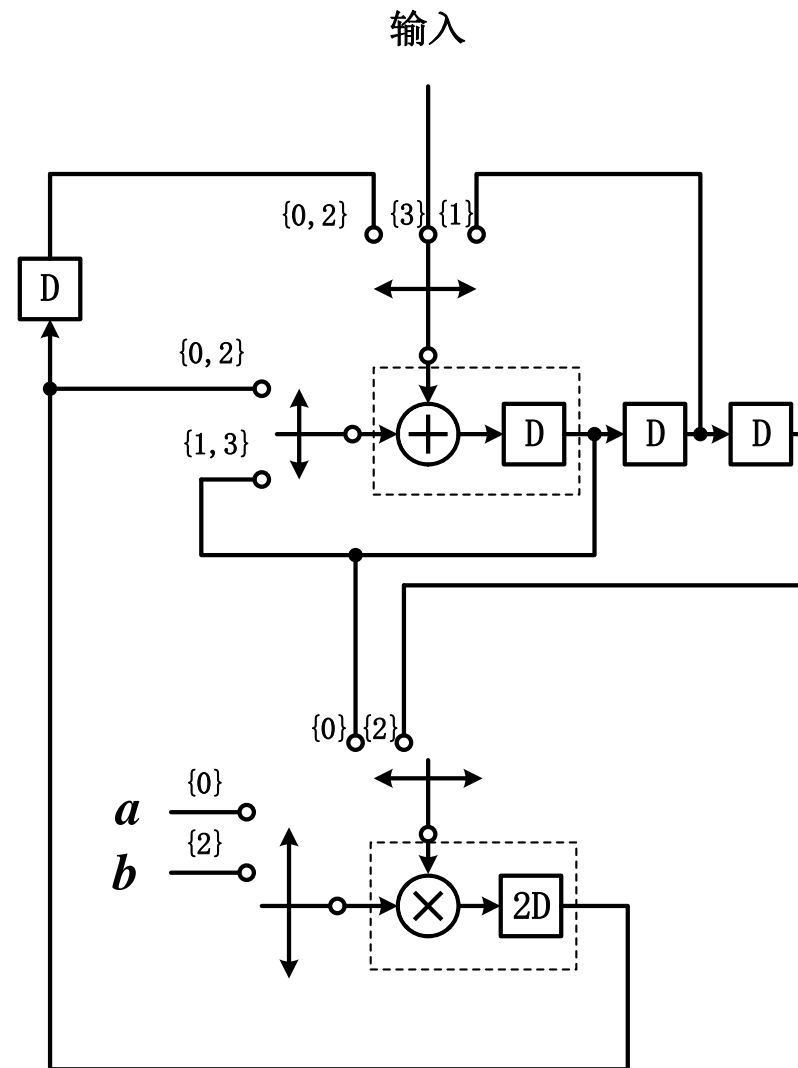
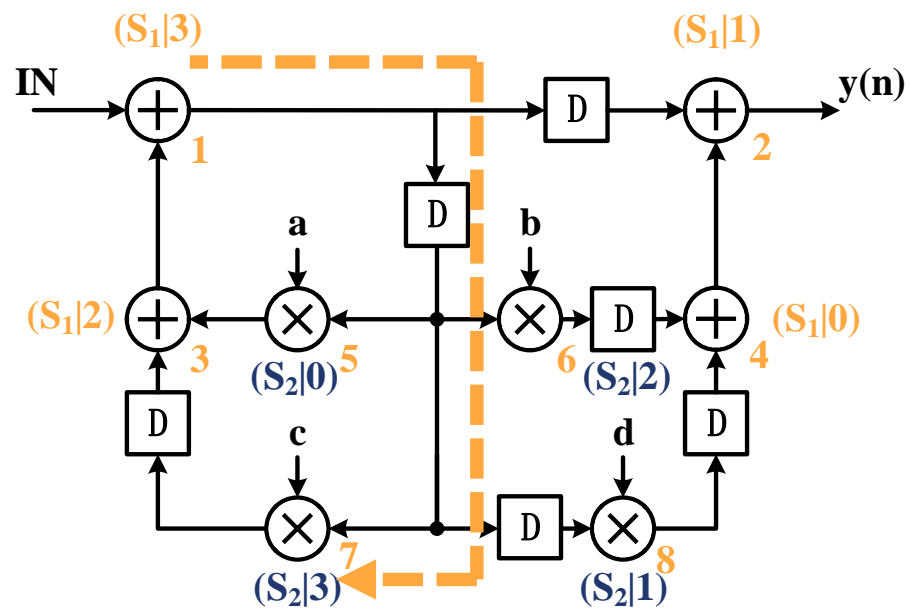
- 节点1: $D_F(1 \rightarrow 6) = 4(1) - 1 + 2 - 3 = 2$
- 系数: b



02. 折叠变换 —— Biquad滤波器

考察节点7 ($S_2|3$) 输入:

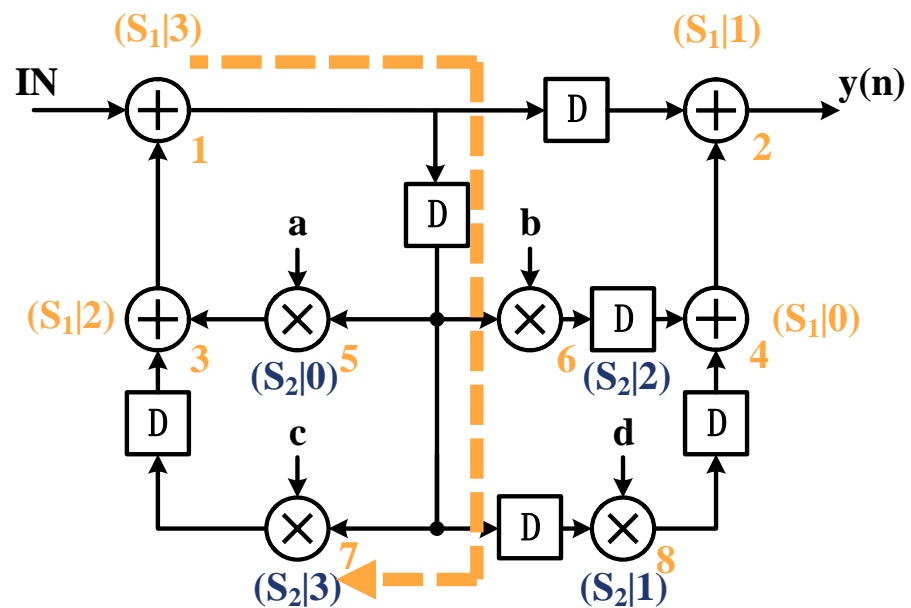
- 节点1: $D_F(1 \rightarrow 7) = 4(1) - 1 + 3 - 3 = 3$
- 系数: c



02. 折叠变换 —— Biquad滤波器

考察节点7 ($S_2|3$) 输入:

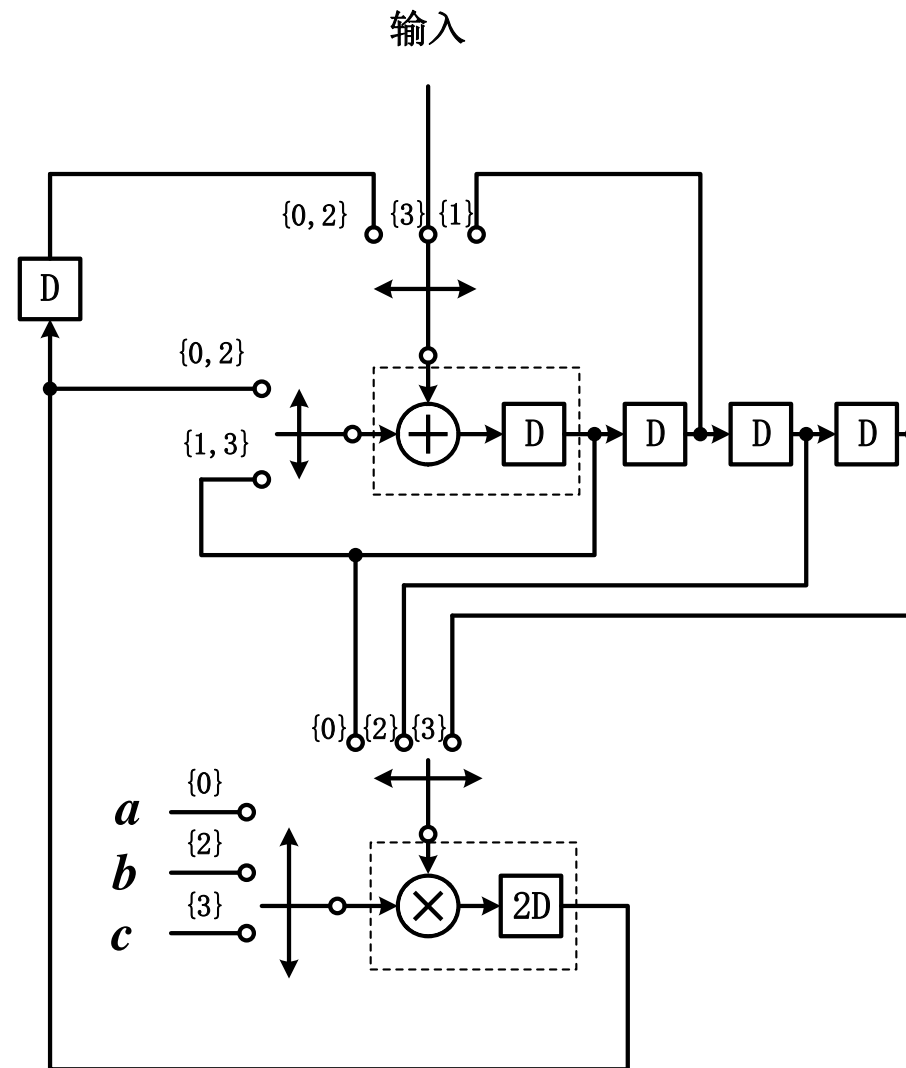
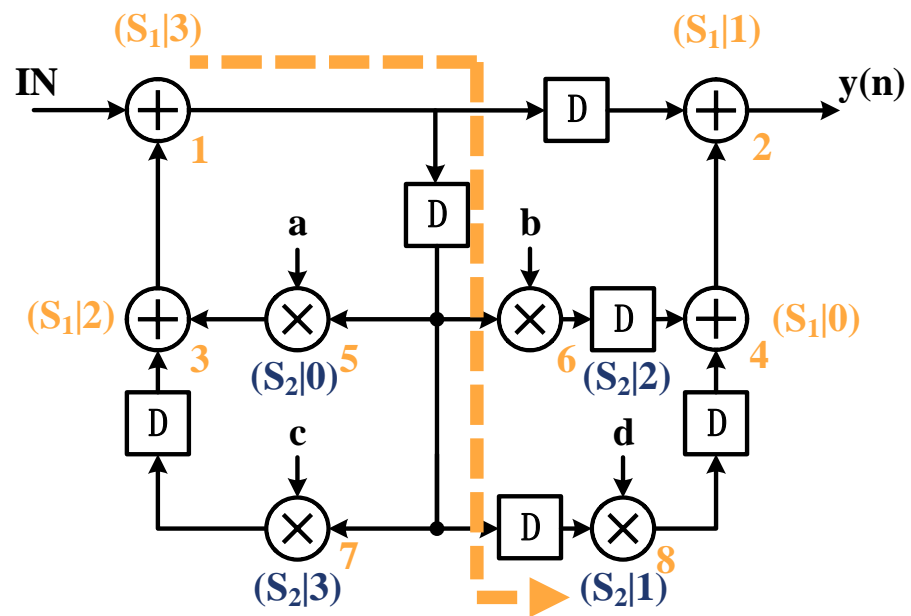
- 节点1: $D_F(1 \rightarrow 7) = 4(1) - 1 + 3 - 3 = 3$
- 系数: c



02. 折叠变换 —— Biquad滤波器

考察节点8 ($S_2|1$) 输入:

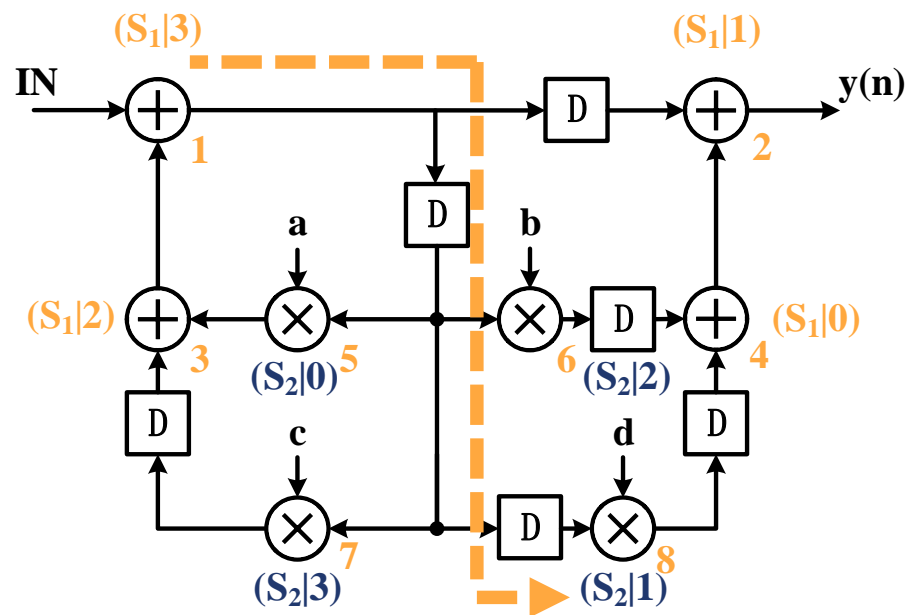
- 节点1: $D_F(1 \rightarrow 8) = 4(2) - 1 + 1 - 3 = 5$
- 系数: d



02. 折叠变换 —— Biquad滤波器

考察节点8 ($S_2|1$) 输入:

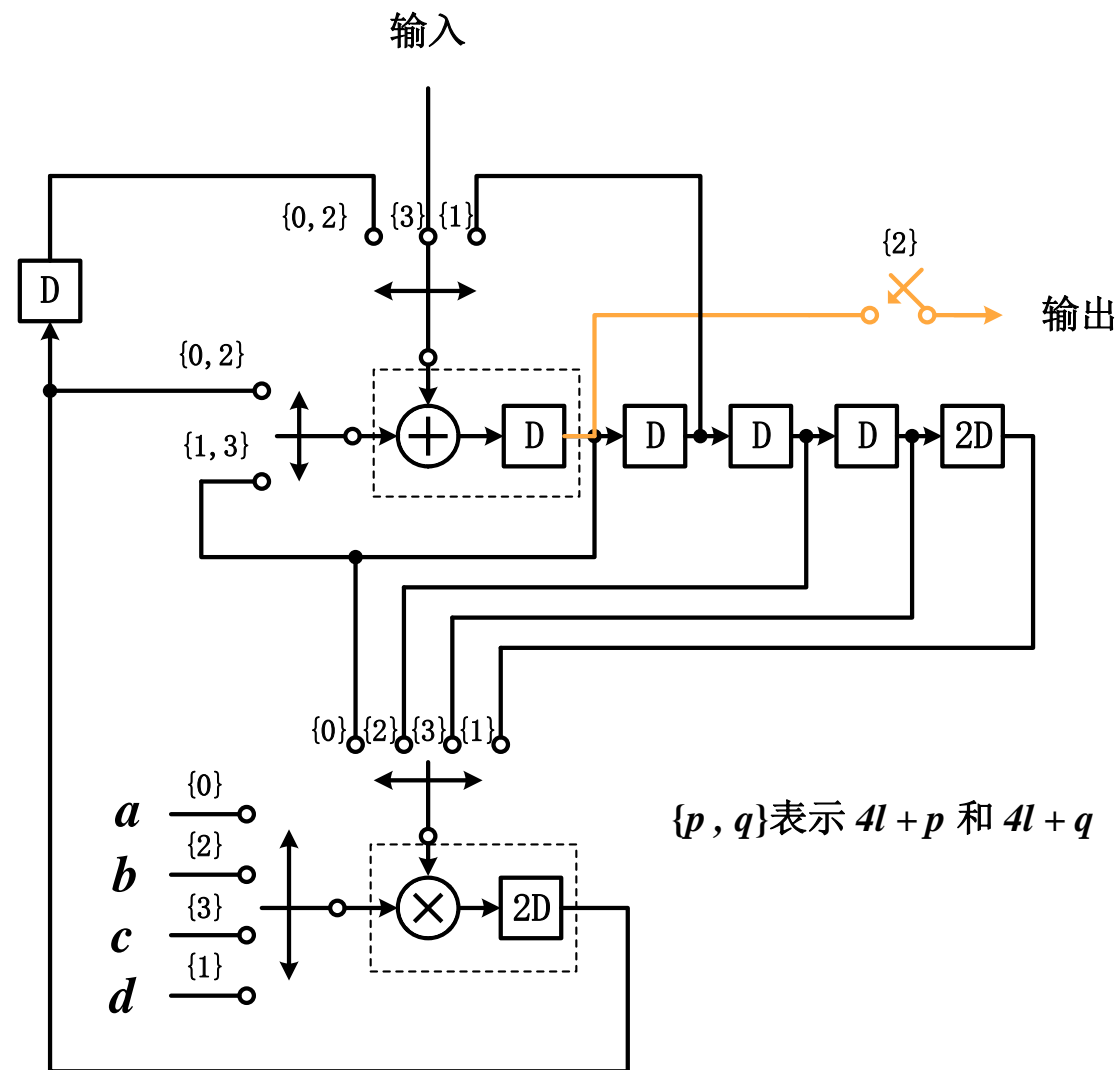
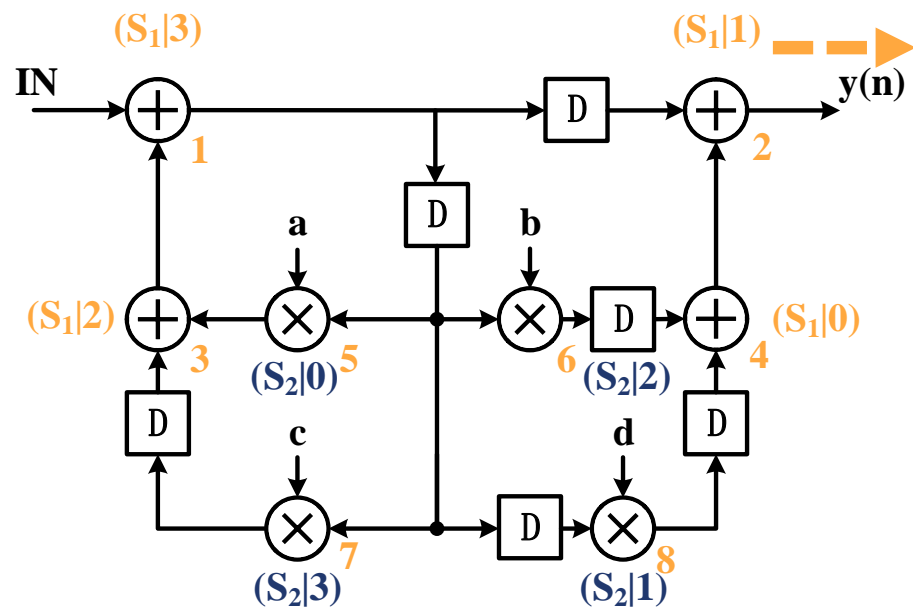
- 节点1: $D_F(1 \rightarrow 8) = 4(2) - 1 + 1 - 3 = 5$
- 系数: d



02. 折叠变换 —— Biquad滤波器

考察输出端的输入:

