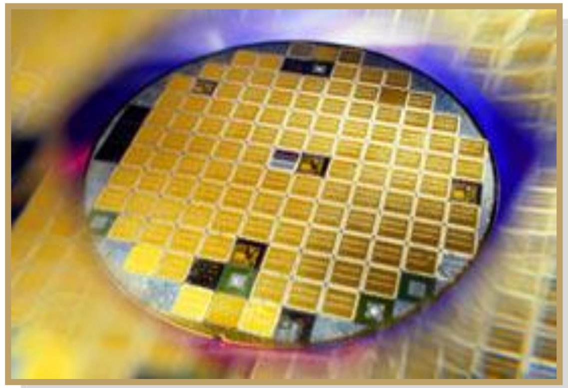
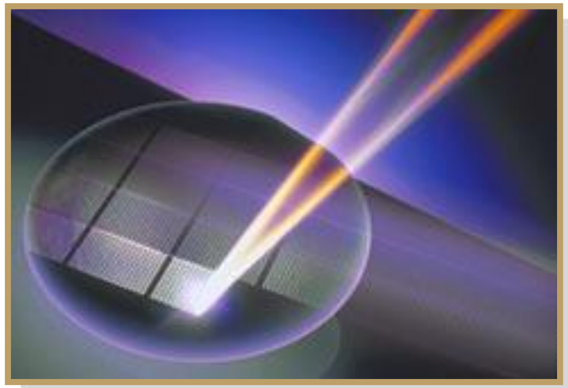
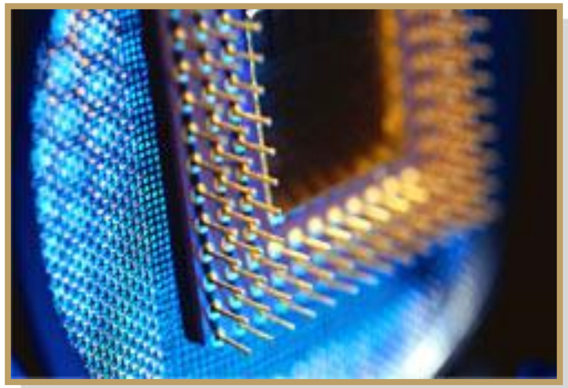
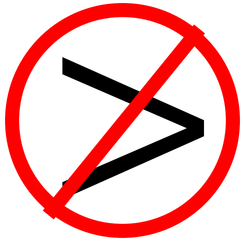
《VLSI数字通信原理与设计》课程 主讲人 贺光辉

第二章：迭代边界

引言

极限速度



电路中的极限速度?