- 节点2 ($S_1|1$) 直接输入到输出端
- 因为内部有一级流水线, 所以输出的序号为2





目录

01 折叠的基本概念

02 折叠变换

03 寄存器最小化技术

04 折叠架构的寄存器最小化

05 本章总结



03. 寄存器最小化技术——寿命分析

定义：

折叠会插入寄存器，寿命分析是计算用硬件实现DSP算法所需的最少寄存器数的过程

一个样值（也称为变量）从产生的时刻到被应用的时刻是激活的。变量被应用后，就消亡了。

在寿命分析中，算出每个单位时间里的激活变量数，就可确定任意单位时间里的最大激活变量数，这即是实现DSP程序所需的最小寄存器数

03. 寄存器最小化技术——线性寿命图

用于图形化表示线性模式下的每个变量的寿命

示例：变量a在 $n \in \{1, 2, 3, 4\}$ 单位时间里激活

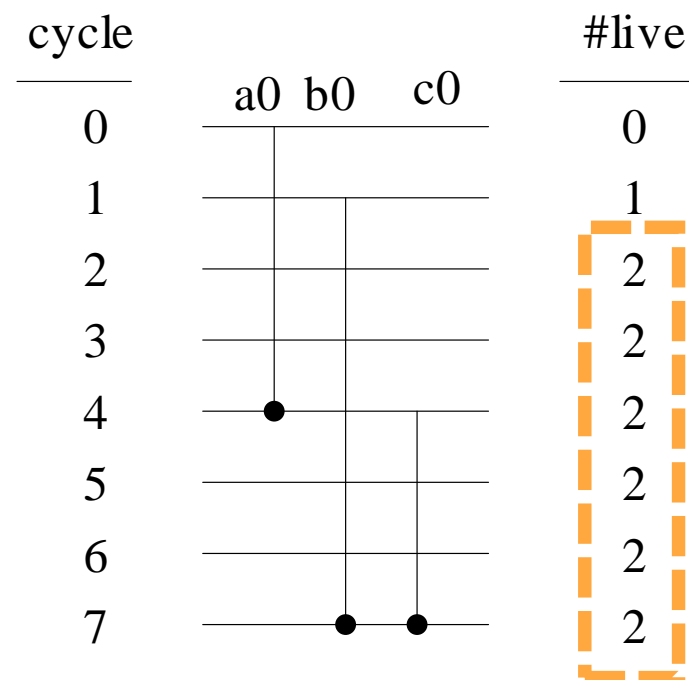
变量b在 $n \in \{2, 3, 4, 5, 6, 7\}$ 单位时间里激活

变量c在 $n \in \{5, 6, 7\}$ 单位时间里激活

□ 假设之前和之后的程序迭代的变量寿命不和变量a, b, c的寿命交叉

请注意

变量在它所产生的时钟周期是非激活的
在它被应用的时钟周期里是激活的

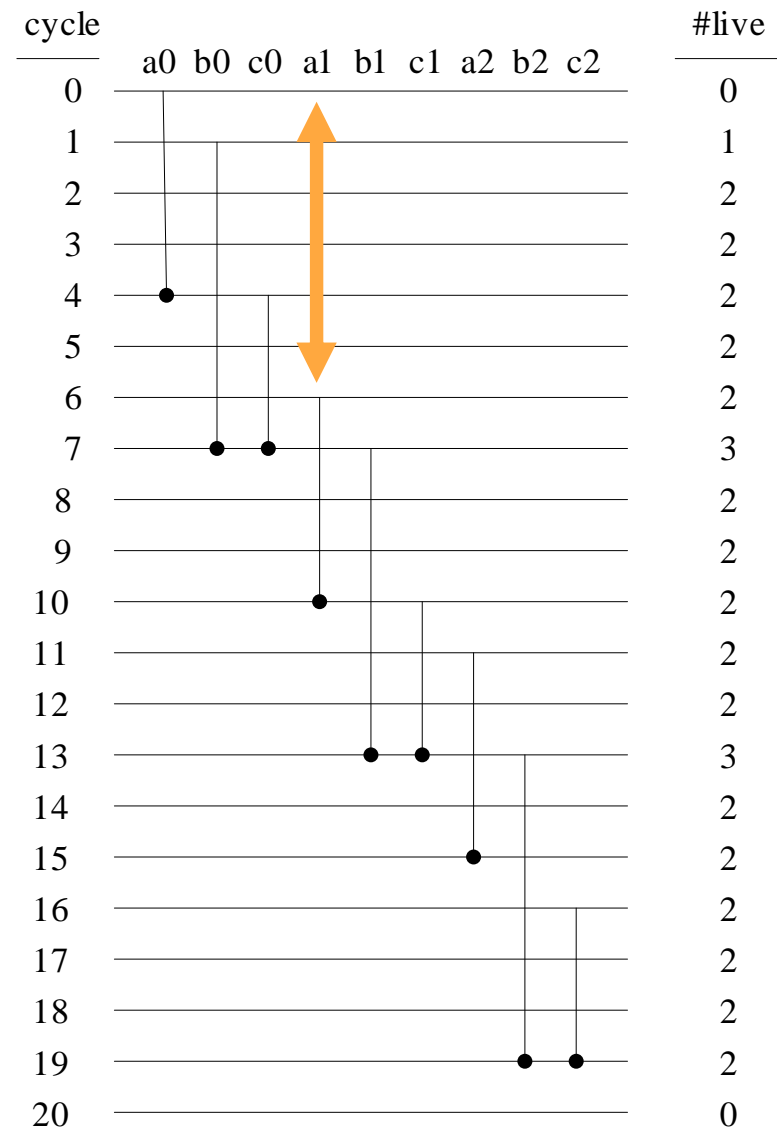
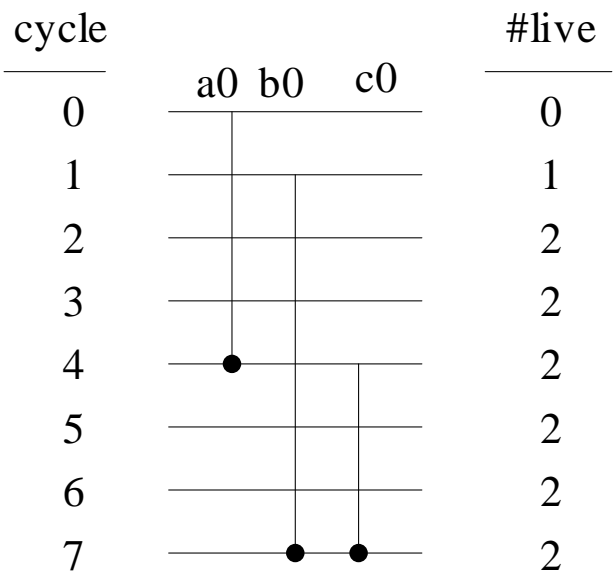


最小寄存器数=2

03. 寄存器最小化技术——线性寿命图

DSP程序通常为周期性的

- 例如：迭代周期 $N = 6$



N=6

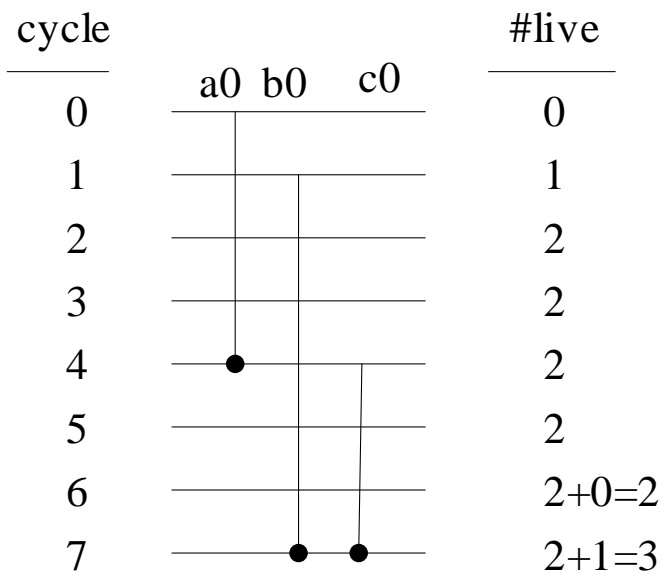
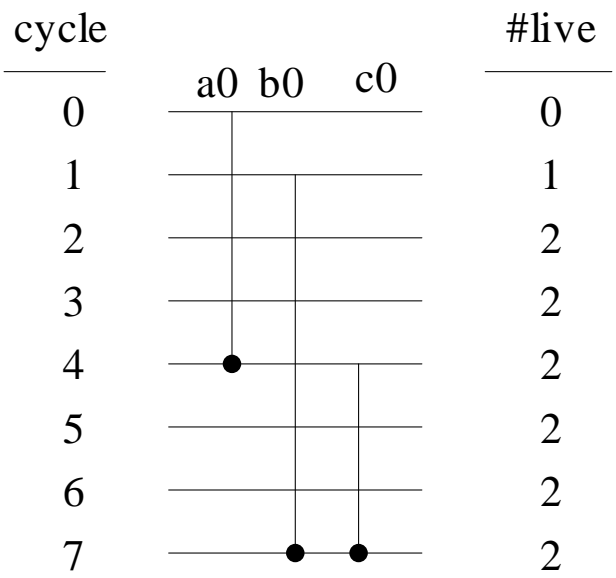
最大激活寄存器数 ω

03. 寄存器最小化技术——线性寿命图

DSP程序通常为周期性的

- 例如：迭代周期 $N = 6$

只考虑首次迭代



最大激活寄存器数 ∞

03. 寄存器最小化技术——寿命表

3X3矩阵转置:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$



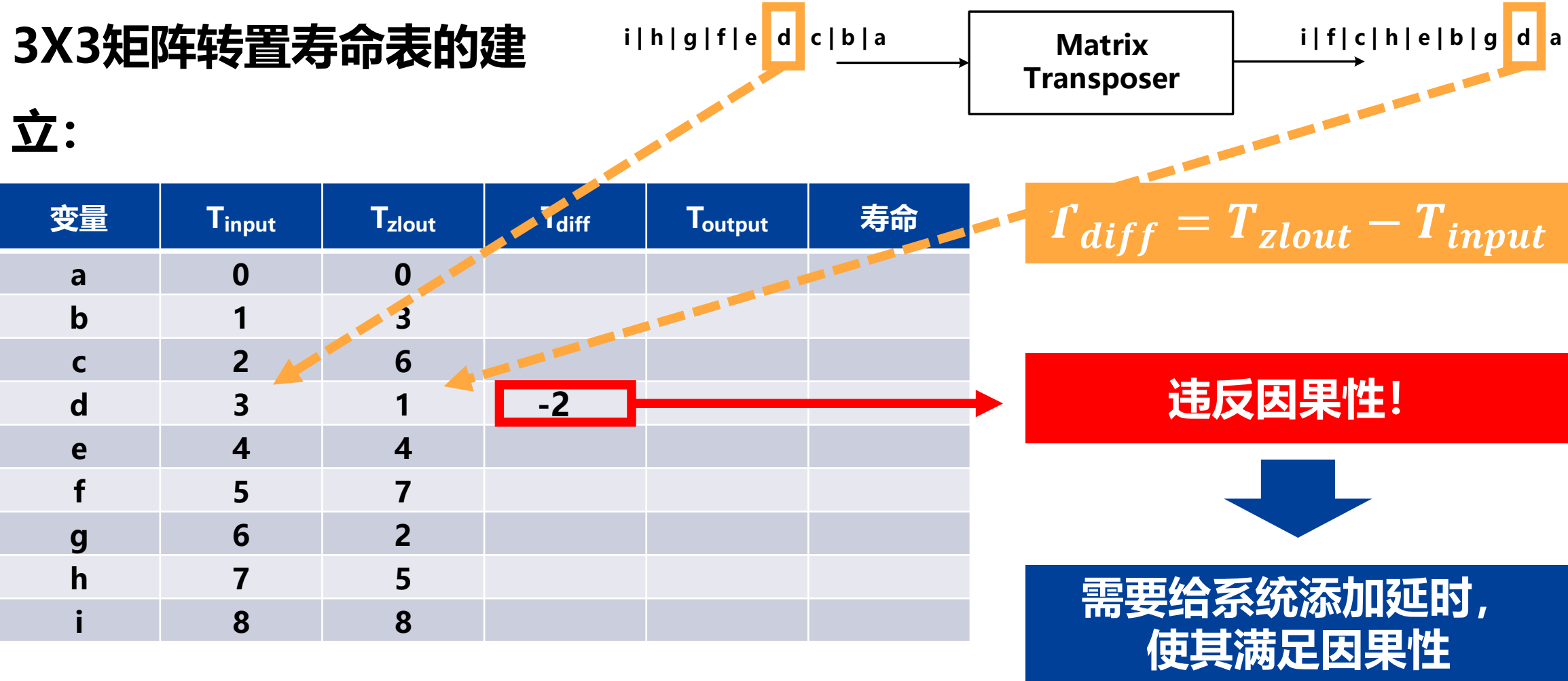
$$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$



一次迭代 $T_{\text{iter}} = 9$ 时钟周期

03. 寄存器最小化技术——寿命表

3X3矩阵转置寿命表的建立:



□ T_{input} : 输入时间; T_{zlout} : 0延时输出时间;

03. 寄存器最小化技术——寿命表

3X3矩阵转置寿命表的建立:



变量	T _{input}	T _{zlout}	T _{diff}	T _{output}	寿命
a	0	0	0		
b	1	3	2		
c	2	6	4		
d	3	1	-2		
e	4	4	0		
f	5	7	2		
g	6	2	-4		
h	7	5	-2		
i	8	8	0		

添加延时

$T_{lat}=T_{diff}$ 最大负值的幅度

$T_{lat} = |T_{negative\ diff\ max}| = 4$



$T_{output} = T_{zlout} + T_{lat}$

□ T_{input}: 输入时间; T_{zlout}: 0延时输出时间; T_{output}: 输出时间

03. 寄存器最小化技术——寿命表

3X3矩阵转置寿命表的建立:



变量	T _{input}	T _{zlout}	T _{diff}	T _{output}	寿命
a	0	0	0	4	
b	1	3	2	7	
c	2	6	4	10	
d	3	1	-2	5	
e	4	4	0	8	
f	5	7	2	11	
g	6	2	-4	6	
h	7	5	-2	9	
i	8	8	0	12	

添加延时

$T_{lat}=T_{diff}$ 最大负值的幅度

$T_{lat} = |T_{negative\ diff\ max}| = 4$



$T_{output} = T_{zlout} + T_{lat}$

□ T_{input}: 输入时间; T_{zlout}: 0延时输出时间; T_{output}: 输出时间

03. 寄存器最小化技术——寿命表

3X3矩阵转置寿命表的建立:



变量	T _{input}	T _{zlout}	T _{diff}	T _{output}	寿命
a	0	0	0	4	0→4
b	1	3	2	7	1→7
c	2	6	4	10	2→10
d	3	1	-2	5	3→5
e	4	4	0	8	4→8
f	5	7	2	11	5→11
g	6	2	-4	6	6→6
h	7	5	-2	9	7→9
i	8	8	0	12	8→12

添加延时

$T_{lat}=T_{diff}$ 最大负值的幅度

$T_{lat} = |T_{negative\ diff\ max}| = 4$

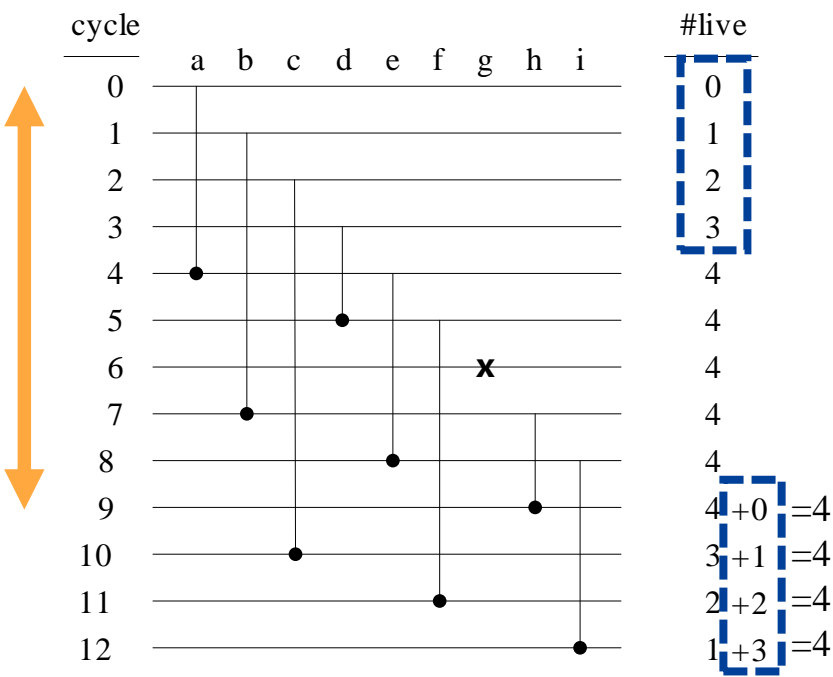


$T_{output} = T_{zlout} + T_{lat}$

□ T_{input}: 输入时间; T_{zlout}: 0延时输出时间; T_{output}: 输出时间

03. 寄存器最小化技术——线性寿命图

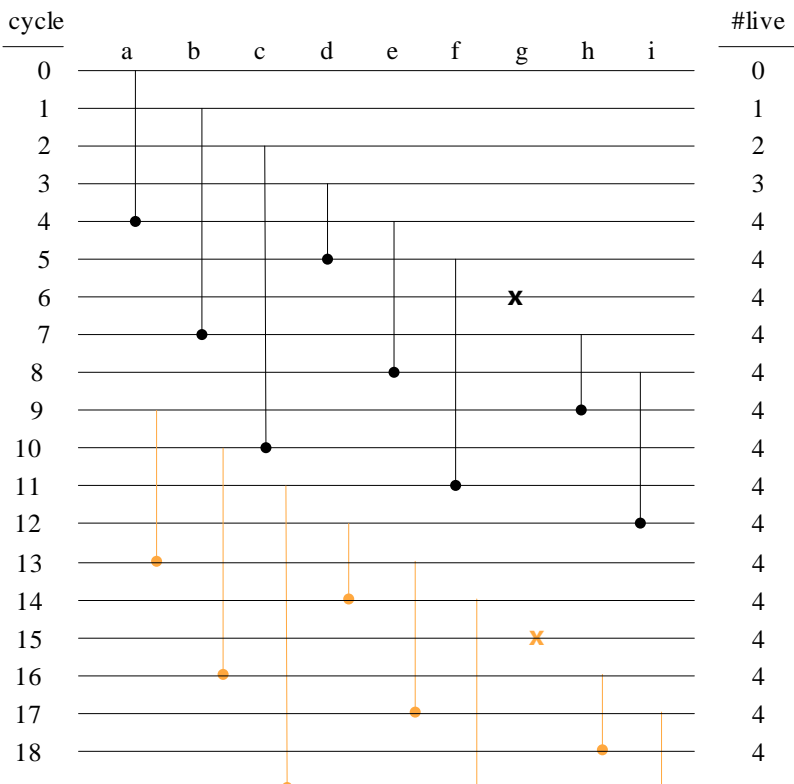
3X3矩阵转置寿命图的建立:



迭代周期N=9



=



最大激活寄存器数=4

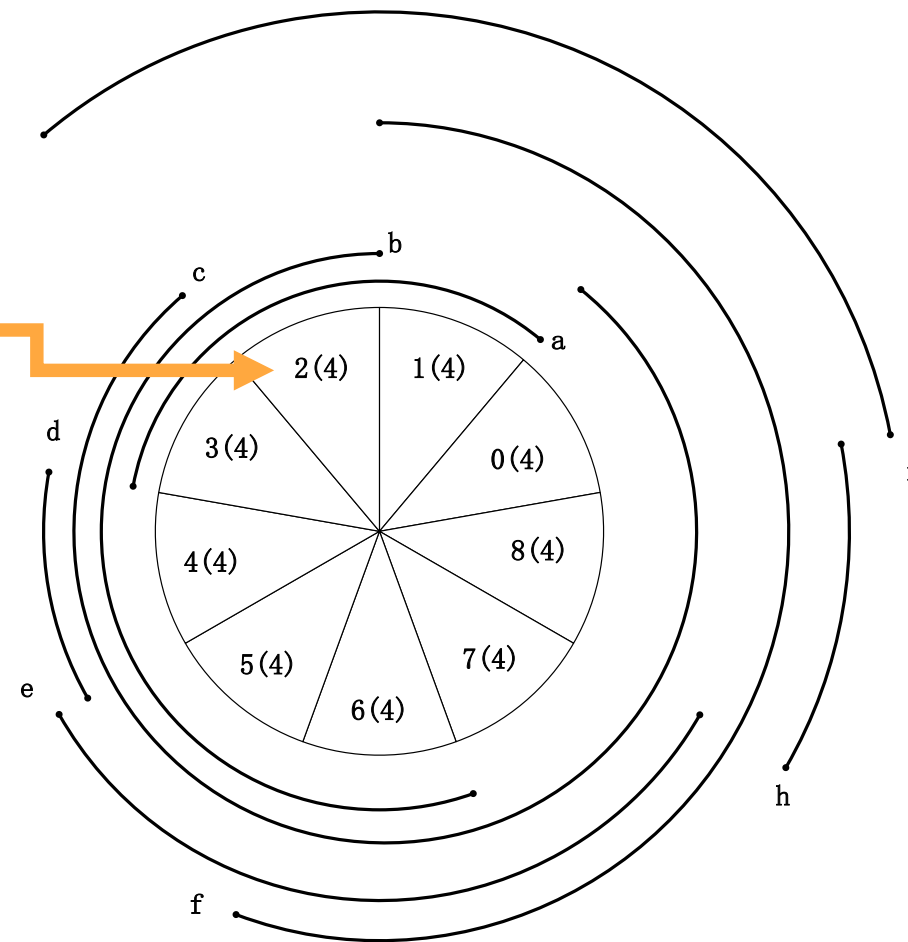
03. 寄存器最小化技术——循环寿命图

有利于表示DSP程序的周期行为

- i 表示时间块 i 和所有 $\{Nl+i\}$ 时刻
 - $i=2 \Rightarrow \{2, 11, 20, 29, \dots\}$ 时刻
- 括号内数字表示每个时间块激活变量数

请注意

在单位时间 j 产生并与单位时间 k 被应用的变量，在单位时间 $j+1$ 到 k 是激活的，因为变量在它产生的单位时间里是不需要存储的



3X3矩阵转置循环寿命图

03. 寄存器最小化技术——前向-后向分配技术

■ 将数据分配给最小寄存器数的分配方案

■ 分配步骤：

步骤
1

通过寿命分析确定最小寄存器数

03. 寄存器最小化技术——前向-后向分配技术

■ 将数据分配给最小寄存器数的分配方案

■ 分配步骤：

步骤
2

- ◆ 在寿命期开始所对应的时间输入每个变量；
- ◆ 如果给定周期里有多多个变量输入，这些变量就分配给多个寄存器，使得寿命最长的变量分配给初始寄存器，其他变量按寿命递减排序逐个分配给寄存器。

03. 寄存器最小化技术——前向-后向分配技术

■ 将数据分配给最小寄存器数的分配方案

■ 分配步骤：

步骤
3

- ◆ 每个变量都以这种前向方式分配，直至变量消亡或者到达末尾寄存器；
- ◆ 在前向分配中，如果寄存器 i 在当前周期里存有某变量，则寄存器 $i+1$ 在下一个周期存有相同的变量。

03. 寄存器最小化技术——前向-后向分配技术

■ 将数据分配给最小寄存器数的分配方案

■ 分配步骤：

步骤
4

- ◆ 因为分配是周期性的，当前迭代的分配在接下来的迭代中还会重复；
- ◆ 因此，如果 R_j 在周期 l 被某个变量占据，则 R_j 会在周期 $l+N$ 占有相同的变量， N 表示分配周期。因此，对任意 j 和 l ， R_j 在单位时间 $l+N$ 是发生重叠。

03. 寄存器最小化技术——前向-后向分配技术

将数据分配给最小寄存器数的分配方案

分配步骤：

步骤 5

- ◆ 对到达末尾寄存器但仍没有消亡的变量，可以计算出它的剩余寿命，然后基于先到先得的原则，将这些变量以后向的方式分配给寄存器；
- ◆ 如果后向分配中有多个寄存器可用，首先尝试选择在末尾寄存器和该寄存器之间已进行过后向分配的寄存器；
- ◆ 如果有不止一个寄存器符合后向分配，在候选寄存器中，选择有最少但足够多的前向寄存器来完成变量分配的那个寄存器；
- ◆ 一个变量经过后向分配完成后，再进行前向分配，直至消亡或再次到达末尾寄存器。

03. 寄存器最小化技术——前向-后向分配技术

示例1: 3X3矩阵转置

变量	寿命
a	0→4
b	1→7
c	2→10
d	3→5
e	4→8
f	5→11
g	6→6
h	7→9
i	8→12

cycle	Input	R1	R2	R3	R4	Output
0	a					
1	b	a				
2	c	b	a			
3	d	c	b	a		
4	e	d	c	b	<u>a</u>	A
5	f	e	<u>d</u>	c	b	D
6	<u>g</u>	f	e	b	c	G
7	h	c	f	e	<u>b</u>	B
8	i	h	c	f	<u>e</u>	E
9		i	<u>h</u>	c	f	H
10			i	f	<u>c</u>	C
11				i	<u>f</u>	F
12					<u>i</u>	I

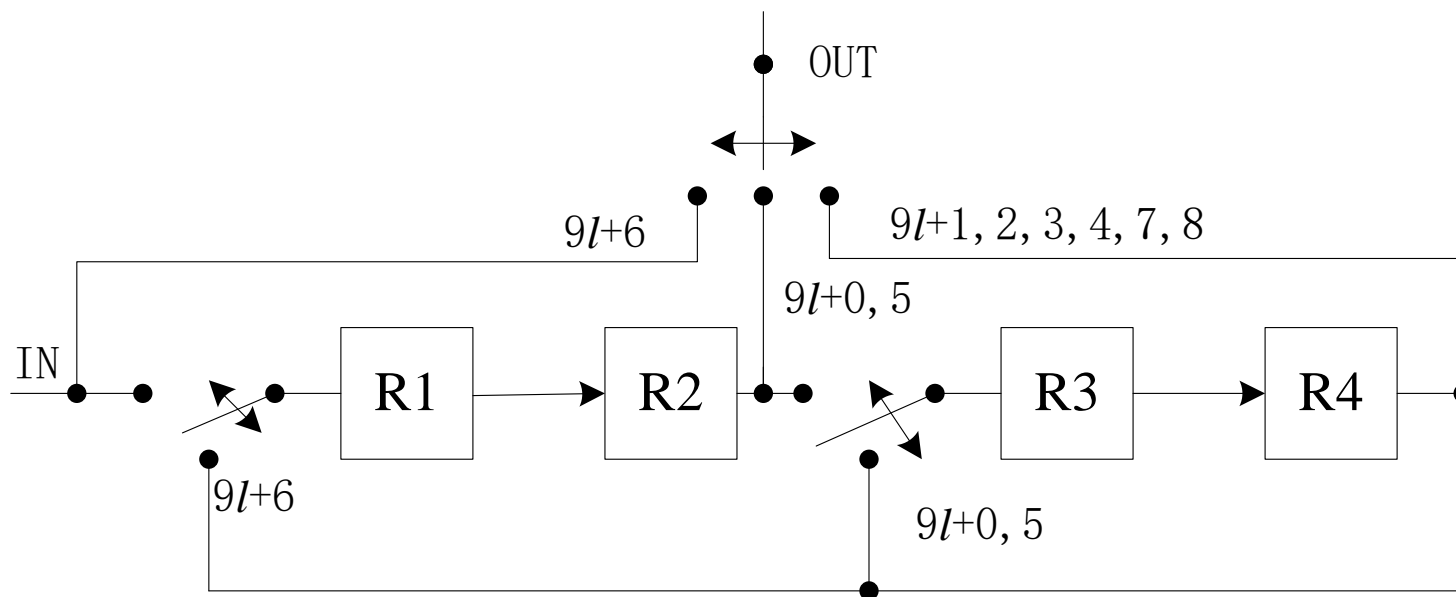
寄存器分配表

03. 寄存器最小化技术——前向-后向分配技术

示例1：

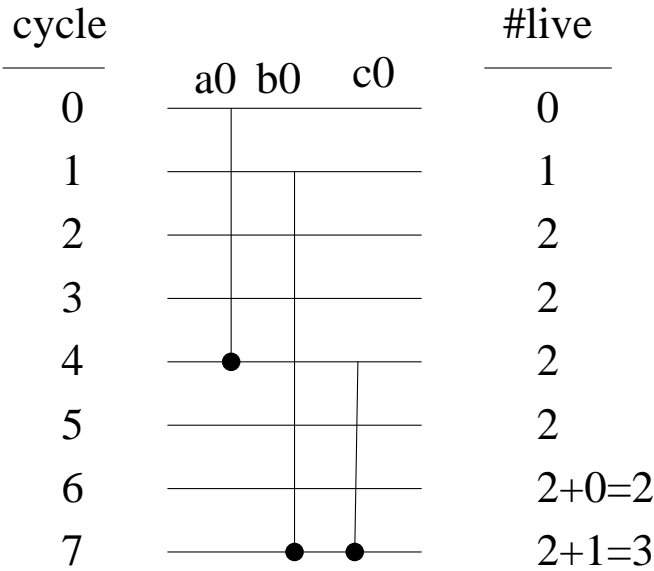
3X3矩阵转置折叠后的架

构：



03. 寄存器最小化技术——前向-后向分配技术

■ 示例2:



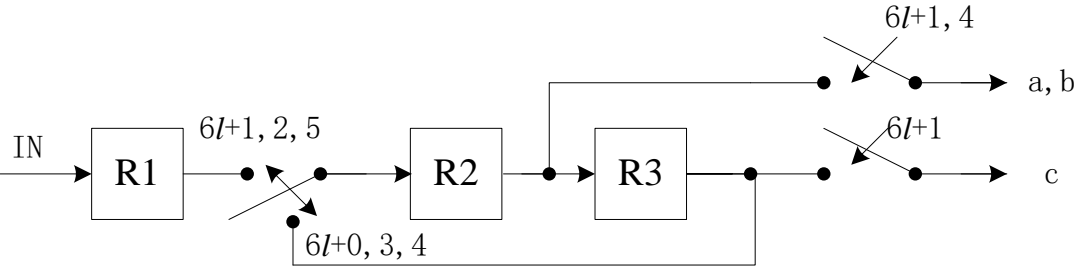
线性寿命图

cycle	input	R1	R2	R3	output
0	a				
1	b	a			
2		b	a		
3			b	a	
4	c		<u>a</u>	b	a
5		c	b		
6			c	b	
7			<u>b</u>	<u>c</u>	b,c

寄存器分配表

03. 寄存器最小化技术——前向-后向分配技术

■ 示例2:



折叠后的架构

cycle	input	R1	R2	R3	output
0	a				
1	b	a			
2		b	a		
3			b	a	
4	c		<u>a</u>	b	a
5		c	b		
6			c	b	
7			<u>b</u>	<u>c</u>	b,c

寄存器分配表



目录

01 折叠的基本概念

02 折叠变换

03 寄存器最小化技术

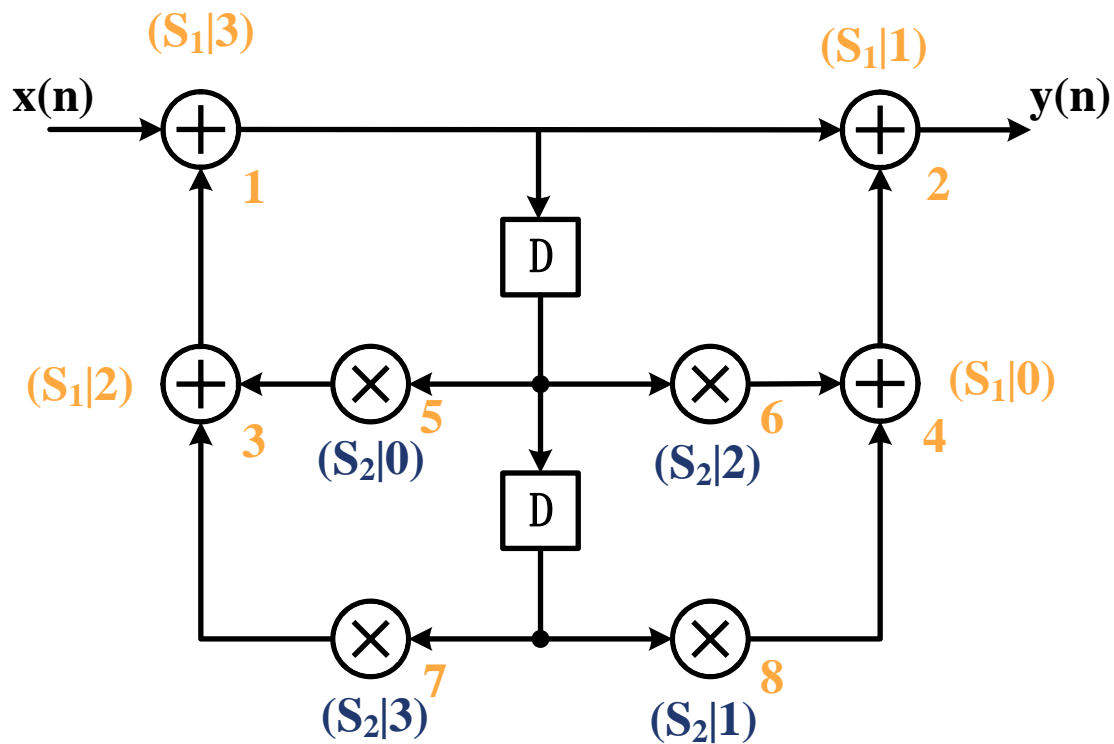
04 折叠架构的寄存器最小化

05 本章总结



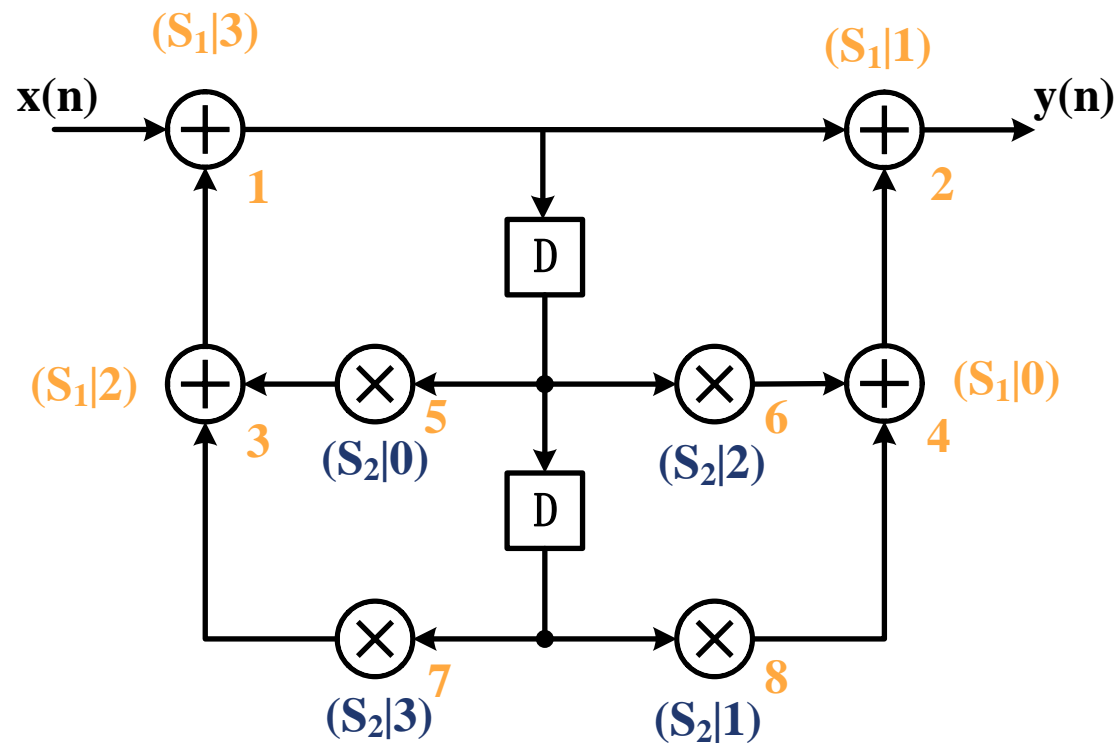
04. 折叠架构的寄存器最小化——Biquad滤波器

对最原始的DFG折叠会怎样？



04. 折叠架构的寄存器最小化——Biquad滤波器

对最原始的DFG折叠会怎样？



$$D_F(U \rightarrow V) = Nw(e) - P_u + v - u$$

$$D_F(1 \rightarrow 2) = -3$$

$$D_F(1 \rightarrow 5) = 0$$

$$D_F(1 \rightarrow 6) = 2$$

$$D_F(1 \rightarrow 7) = 7$$

$$D_F(1 \rightarrow 8) = 5$$

$$D_F(3 \rightarrow 1) = 0$$

$$D_F(4 \rightarrow 2) = 0$$

$$D_F(5 \rightarrow 3) = 0$$

$$D_F(6 \rightarrow 4) = -4$$

$$D_F(7 \rightarrow 3) = -3$$

$$D_F(8 \rightarrow 4) = -3$$

04. 折叠架构的寄存器最小化——Biquad滤波器

■ 对最原始的DFG折叠会怎样？



$$D_F(U \rightarrow V) = Nw(e) - P_u + v - u$$

$$D_F(1 \rightarrow 2) = -3$$

$$D_F(1 \rightarrow 5) = 0$$

$$D_F(1 \rightarrow 6) = 2$$

$$D_F(1 \rightarrow 7) = 7$$

$$D_F(1 \rightarrow 8) = 5$$

$$D_F(3 \rightarrow 1) = 0$$

$$D_F(4 \rightarrow 2) = 0$$

$$D_F(5 \rightarrow 3) = 0$$

$$D_F(6 \rightarrow 4) = -4$$

$$D_F(7 \rightarrow 3) = -3$$

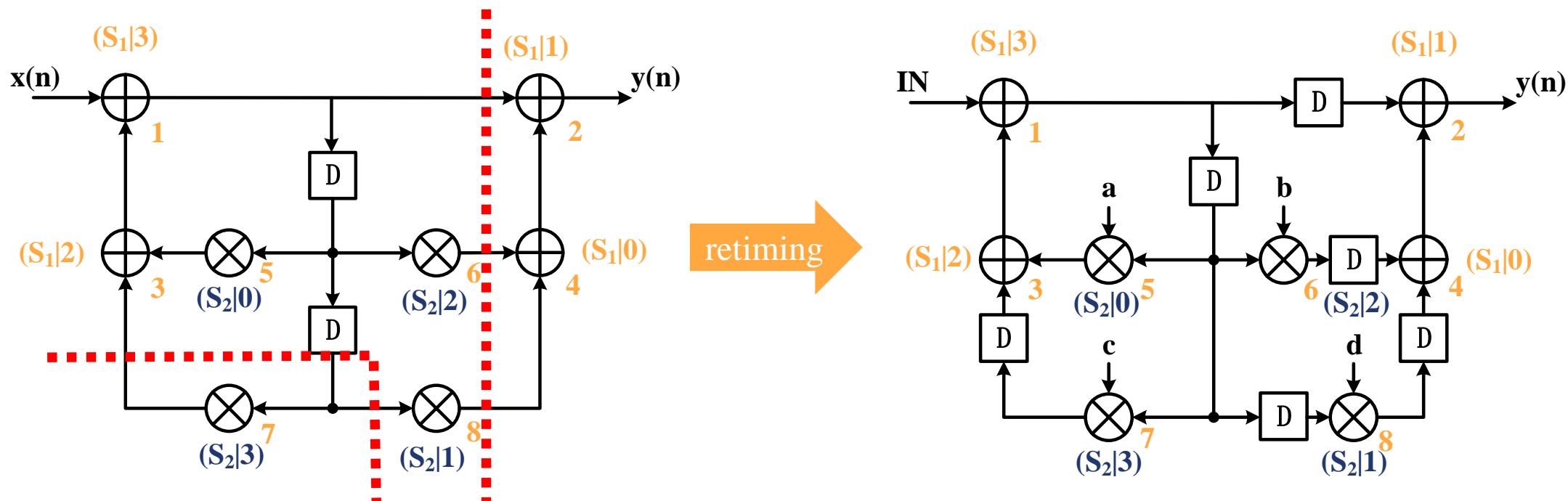
$$D_F(8 \rightarrow 4) = -3$$

04. 折叠架构的寄存器最小化——基本过程

1. 进行折叠的重定时
2. 写出折叠方程
3. 用折叠方程构造寿命表
4. 画出寿命图并确定所需寄存器数
5. 进行前向-后向寄存器分配
6. 画出最小寄存器数的折叠架构