—包含环路系统的极限速度

迭代边界

目录

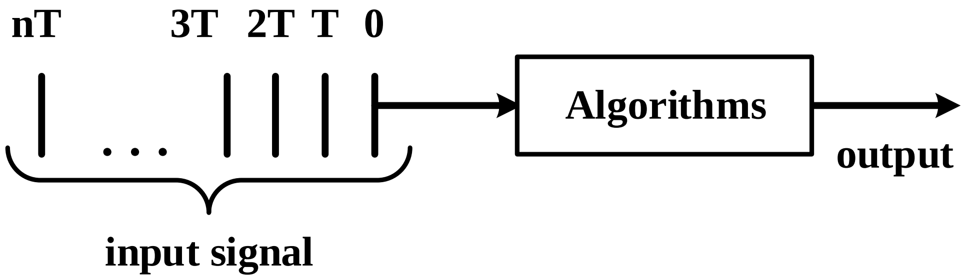
DSP算法

- **FIR-filter：** $y(n)=\sum_{k=0}^{N-1} h(k)x(n-k)$

DSP算法无终止，重复执行同样的代码

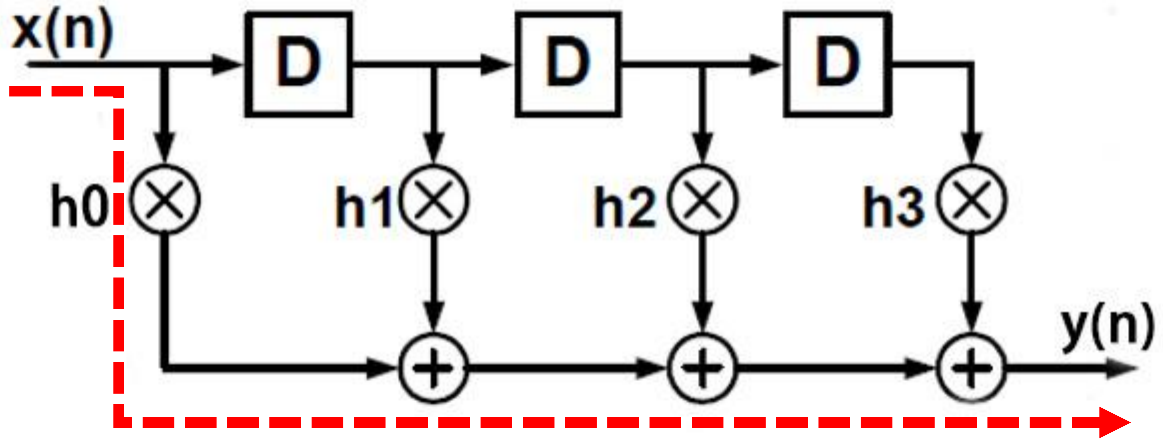
$$for\ n=0\ to\ \infty$$

$$y(n)=h_0x(n)+h_1x(n-1)+h_2x(n-2)+h_3x(n-3)$$



- **迭代 ITERATION：** One execution of the loop is one iteration
- **迭代周期 ITERATION PERIOD：** time for one iteration

直接形式4-tap FIR



One Iteration

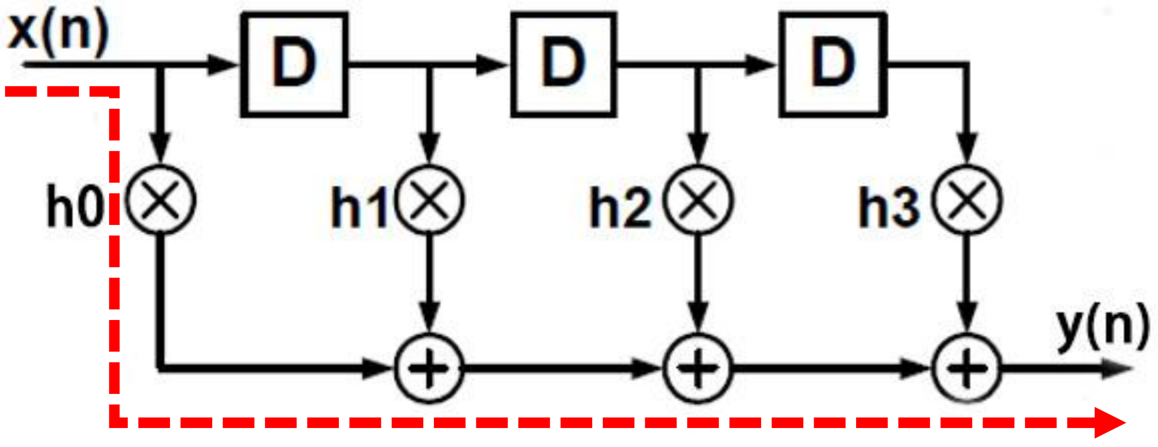
- 1 input
- 4 multipliers
- 3 adders
- 1 output

Critical path：

- longest path without delay element
- Critical path sets bounds on clock frequency

$$T_{clk} \geq T_{critical}$$

直接形式4-tap FIR



Clock speed limited by Critical Path!