04. 折叠架构的寄存器最小化——Biquad滤波器

1. 进行折叠的重定时



04. 折叠架构的寄存器最小化——Biquad滤波器

2. 列出折叠方程组

$$D_F(U \rightarrow V) = Nw(e) - P_u + v - u$$

$$D_F(1 \rightarrow 2) = 4(1) - 1 + 1 - 3 = 1$$

$$D_F(1 \rightarrow 5) = 4(1) - 1 + 0 - 3 = 0$$

$$D_F(1 \rightarrow 6) = 4(1) - 1 + 2 - 3 = 2$$

$$D_F(1 \rightarrow 7) = 4(1) - 1 + 3 - 3 = 3$$

$$D_F(1 \rightarrow 8) = 4(2) - 1 + 1 - 3 = 5$$

$$D_F(3 \rightarrow 1) = 4(0) - 1 + 3 - 2 = 0$$

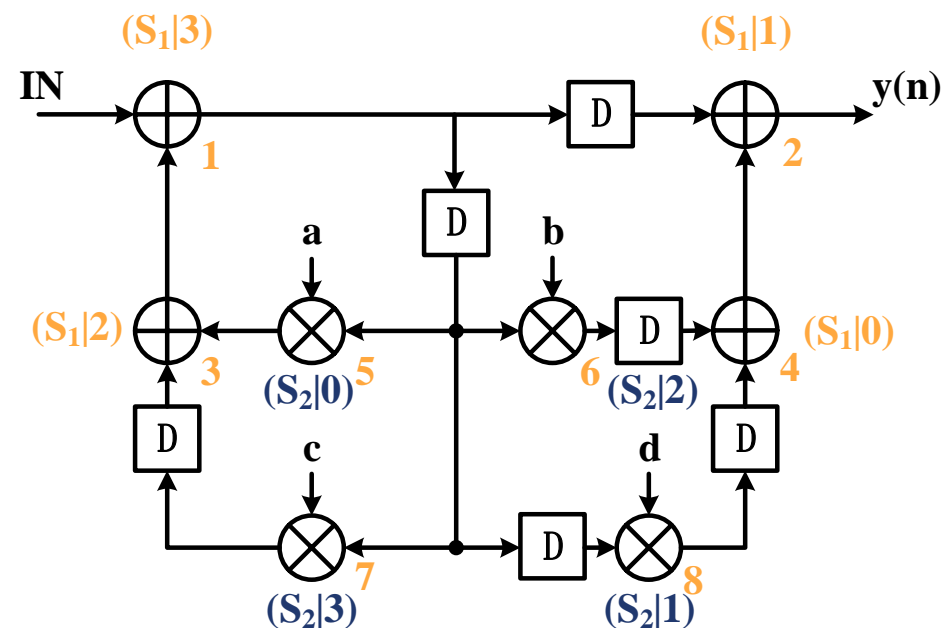
$$D_F(4 \rightarrow 2) = 4(0) - 1 + 1 - 0 = 0$$

$$D_F(5 \rightarrow 3) = 4(0) - 2 + 2 - 0 = 0$$

$$D_F(6 \rightarrow 4) = 4(1) - 2 + 0 - 2 = 0$$

$$D_F(7 \rightarrow 3) = 4(1) - 2 + 2 - 3 = 1$$

$$D_F(8 \rightarrow 4) = 4(1) - 2 + 0 - 1 = 1$$



04. 折叠架构的寄存器最小化——Biquad滤波器

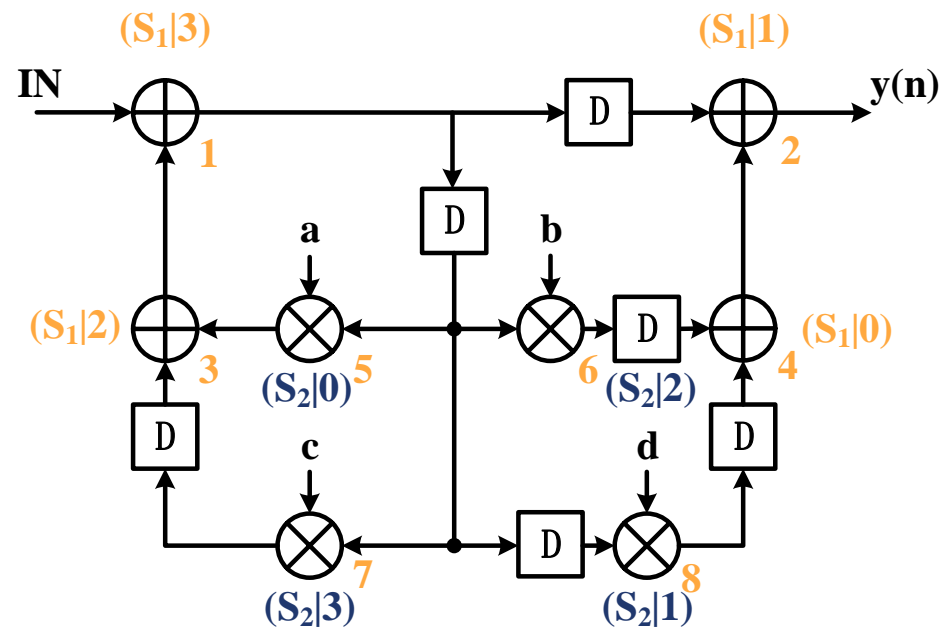
3. 构造寿命表

寿命 = $T_{output} - T_{input}$

$$T_{output} = T_{input} + \max_V \{D_F(U \rightarrow V)\}$$

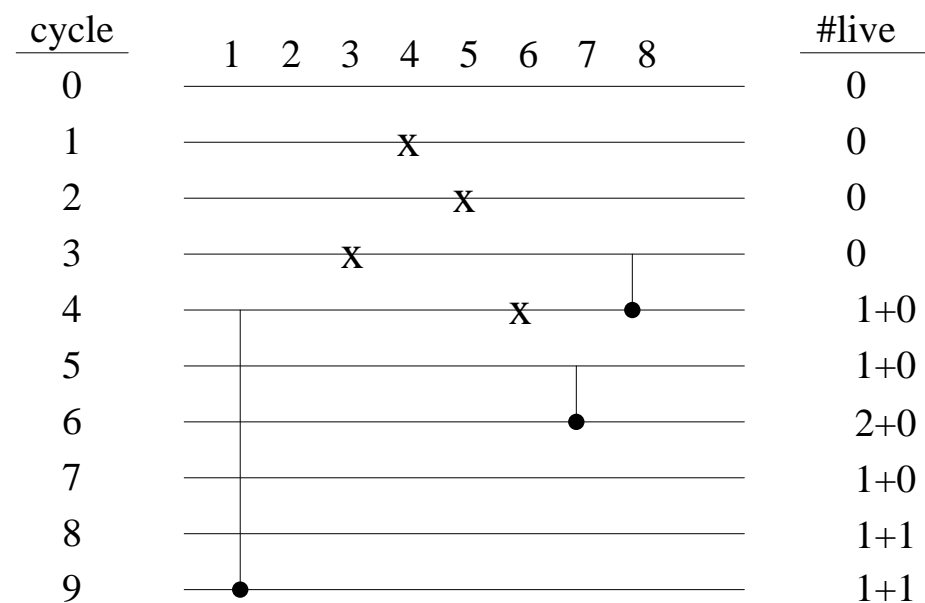
$$T_{input} = u + P_U$$

变量	寿命
1	4→9
2	-
3	3→3
4	1→1
5	2→2
6	4→4
7	5→6
8	3→4



04. 折叠架构的寄存器最小化——Biquad滤波器

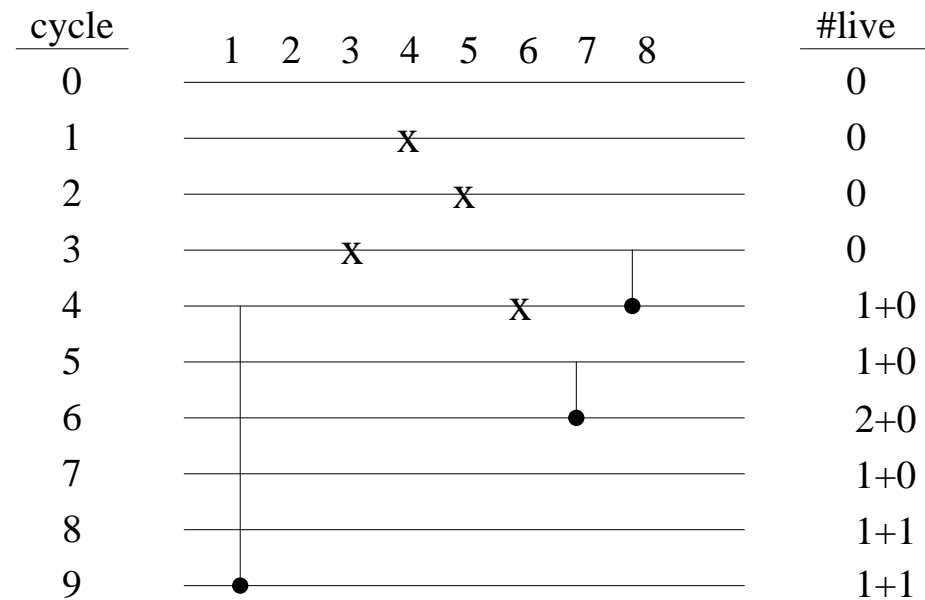
4. 画出寿命图并确定所需寄存器数



最大激活寄存器数=2
所需寄存器数=2

04. 折叠架构的寄存器最小化——Biquad滤波器

5. 进行前向-后向寄存器分配

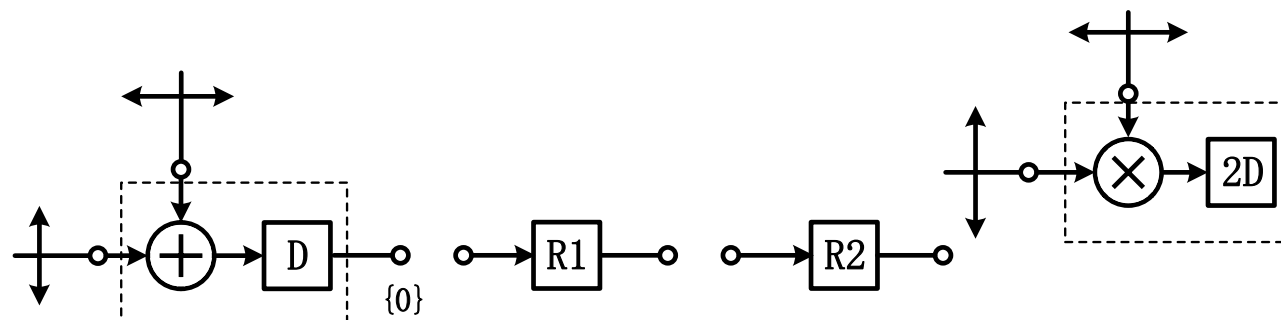
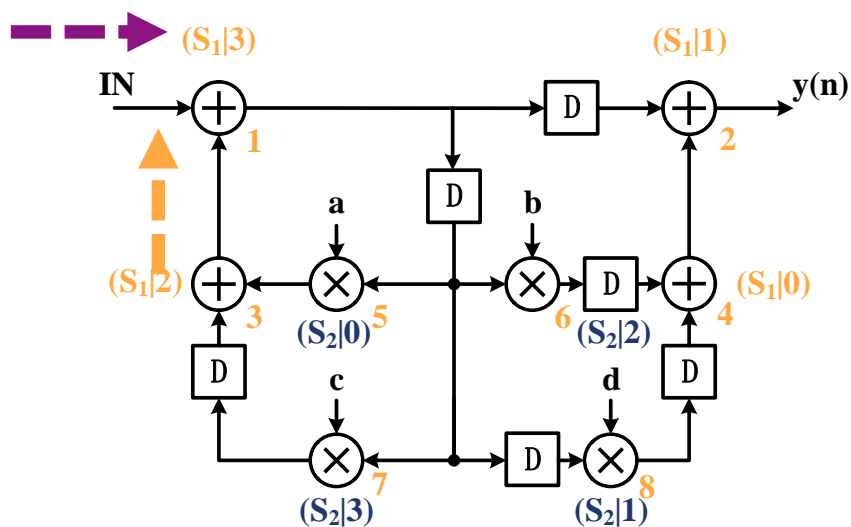


cycle	In	R1	R2	Out	
0					
1					
2					
3	N ₈				
4(0)	N ₁	<u>N₈</u>		<u>N₈</u>	→n ₅
5(1)	N ₇	N ₁			n ₁ →n ₂
6(2)		<u>N₇</u>	N ₁	<u>N₇</u>	n ₁ →n ₆
7(3)			N ₁		n ₁ →n ₇
8(0)			N ₁		
9(1)			N ₁	<u>N₁</u>	→n ₈

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点1 ($S_1|3$) 输入:

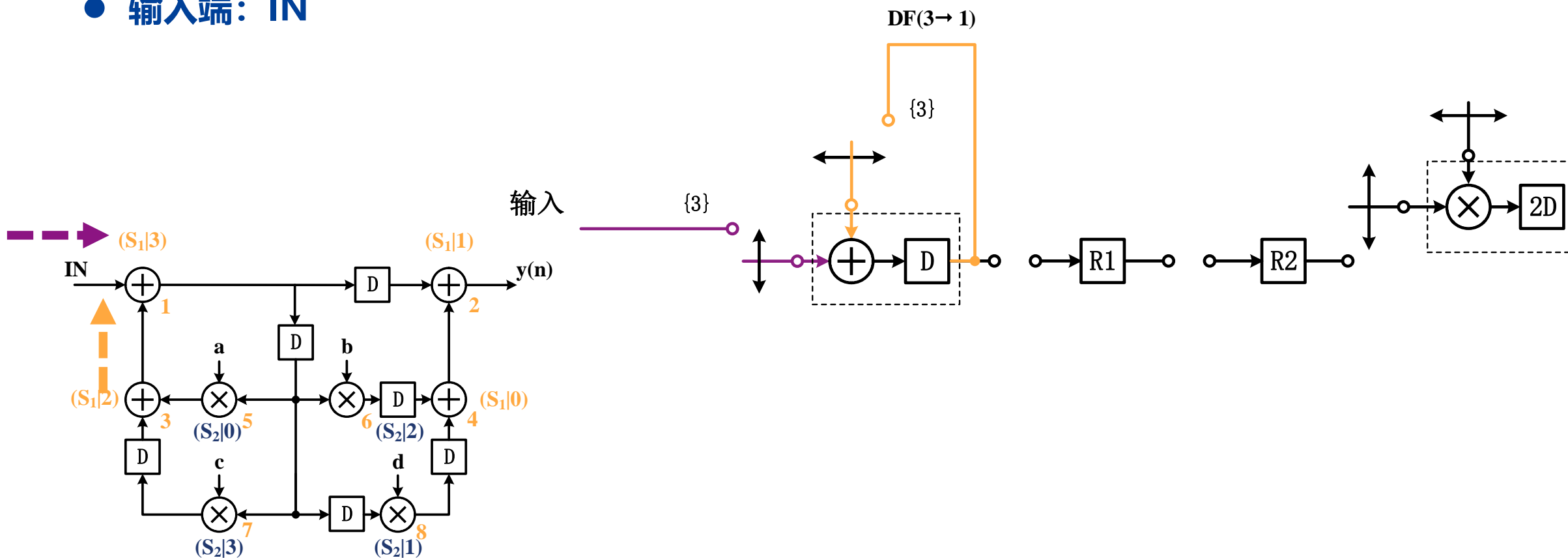
- 节点3: $D_F(3 \rightarrow 1) = 0$
- 输入端: IN



04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点1 ($S_1|3$) 输入:

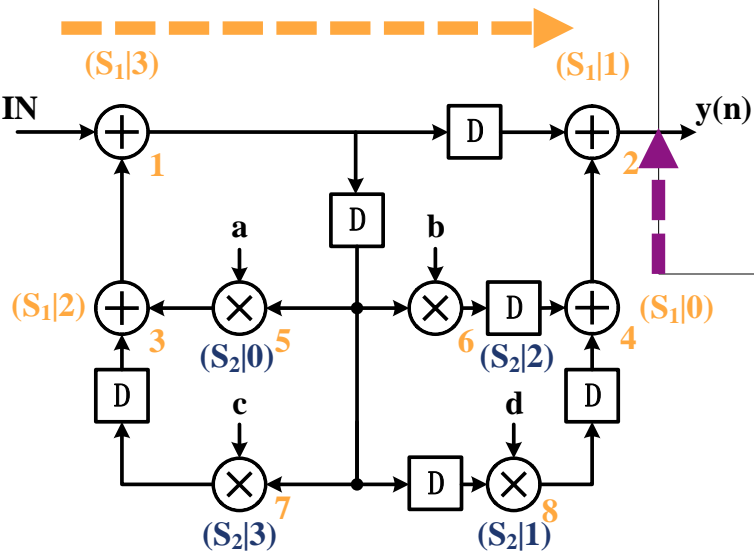
- 节点3: $D_F(3 \rightarrow 1) = 0$
- 输入端: IN



04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点2 ($S_1|1$) 输入:

- 节点4: $D_F(4 \rightarrow 2) = 0$
- 节点1: $D_F(1 \rightarrow 2) = 1$, 经过R1
 - 0 : $S_1 \rightarrow R1$
 - 1 : $R1 \rightarrow S_1$

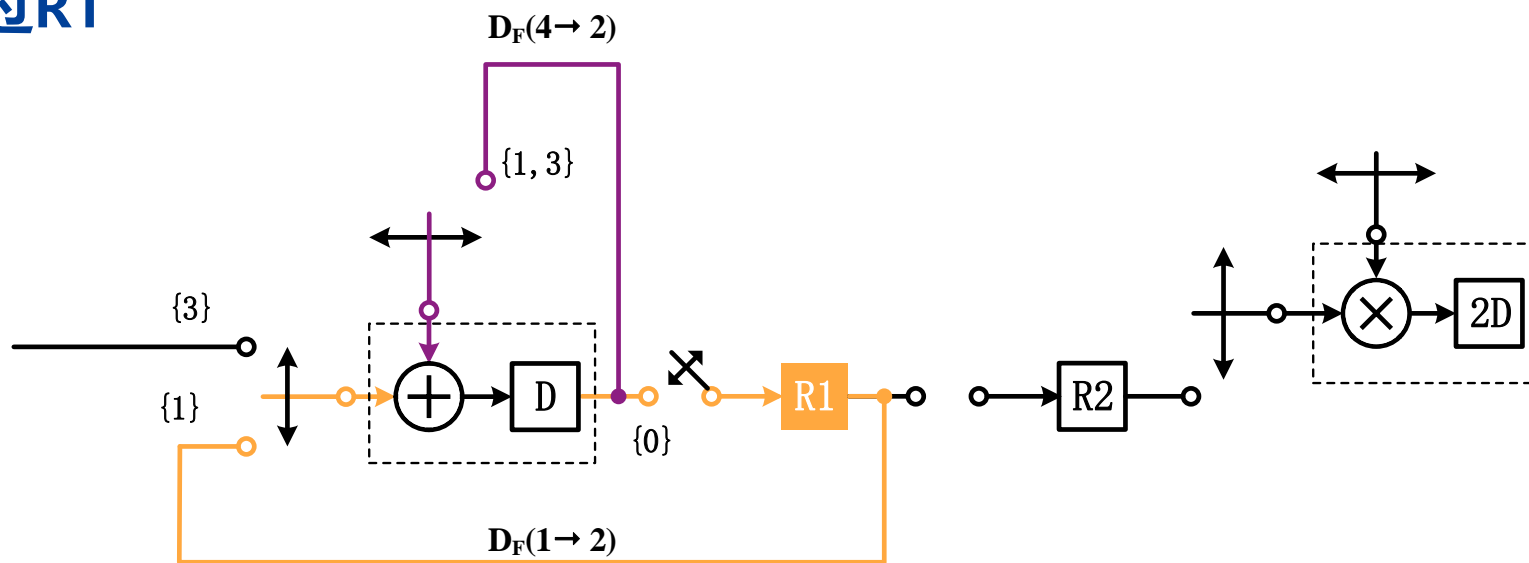
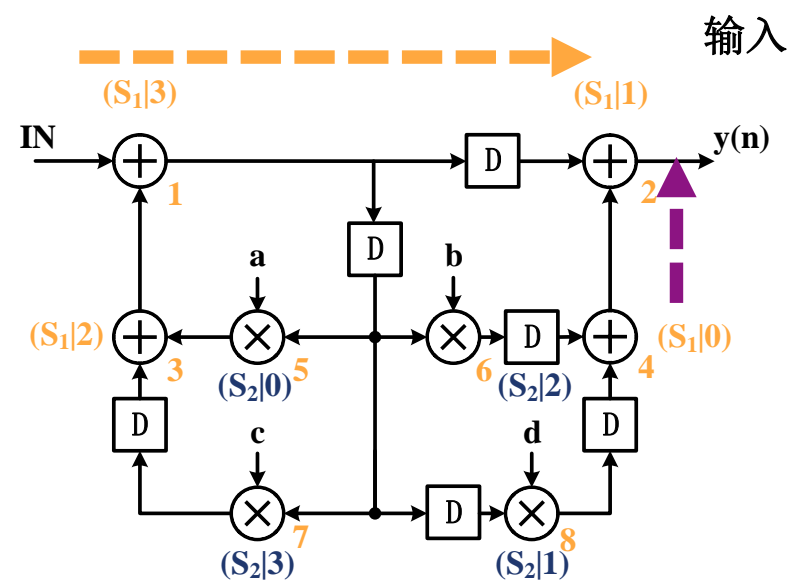


cycle	In	R1	R2	Out	
4(0)	N_1	N_8		N_8	
5(1)	N_7	N_1			$n_1 \rightarrow n_2$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点2 ($S_1|1$) 输入:

- 节点4: $D_F(4 \rightarrow 2) = 0$
- 节点1: $D_F(1 \rightarrow 2) = 1$, 经过R1
 - 0: $S_1 \rightarrow R1$
 - 1: $R1 \rightarrow S_1$

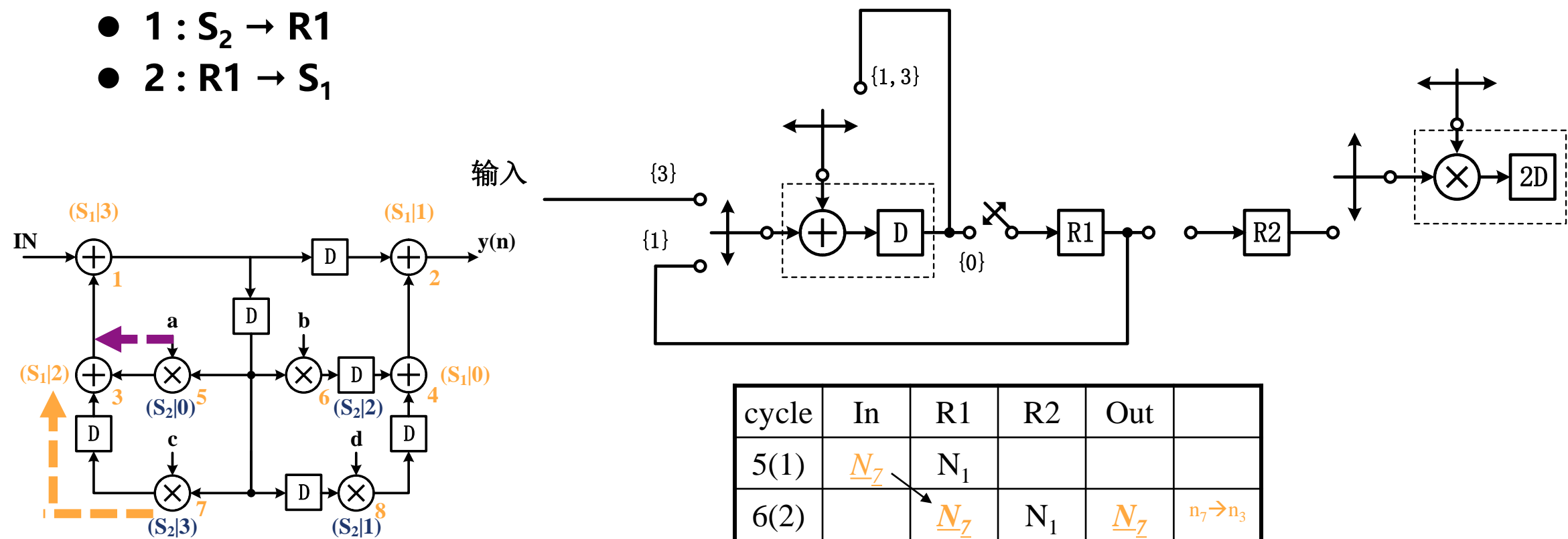


cycle	In	R1	R2	Out	
4(0)	N_1	N_8		N_8	
5(1)	N_7	N_1			$n_1 \rightarrow n_2$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点3 ($S_1|2$) 输入:

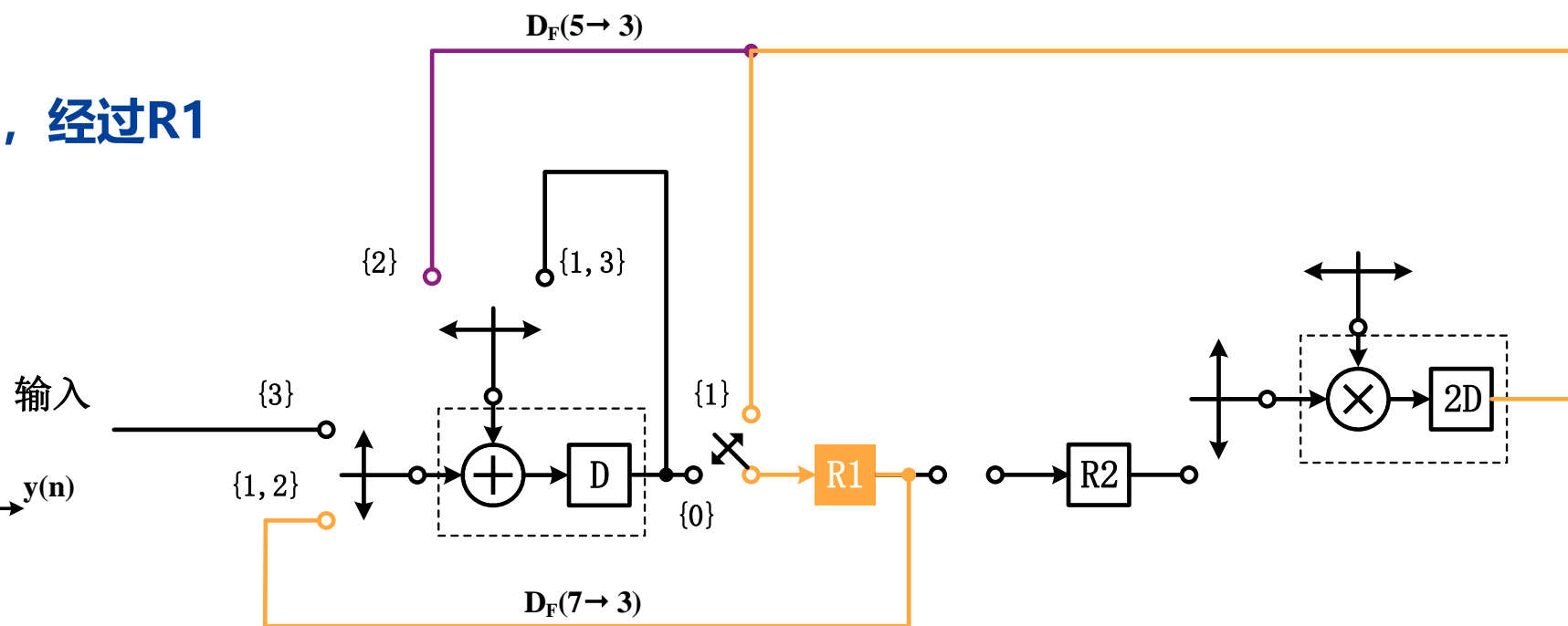
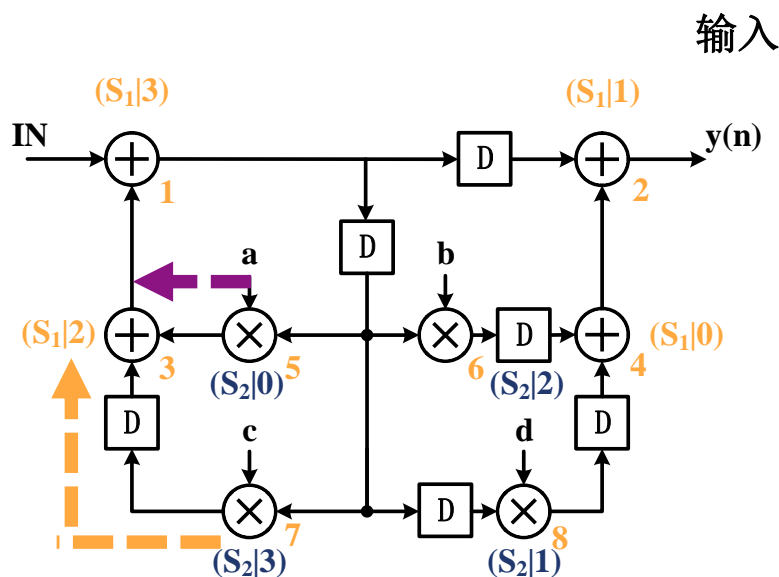
- 节点5: $D_F(5 \rightarrow 3) = 0$
- 节点7: $D_F(7 \rightarrow 3) = 1$, 经过R1
 - 1: $S_2 \rightarrow R1$
 - 2: $R1 \rightarrow S_1$



04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点3 ($S_1|2$) 输入:

- **节点5: $D_F(5 \rightarrow 3) = 0$**
- **节点7: $D_F(7 \rightarrow 3) = 1$, 经过R1**
 - **1 : $S_2 \rightarrow R1$**
 - **2 : $R1 \rightarrow S_1$**

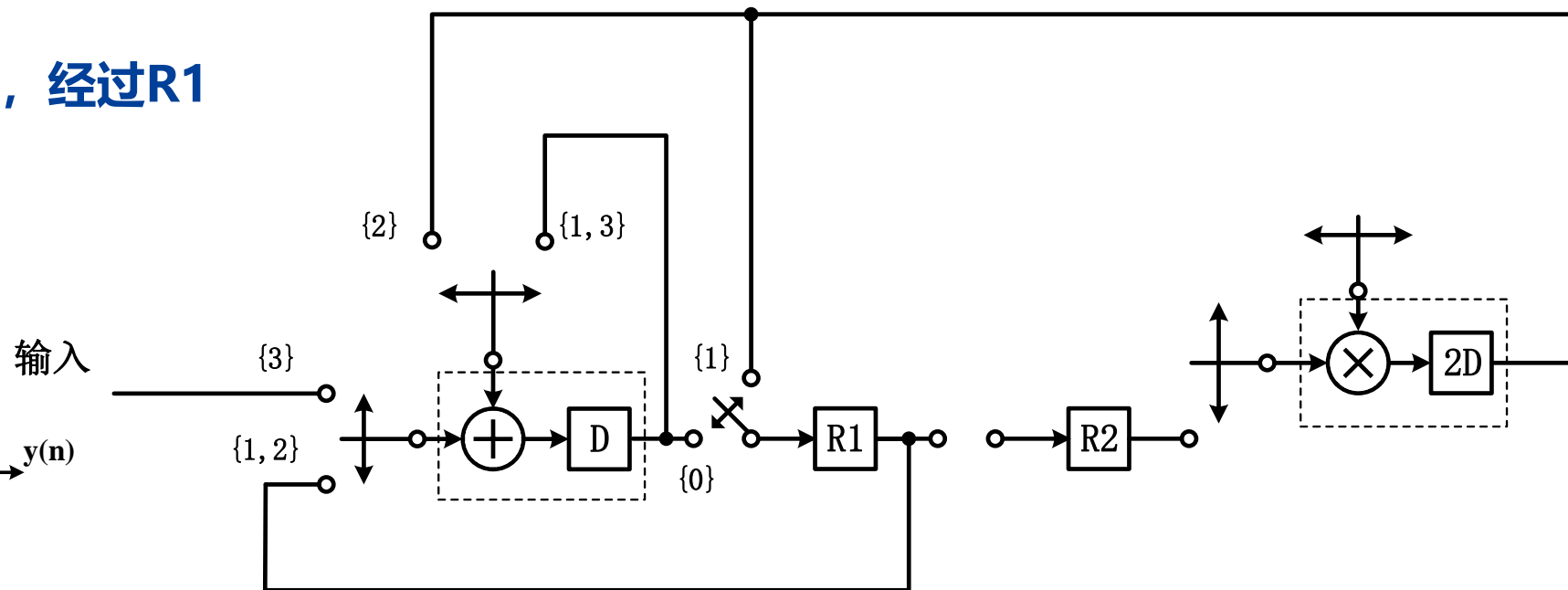
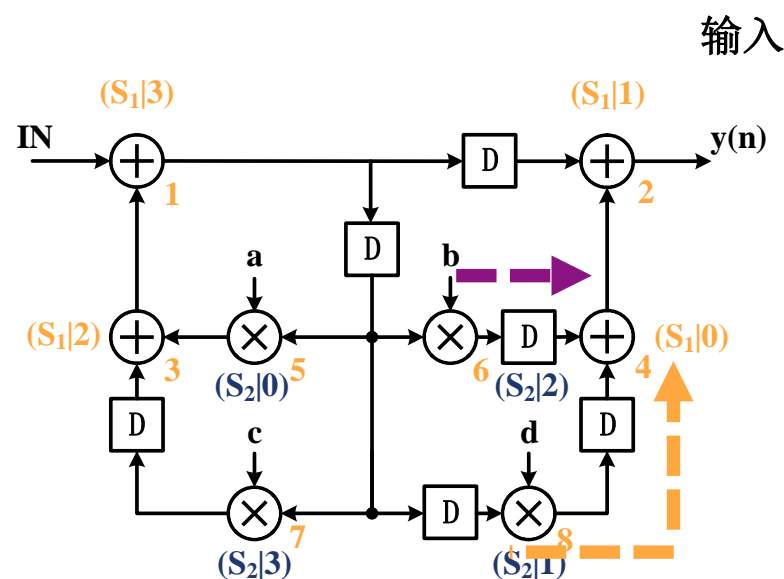


cycle	In	R1	R2	Out	
5(1)	\underline{N}_Z	N_1			
6(2)		\underline{N}_Z	N_1	\underline{N}_Z	$n_7 \rightarrow n_3$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点4 ($S_1|0$) 输入:

- 节点6: $D_F(6 \rightarrow 4) = 0$
- 节点8: $D_F(8 \rightarrow 4) = 1$, 经过R1
 - 3 : $S_2 \rightarrow R1$
 - 0 : $R1 \rightarrow S_1$

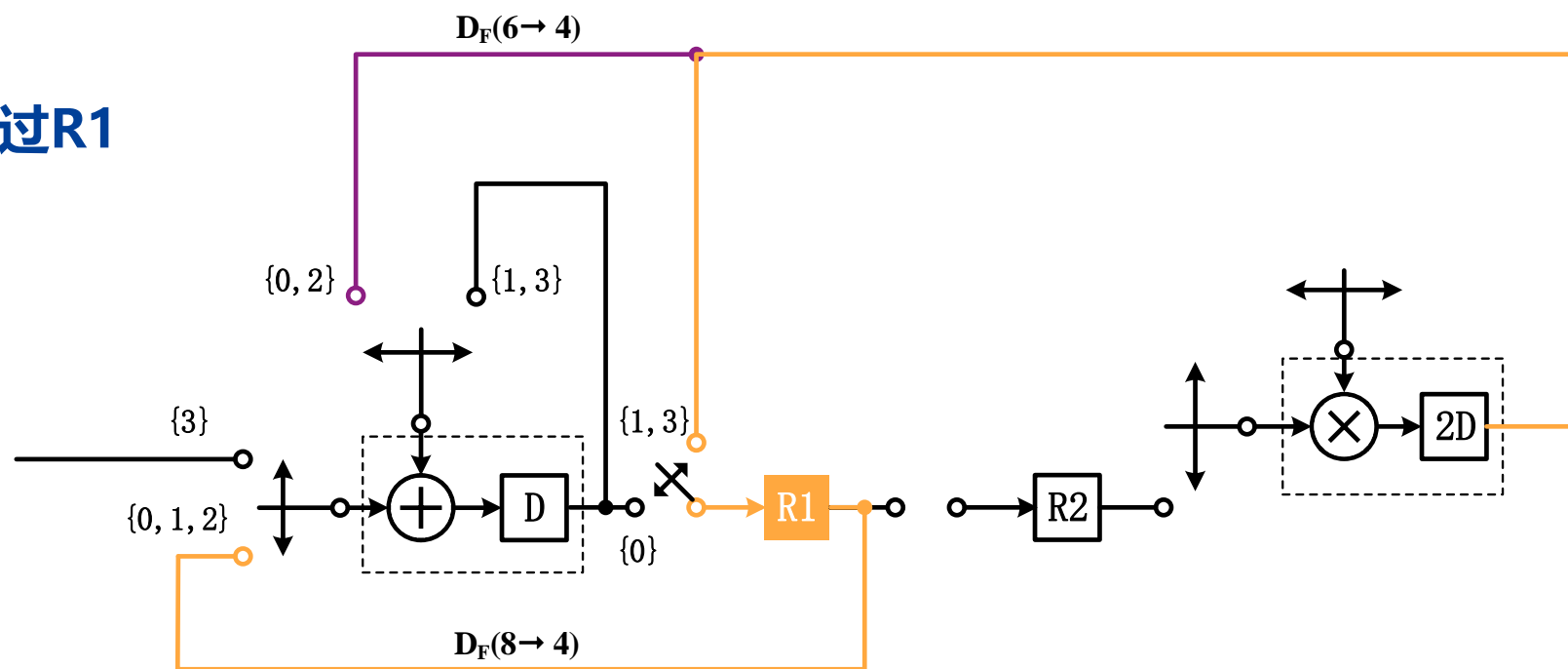
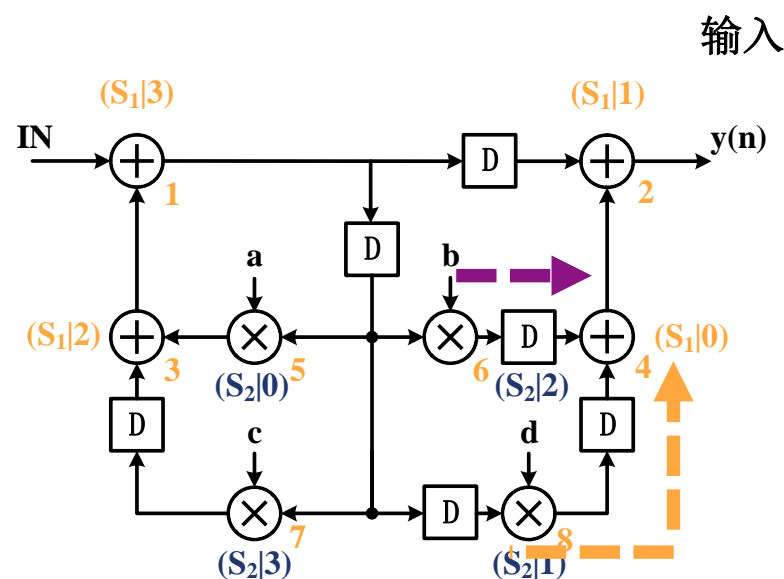


cycle	In	R1	R2	Out	
3	N_8				
4(0)	N_1	N_8		N_8	$n_8 \rightarrow n_4$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点4 ($S_1|0$) 输入:

- 节点6: $D_F(6 \rightarrow 4) = 0$
- 节点8: $D_F(8 \rightarrow 4) = 1$, 经过R1
 - 3 : $S_2 \rightarrow R1$
 - 0 : $R1 \rightarrow S_1$