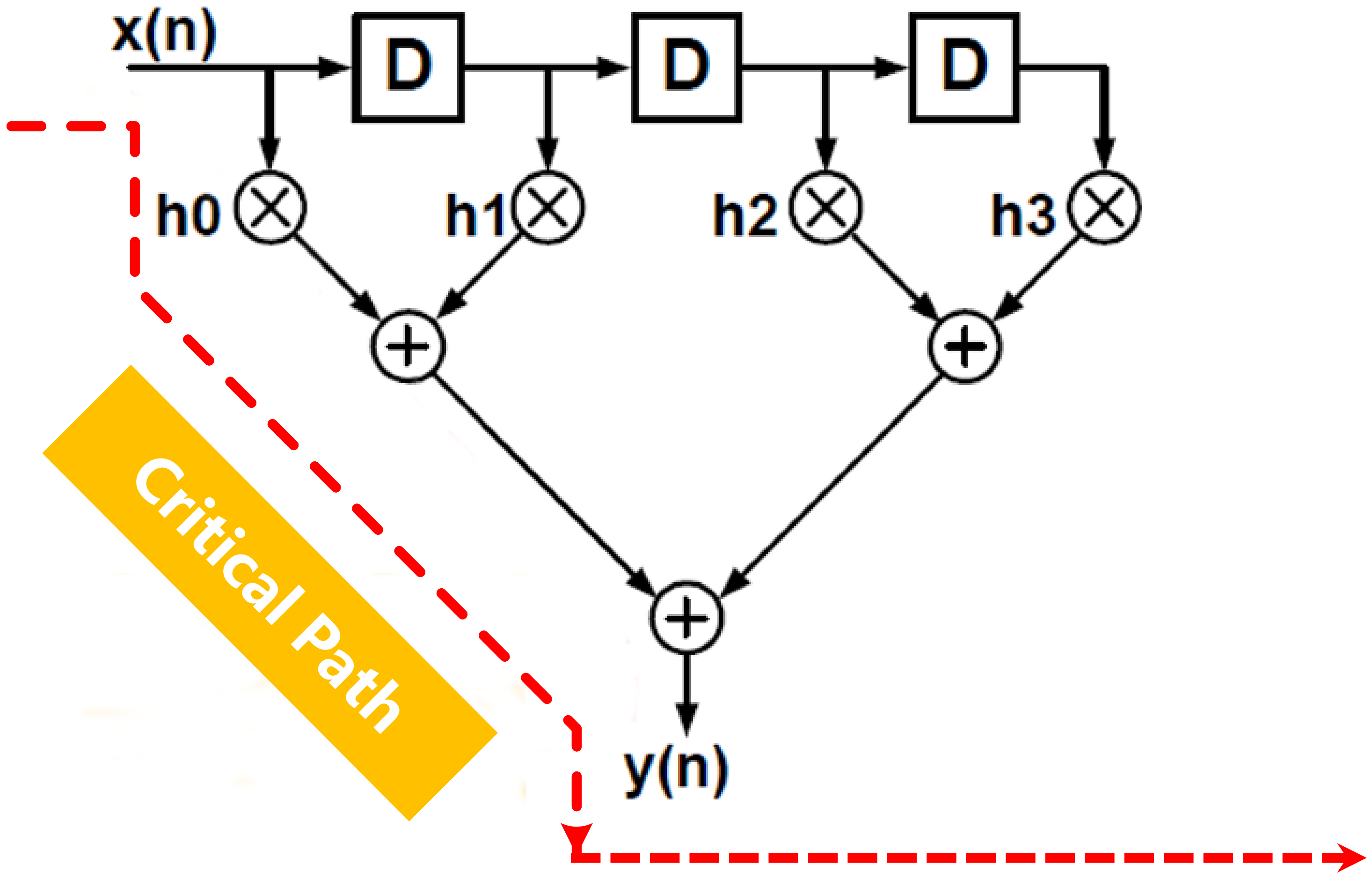
$$T_{critical} = T_M + (N-1)T_A$$

N = Nr. of Taps

Critical path：

- longest path without delay element
- Critical path sets bounds on clock frequency