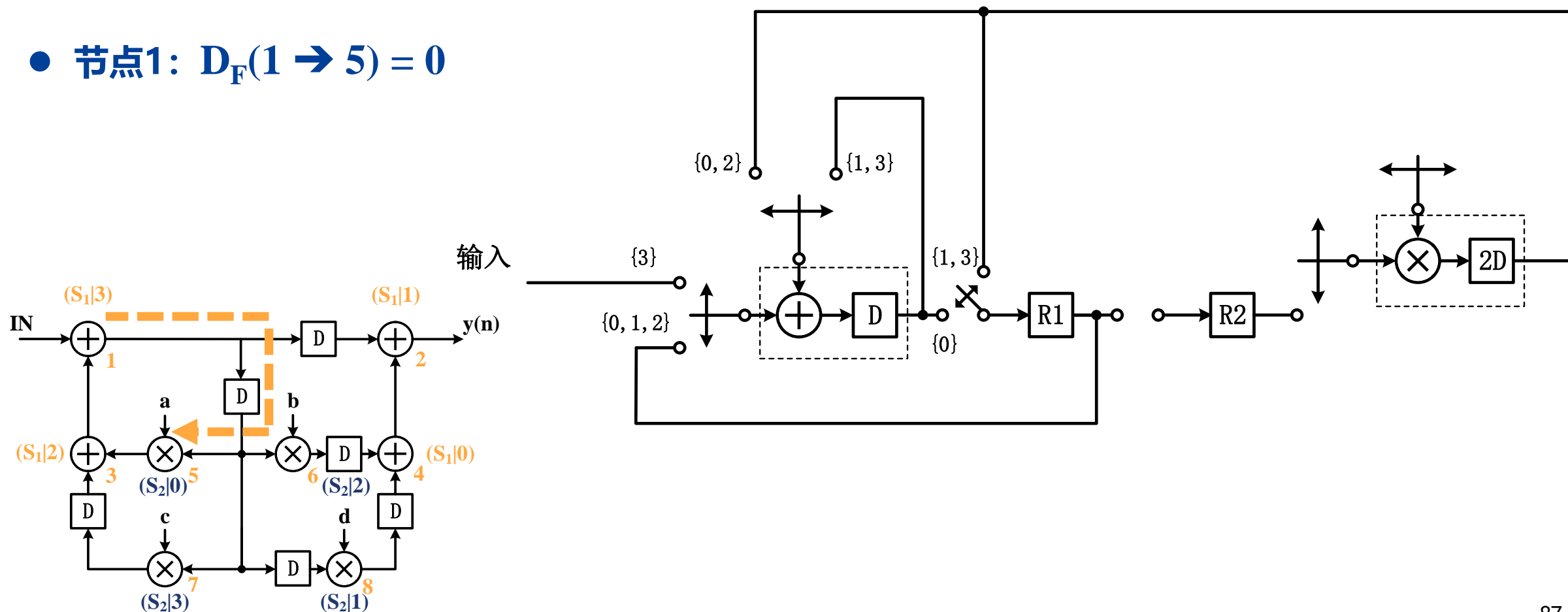


cycle	In	R1	R2	Out	
3	N_8				
4(0)	N_1	N_8		N_8	$n_8 \rightarrow n_4$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点5 ($S_2|0$) 输入:

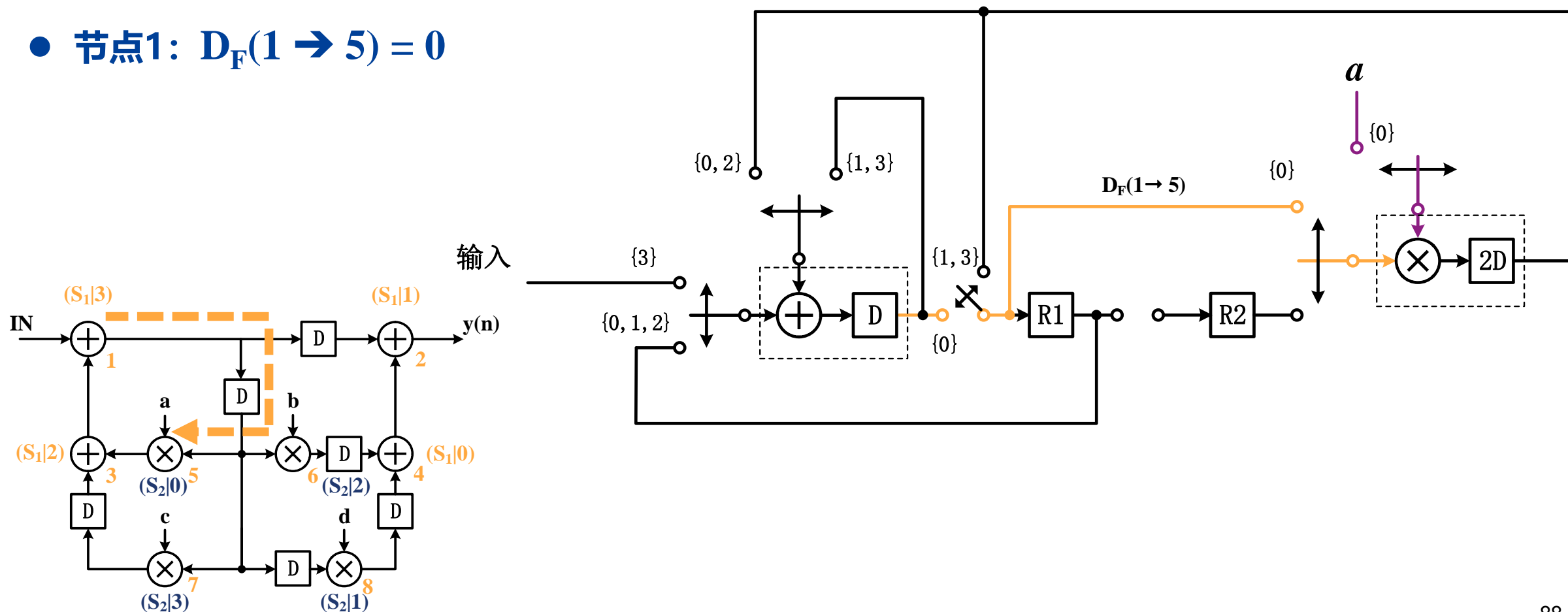
- 系数: a
- 节点1: $D_F(1 \rightarrow 5) = 0$



04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点5 ($S_2|0$) 输入:

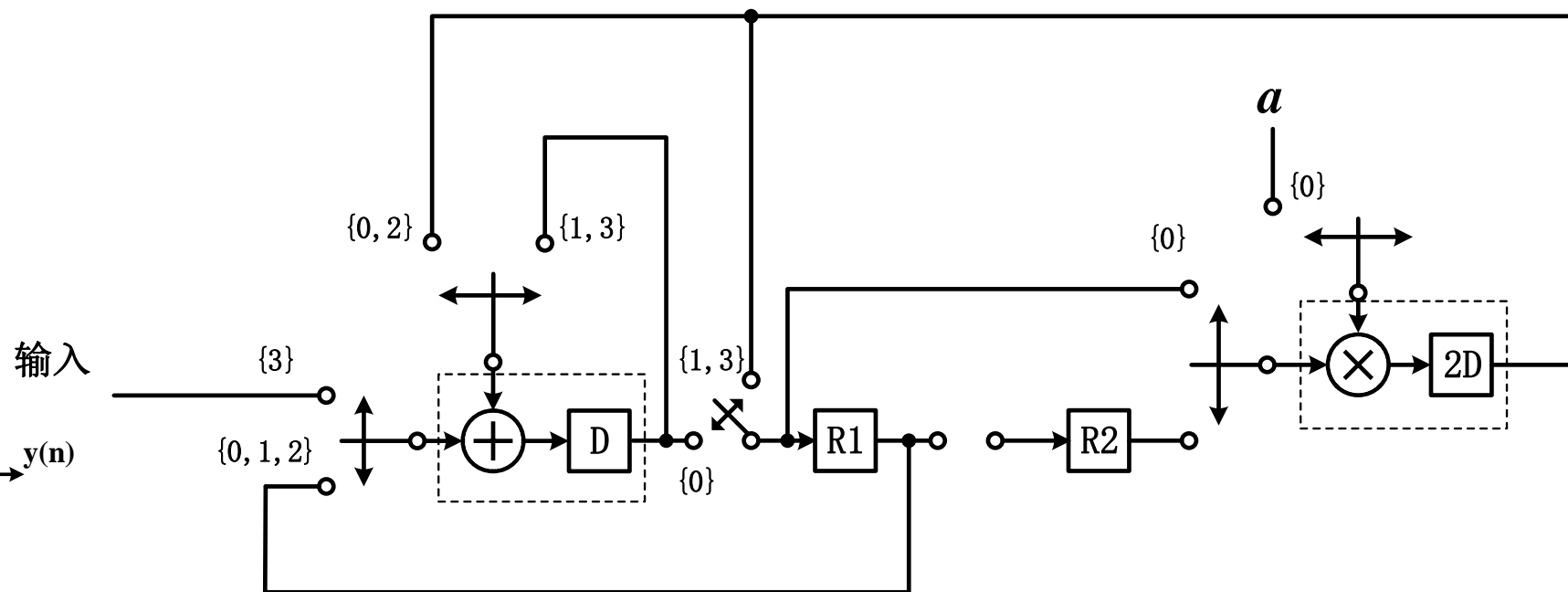
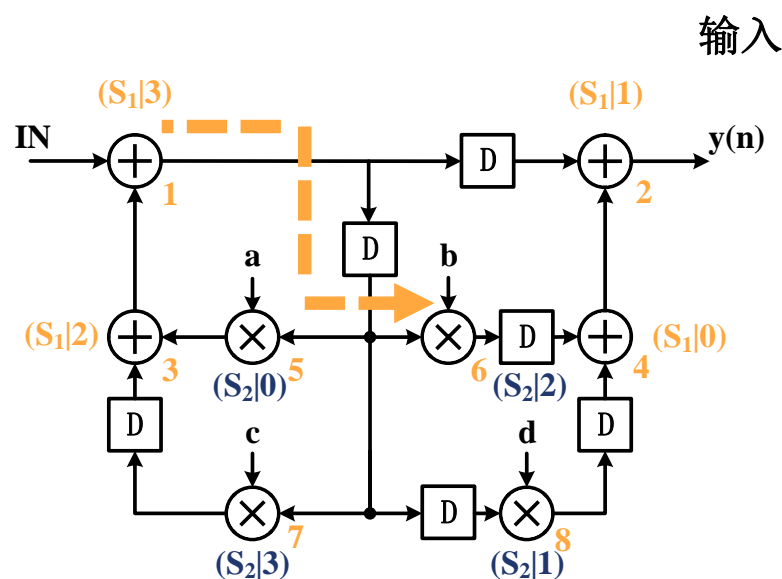
- 系数: a
- 节点1: $D_F(1 \rightarrow 5) = 0$



04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点6 ($S_2|2$) 输入:

- 系数: b
- 节点1: $D_F(1 \rightarrow 6) = 2$
 - $0 : S_1 \rightarrow R1$
 - $1 : R1 \rightarrow R2$
 - $2 : R2 \rightarrow S_2$

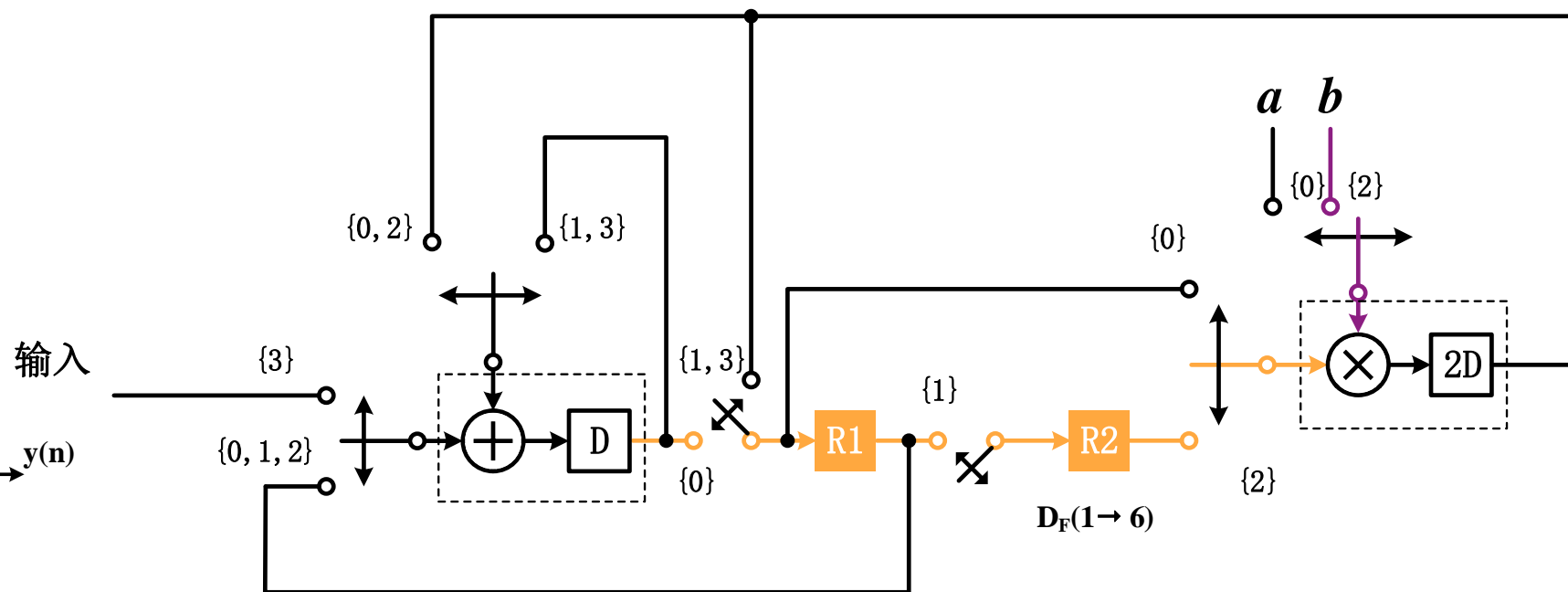
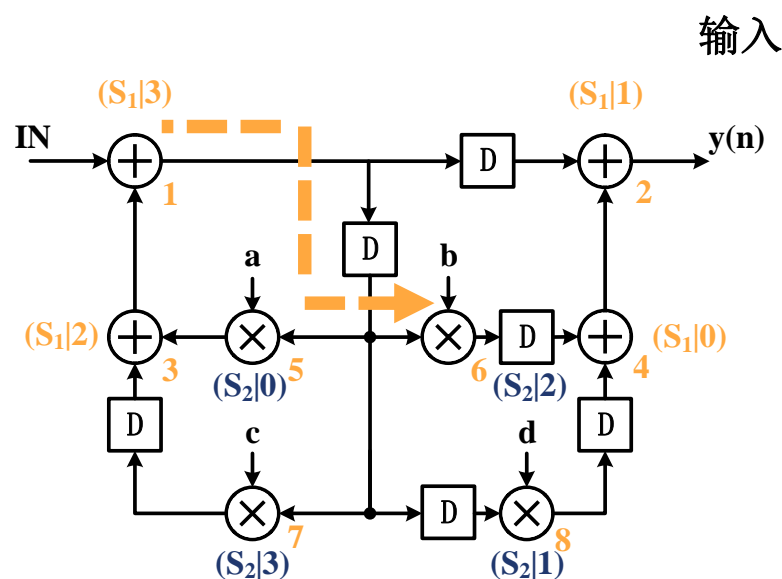


cycle	In	R1	R2	Out	
4(0)	$\underline{N_I}$	$\mathbf{N_8}$		$\mathbf{N_8}$	
5(1)	$\mathbf{N_7}$	$\underline{N_I}$			$n_1 \rightarrow n_2$
6(2)		$\mathbf{N_7}$	$\underline{N_I}$	$\mathbf{N_7}$	$n_1 \rightarrow n_6$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点6 ($S_2|2$) 输入:

- 系数: b
- 节点1: $D_F(1 \rightarrow 6) = 2$
 - $0 : S_1 \rightarrow R1$
 - $1 : R1 \rightarrow R2$
 - $2 : R2 \rightarrow S_2$

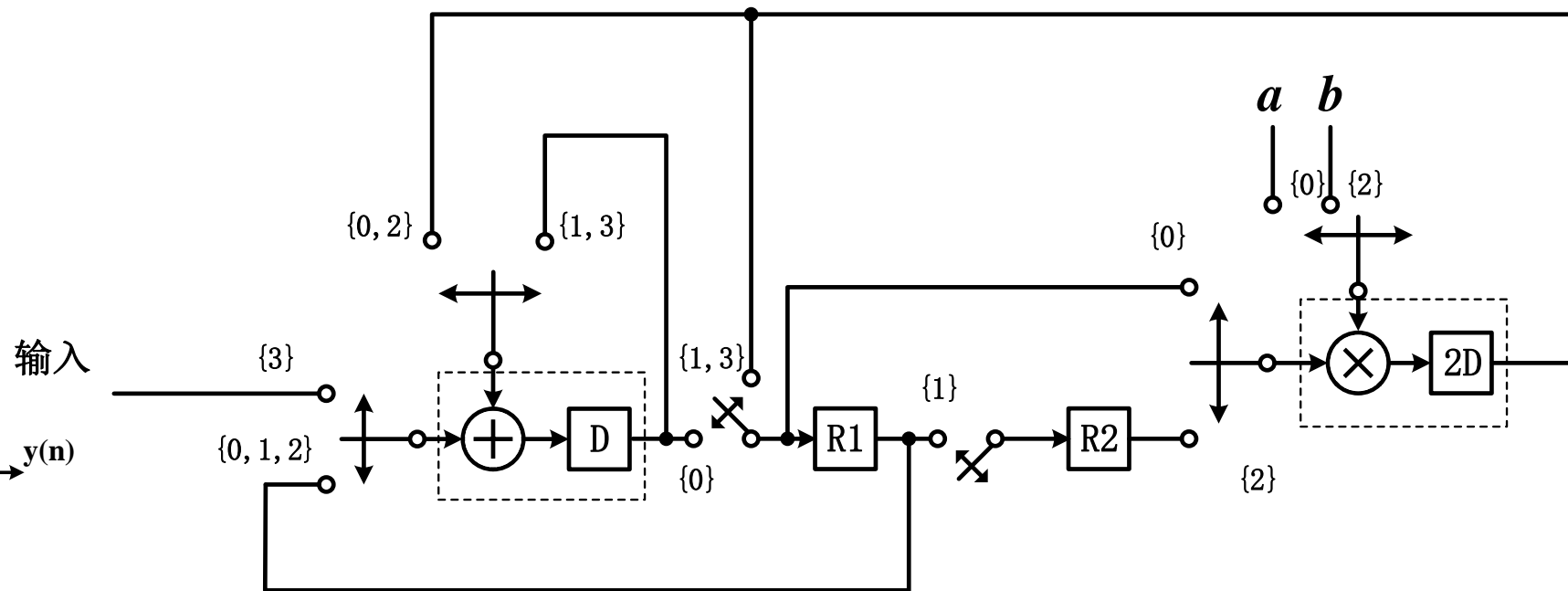
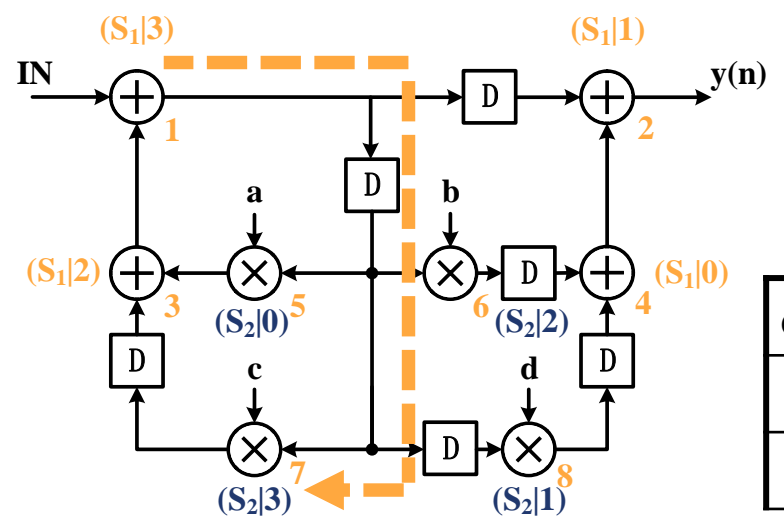


cycle	In	R1	R2	Out	
4(0)	$\underline{N_I}$	$\mathbf{N_8}$		$\mathbf{N_8}$	
5(1)	$\mathbf{N_7}$	$\underline{N_I}$			$n_1 \rightarrow n_2$
6(2)		$\mathbf{N_7}$	$\underline{N_I}$	$\mathbf{N_7}$	$n_1 \rightarrow n_6$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点7 ($S_2|3$) 输入:

- 系数: c
- 节点1: $D_F(1 \rightarrow 7) = 3$
 - $0 : S_1 \rightarrow R1$
 - $1 : R1 \rightarrow R2$
 - $2 : R2 \rightarrow R2$
 - $3 : R2 \rightarrow S_2$

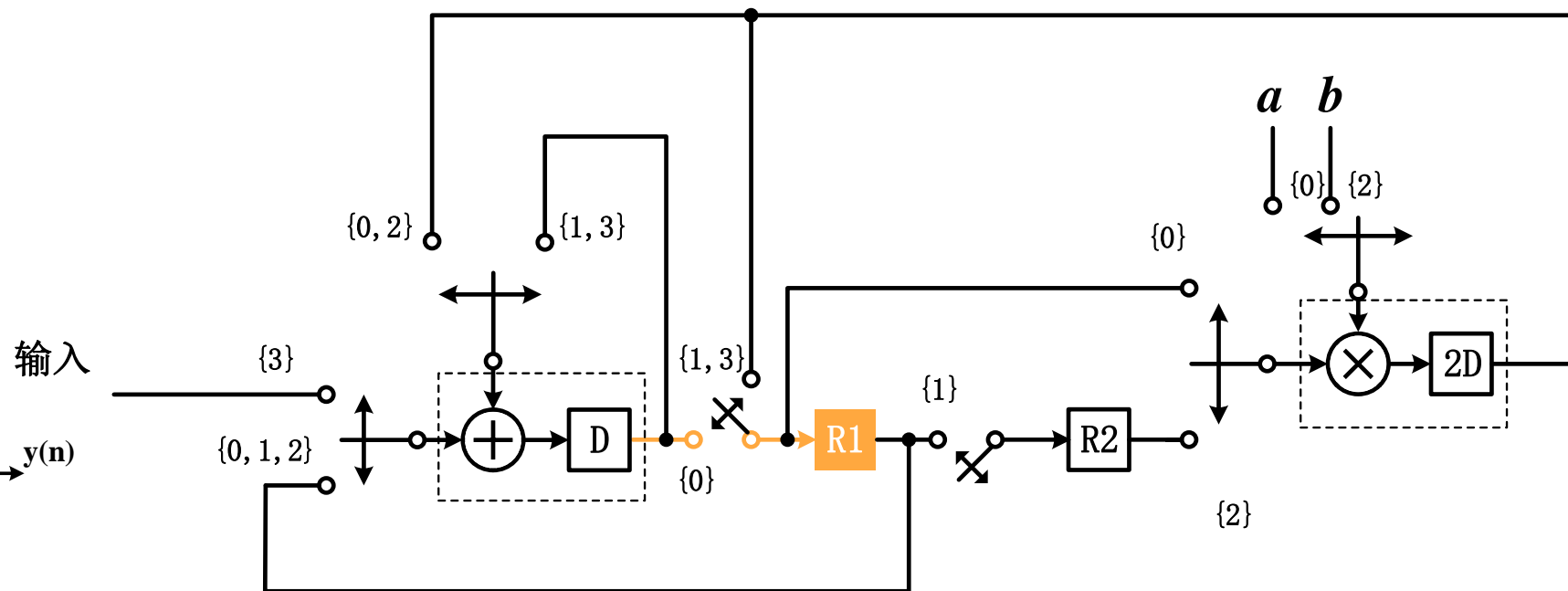
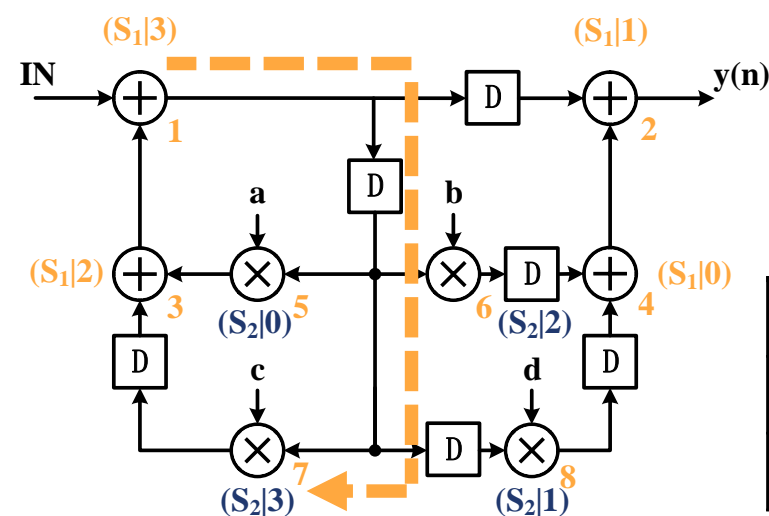


cycle	In	R1	R2	Out		cycle	In	R1	R2	Out	
4(0)	N_I	N_8		N_8	$n_1 \rightarrow n_5$	6(2)		N_7	N_I	N_7	$n_1 \rightarrow n_6$
5(1)	N_7	N_I			$n_1 \rightarrow n_2$	7(3)			N_I		$n_1 \rightarrow n_7$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点7 ($S_2|3$) 输入:

- 系数: c
- 节点1: $D_F(1 \rightarrow 7) = 3$
 - 0: $S_1 \rightarrow R1$
 - 1: $R1 \rightarrow R2$
 - 2: $R2 \rightarrow R2$
 - 3: $R2 \rightarrow S_2$

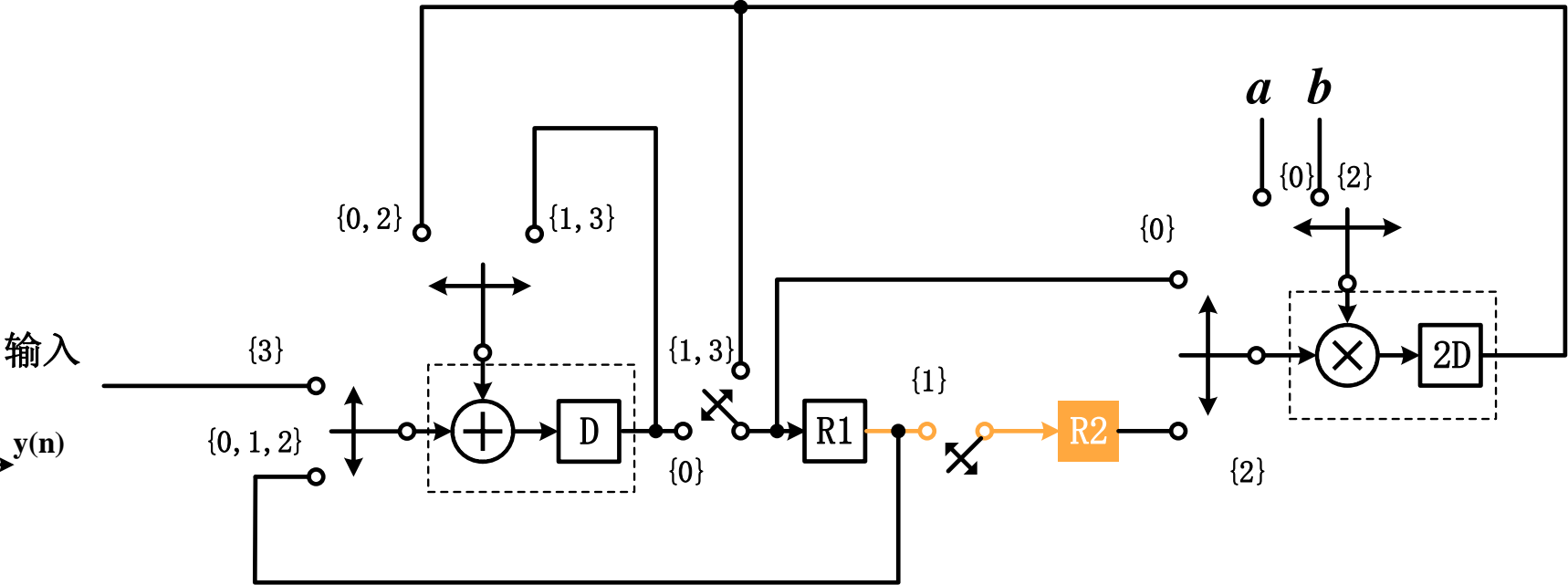
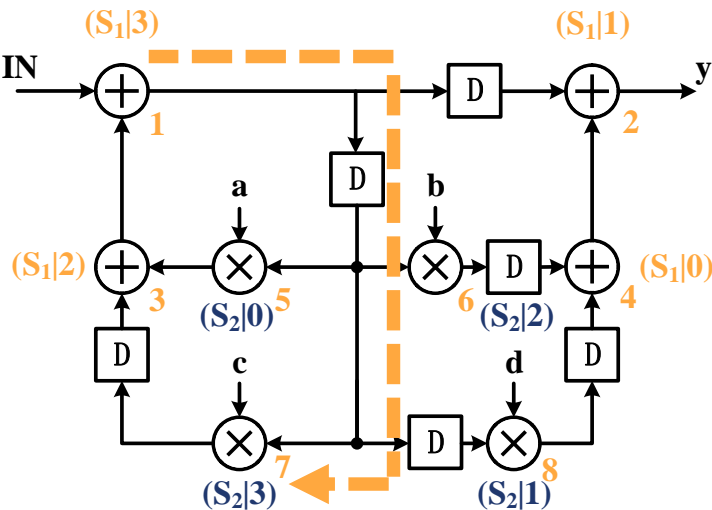


cycle	In	R1	R2	Out		cycle	In	R1	R2	Out	
4(0)	N_I	N_8		N_8	$n_1 \rightarrow n_5$	6(2)		N_7	N_I	N_7	$n_1 \rightarrow n_6$
5(1)	N_7	N_I			$n_1 \rightarrow n_2$	7(3)			N_I		$n_1 \rightarrow n_7$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点7 ($S_2|3$) 输入:

- 系数: c
- 节点1: $D_F(1 \rightarrow 7) = 3$
 - 0: $S_1 \rightarrow R1$
 - 1: $R1 \rightarrow R2$
 - 2: $R2 \rightarrow R2$
 - 3: $R2 \rightarrow S_2$

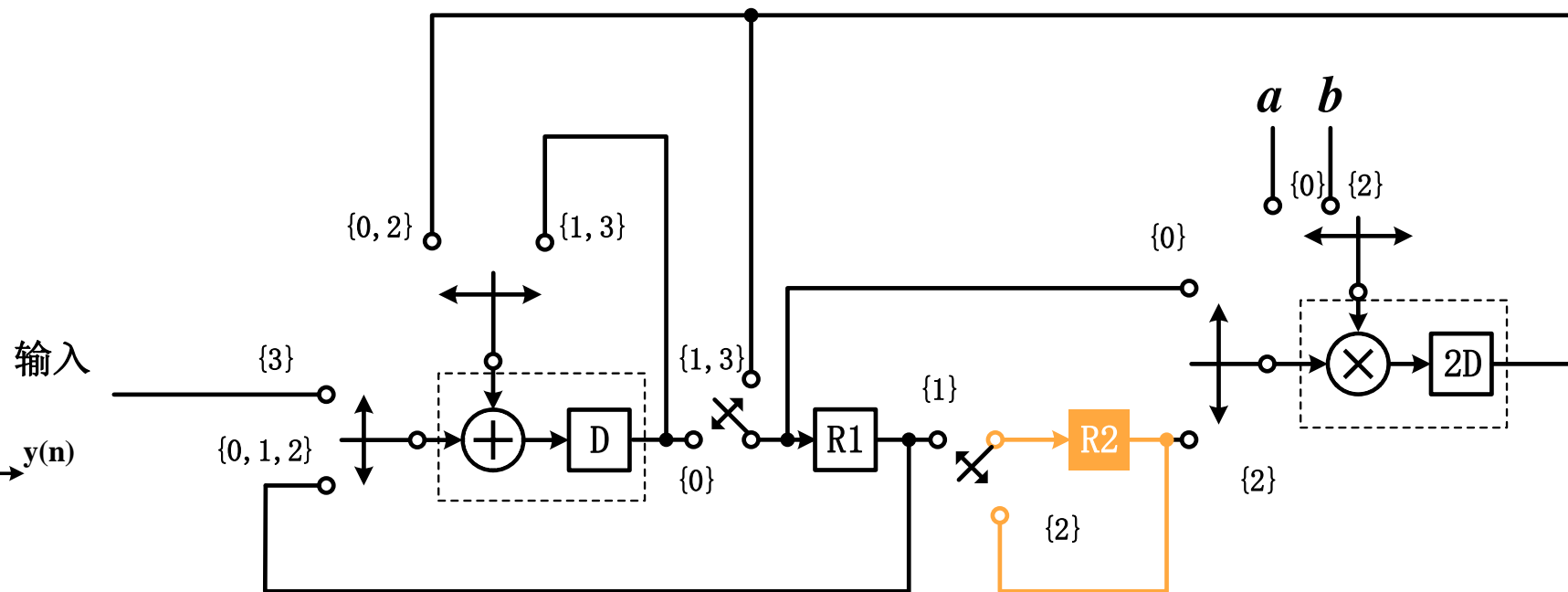
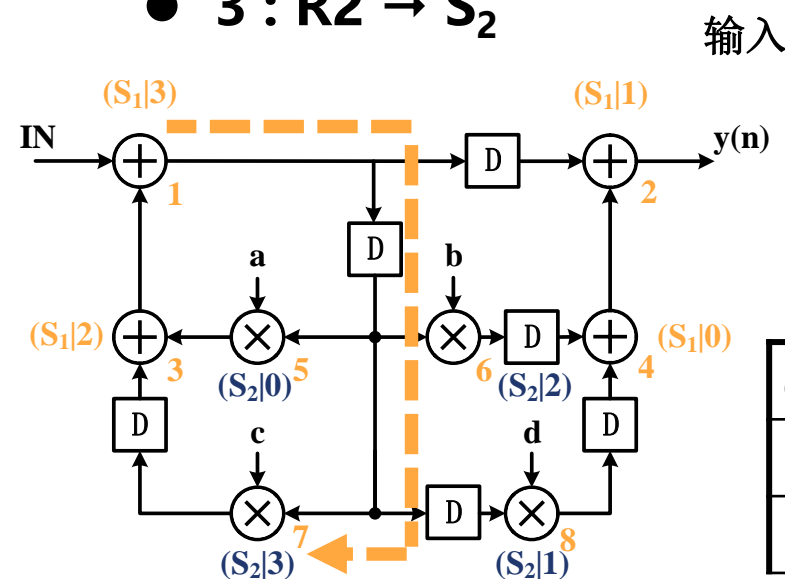


cycle	In	R1	R2	Out		cycle	In	R1	R2	Out	
4(0)	N_I	N_8		N_8	$n_1 \rightarrow n_5$	6(2)		N_7	N_I	N_7	$n_1 \rightarrow n_6$
5(1)	N_7	N_I			$n_1 \rightarrow n_2$	7(3)			N_I		$n_1 \rightarrow n_7$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点7 ($S_2|3$) 输入:

- 系数: c
- 节点1: $D_F(1 \rightarrow 7) = 3$
 - 0: $S_1 \rightarrow R1$
 - 1: $R1 \rightarrow R2$
 - 2: $R2 \rightarrow R2$
 - 3: $R2 \rightarrow S_2$

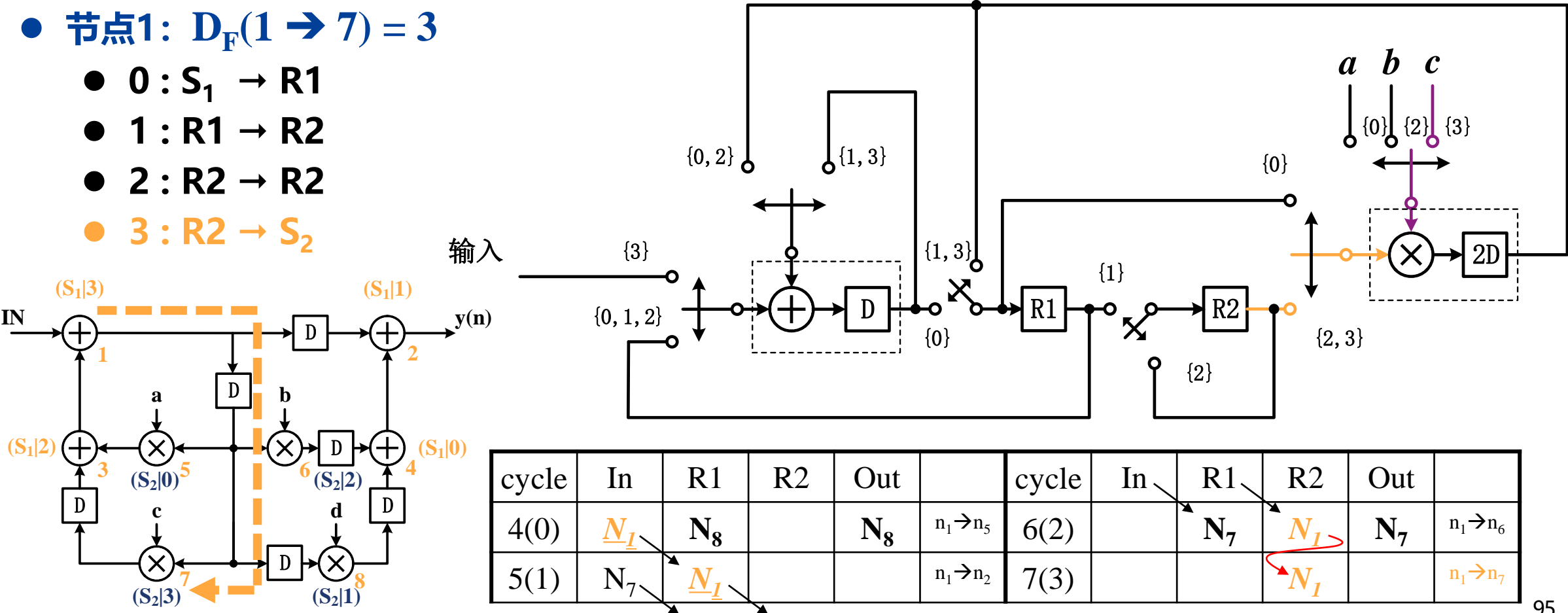


cycle	In	R1	R2	Out		cycle	In	R1	R2	Out	
4(0)	N_I	N_8		N_8	$n_1 \rightarrow n_5$	6(2)		N_7	N_I	N_7	$n_1 \rightarrow n_6$
5(1)	N_7	N_I			$n_1 \rightarrow n_2$	7(3)			N_I		$n_1 \rightarrow n_7$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点7 ($S_2|3$) 输入:

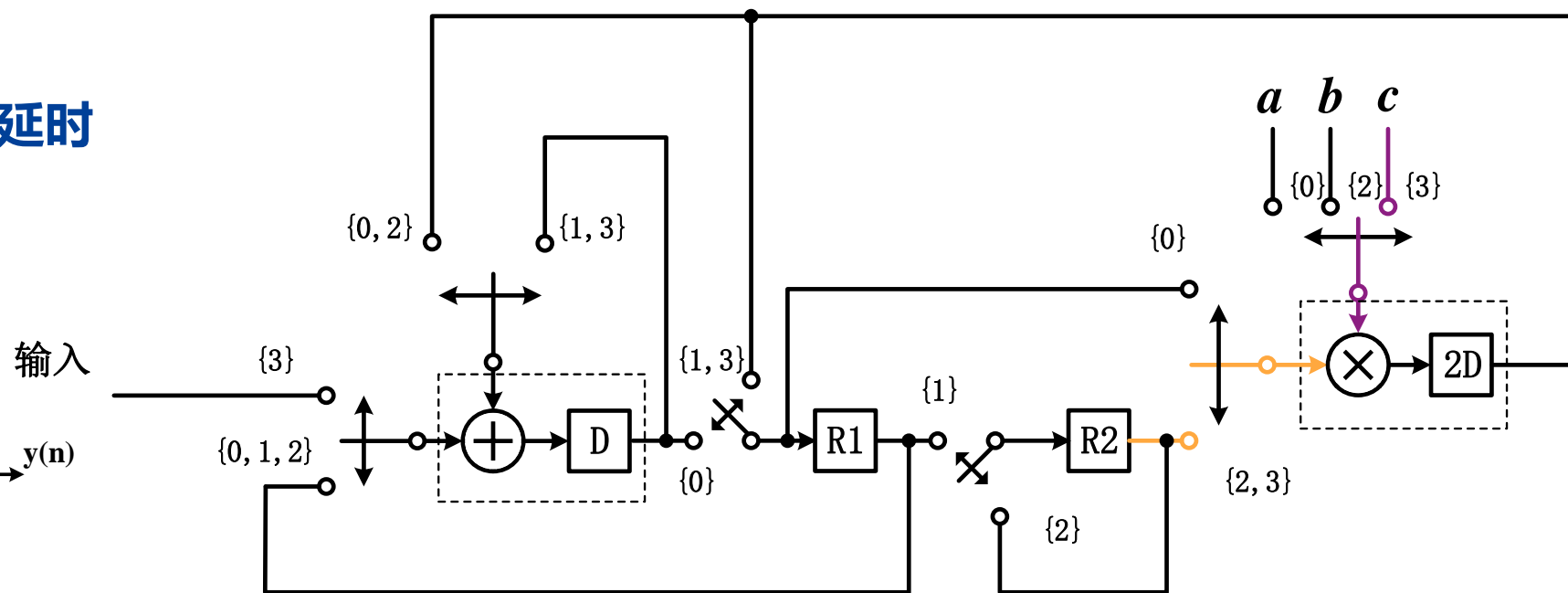
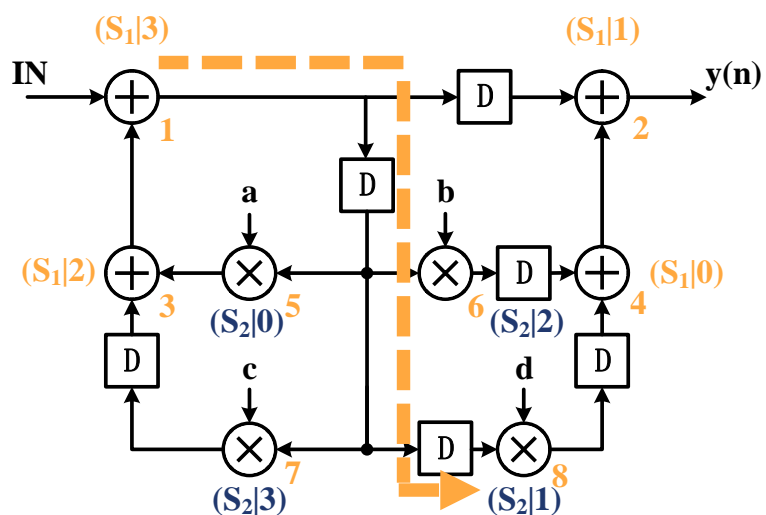
- 系数: c
- 节点1: $D_F(1 \rightarrow 7) = 3$
 - $0 : S_1 \rightarrow R1$
 - $1 : R1 \rightarrow R2$
 - $2 : R2 \rightarrow R2$
 - $3 : R2 \rightarrow S_2$



04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点8 ($S_2|1$) 输入:

- 系数: d
- 节点1: $D_F(1 \rightarrow 8) = 5$
 - 比 $D_F(1 \rightarrow 7)$ 多两个延时
 - $3 : R2 \rightarrow R2$
 - $0 : R2 \rightarrow R2$
 - $1 : R2 \rightarrow S_2$

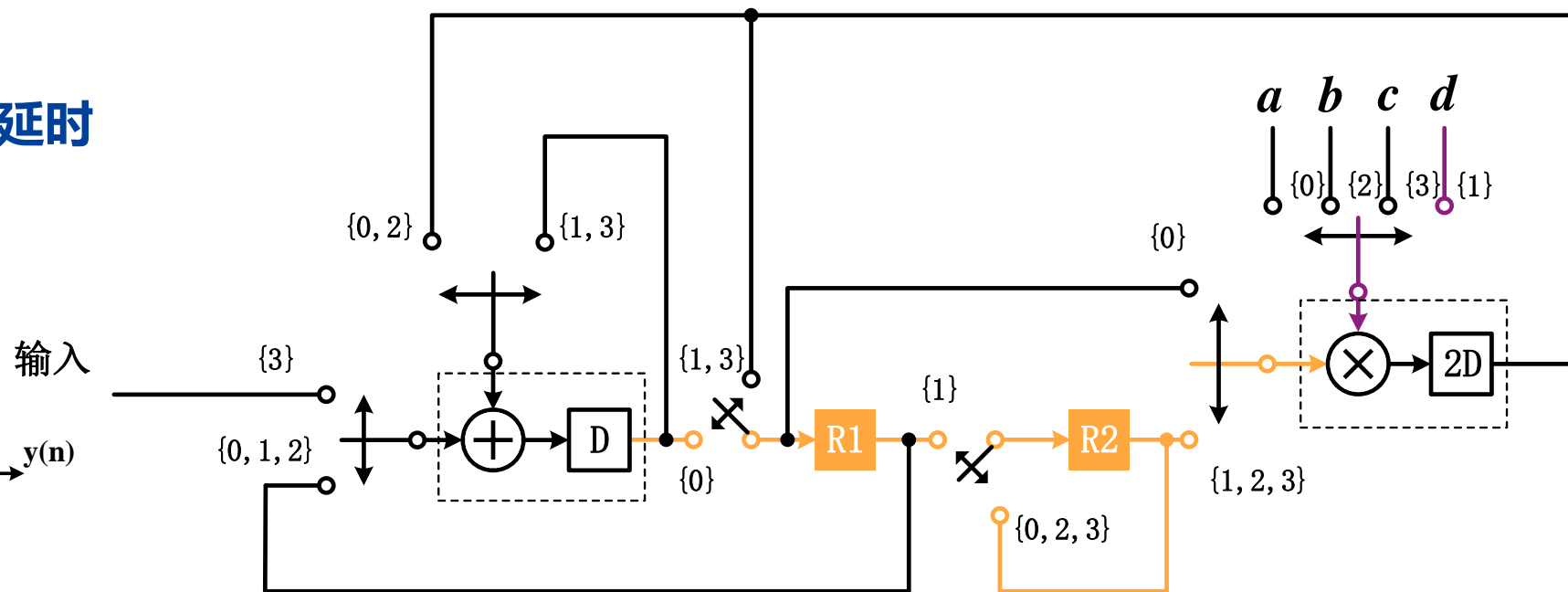
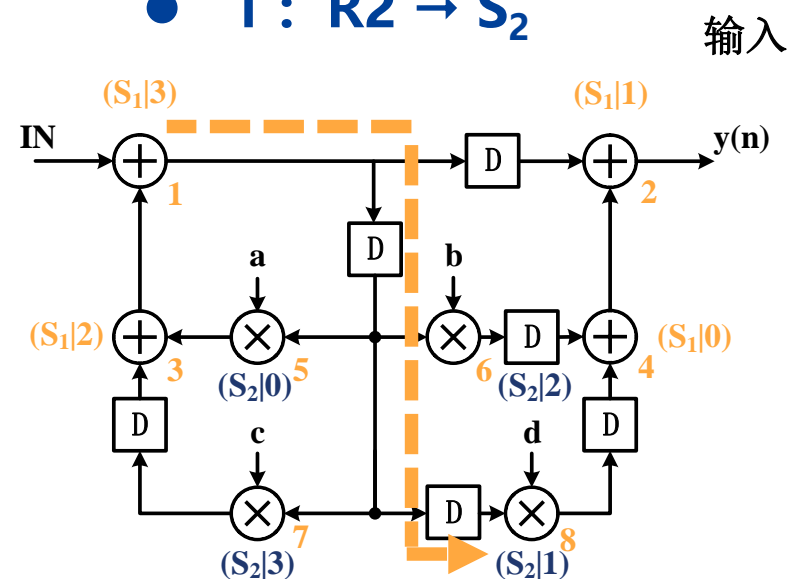


cycle	In	R1	R2	Out	
7(3)			N_1		$n_1 \rightarrow n_7$
8(0)			N_1		
9(1)			N_1	<u>N_1</u>	$n_1 \rightarrow n_8$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察节点8 ($S_2|1$) 输入:

- 系数: d
- 节点1: $D_F(1 \rightarrow 8) = 5$
 - 比 $D_F(1 \rightarrow 7)$ 多两个延时
 - 3: $R2 \rightarrow R2$
 - 0: $R2 \rightarrow R2$
 - 1: $R2 \rightarrow S_2$

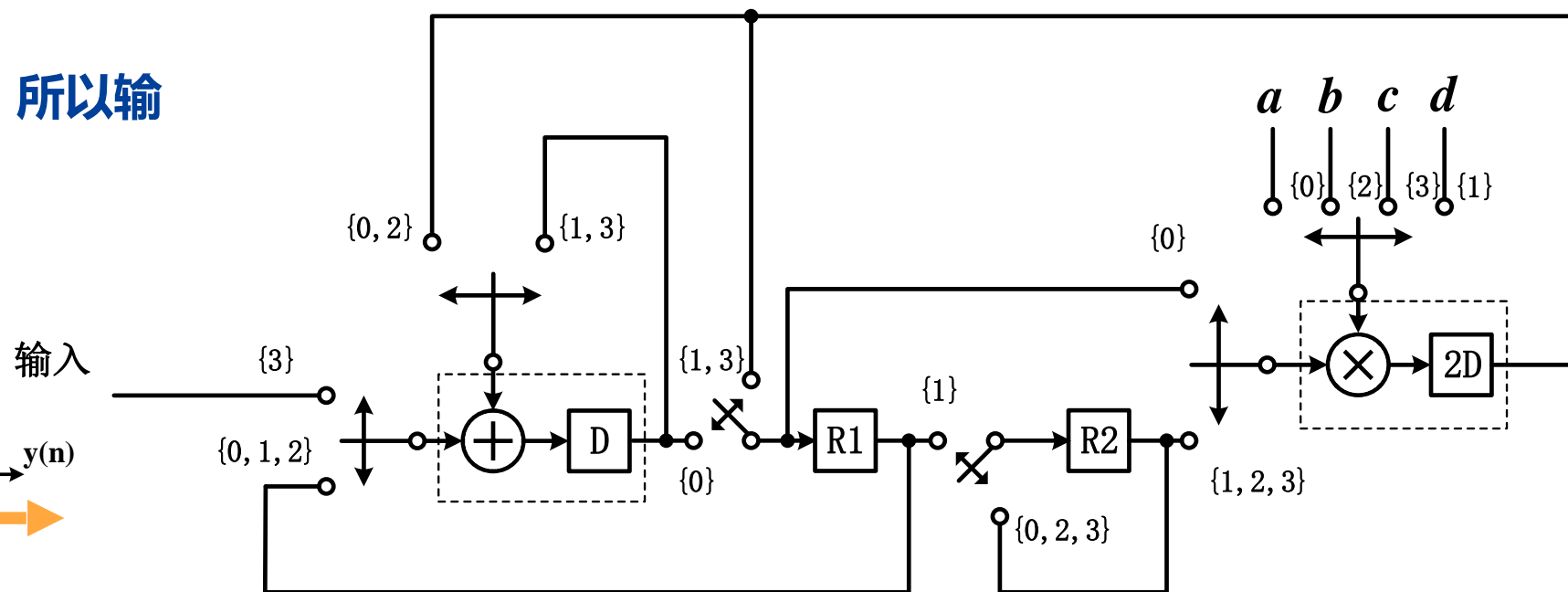
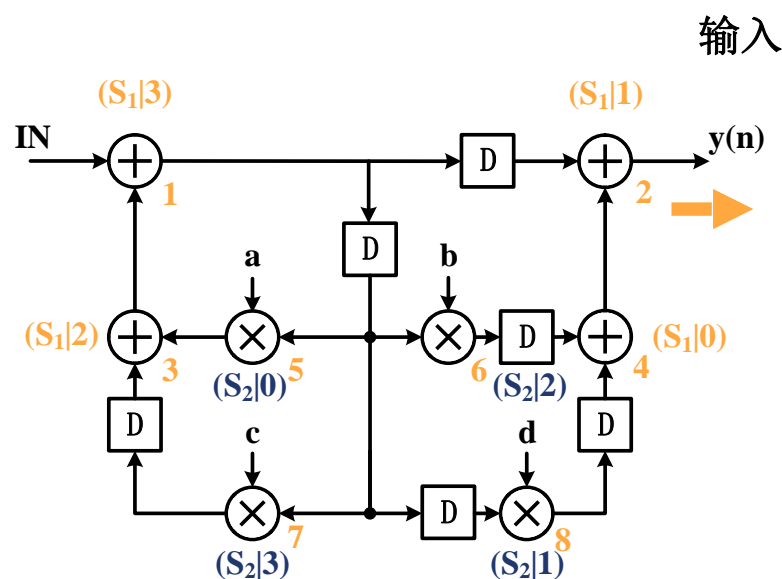


cycle	In	R1	R2	Out	
7(3)			N_1		$n_1 \rightarrow n_7$
8(0)			N_1		
9(1)			N_1	N_1	$n_1 \rightarrow n_8$

04. 折叠架构的寄存器最小化——Biquad滤波器

考察输出端的输入：

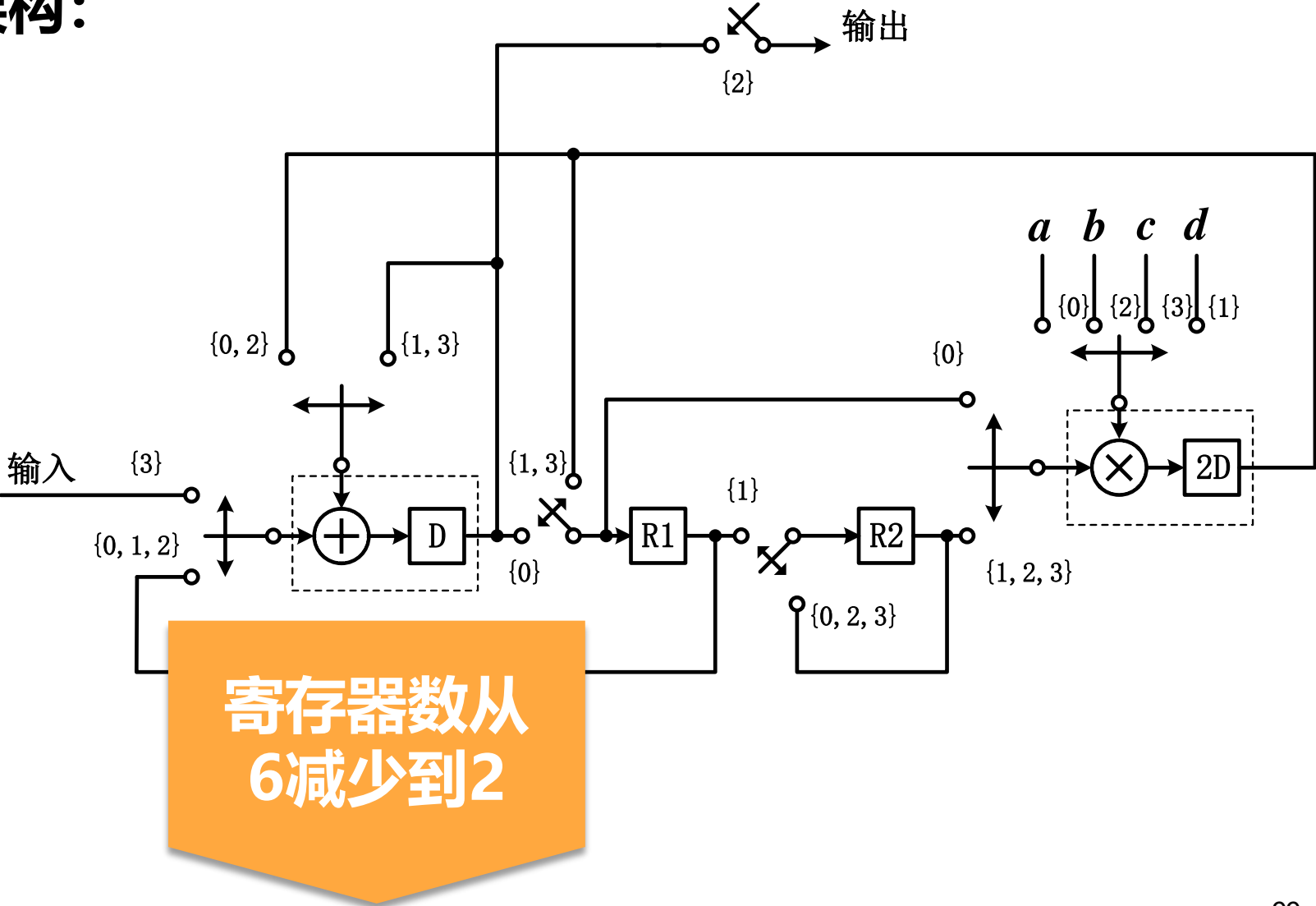
- 节点2 ($S_1|1$) 直接输入到输出端
- 因为内部有一级流水线，所以输出的序号为2



04. 折叠架构的寄存器最小化——Biquad滤波器

最小寄存器数的折叠架构:

cycle	In	R1	R2	Out	
0					
1					
2					
3	N_8				
4(0)	N_1	N_8		N_8	$\rightarrow n_5$
5(1)	N_7	N_1			$n_1 \rightarrow n_2$
6(2)		N_7	N_1	N_7	$n_1 \rightarrow n_6$
7(3)			N_1		$n_1 \rightarrow n_7$
8(0)			N_1		
9(1)			N_1	N_1	$\rightarrow n_8$





目录

01 折叠的基本概念

02 折叠变换

03 寄存器最小化技术

04 折叠架构的寄存器最小化

05 本章总结



■ 折叠是展开的逆过程，能够通过时分复用达到资源共享，从而节省面积和功耗；但是处理时间随之增加

■ 折叠方程： $D_F(U \rightarrow V) = N[l + w(e)] + v - (Nl + u + P_u) \Rightarrow D_F(U \rightarrow V) = Nw(e) - P_u + v - u$

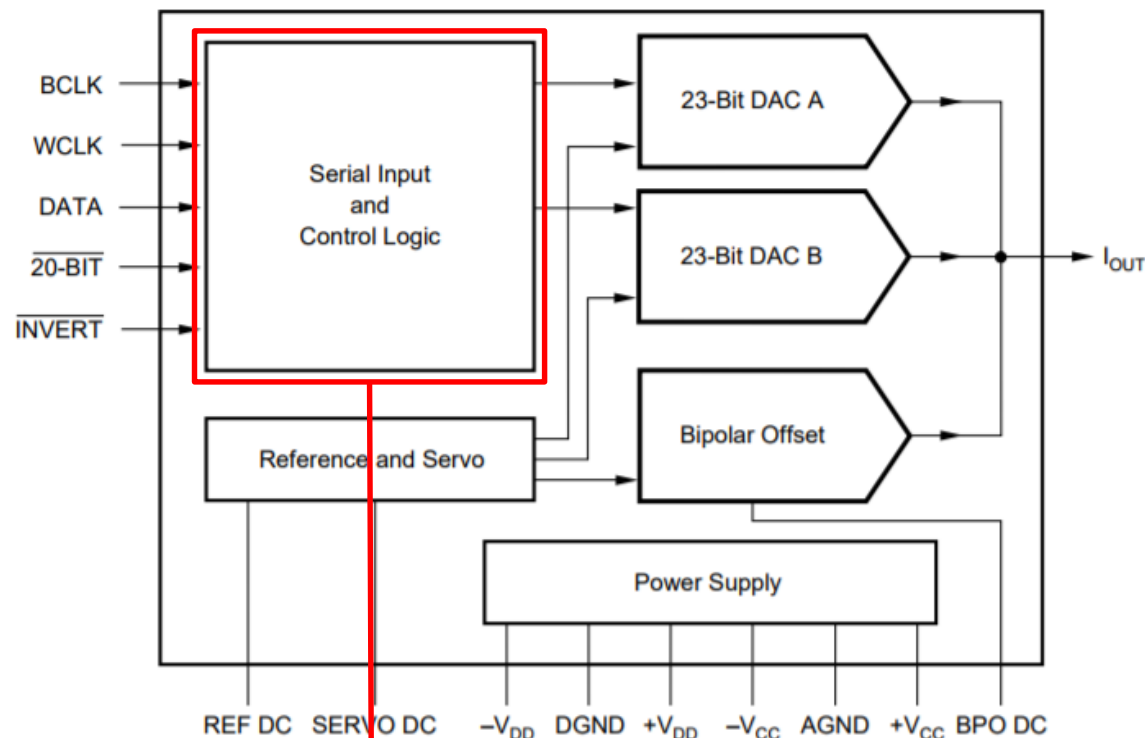
■ 折叠变换流程：

- 确定折叠集
- 列出DFG的折叠方程组
- 可行性约束： $DF(U \rightarrow V) \geq 0$ ，若不可行：改变折叠集；采用重定时等技术
- 画出折叠DFG
 - 画折叠DFG的功能单元、寄存器和多路开关
 - 按折叠方程组连接各边、标注开关的折叠序号

折叠的应用

CD机芯片

音频信号采样频率为44.1kHz,
对芯片处理速度要求不高



索尼 SCD 777ES
PCM1704解码芯片结构

数字处理电路

采用折叠技术

谢谢!