直接形式4-tap FIR with Adder Tree



$$T_{critical} = T_M + (\log_2 N)T_A$$

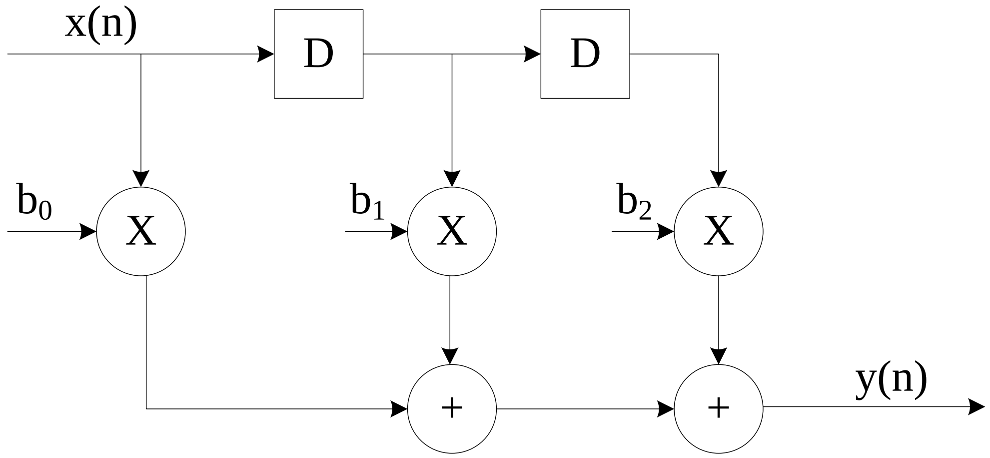
N = Nr. of Taps

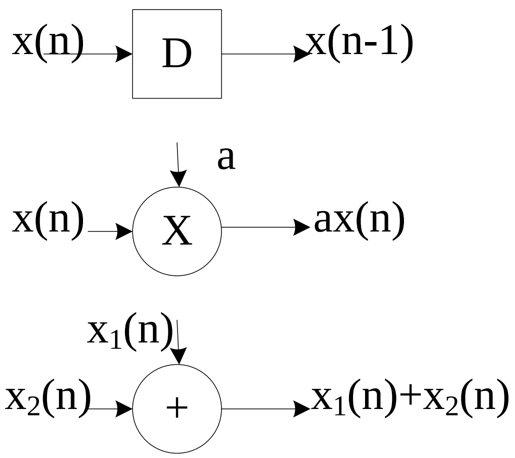
图形表示方法1——框图

框图：

常用于图形化地描述DSP系统，由功能块和有向边（表示从输入到输出的数据流动，含≥0个延时元件）组成，可在不同抽象层次上构建

示例：

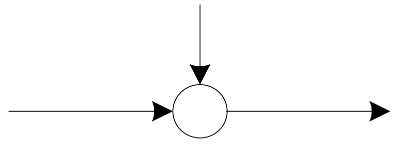




图形表示方法2——信号流图

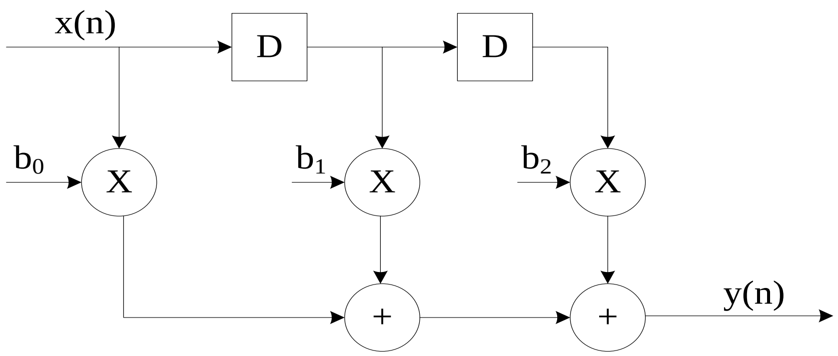
信号流图：

常信号流图是一组节点和有向边的集合，用于分析、表示、评估线性数字网络结构

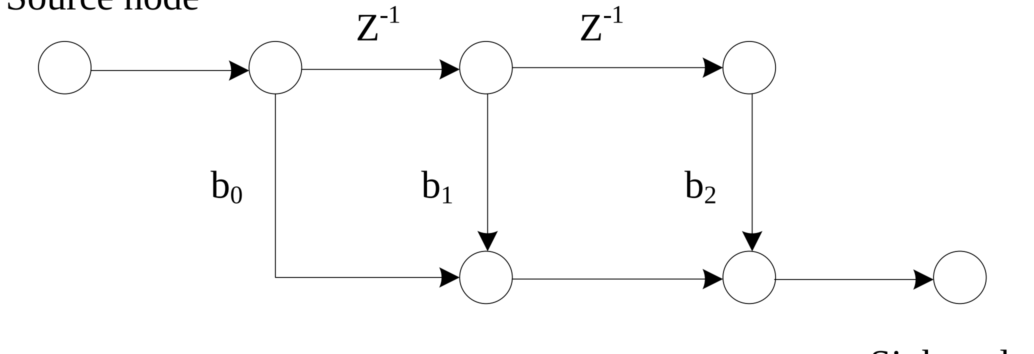
	**The nodes represent computations or tasks**
	**Source node and sink node**
$\xrightarrow{b_0}$ $\xrightarrow{Z^{-1}}$	**The edges are usually restricted to constant gain multipliers or delay elements**

图形表示方法2——信号流图

示例： $y(n)=b_0x(n)+b_1x(n-1)+b_2x(n-2)$



框图



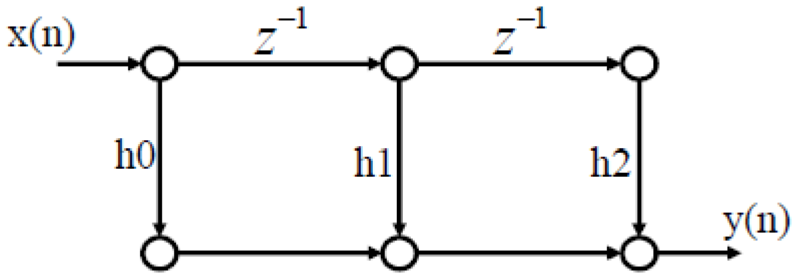
信号流图

图形表示方法2——信号流图

变换：Transposition

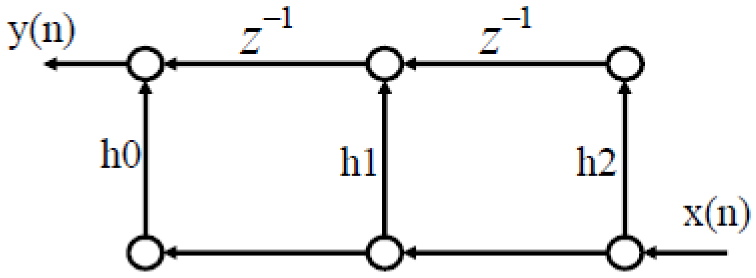
Linear SFGs can be transformed without changing the system functions. For example, Flow graph reversal or transposition。

—Note：only applicable to single-input-single-output systems



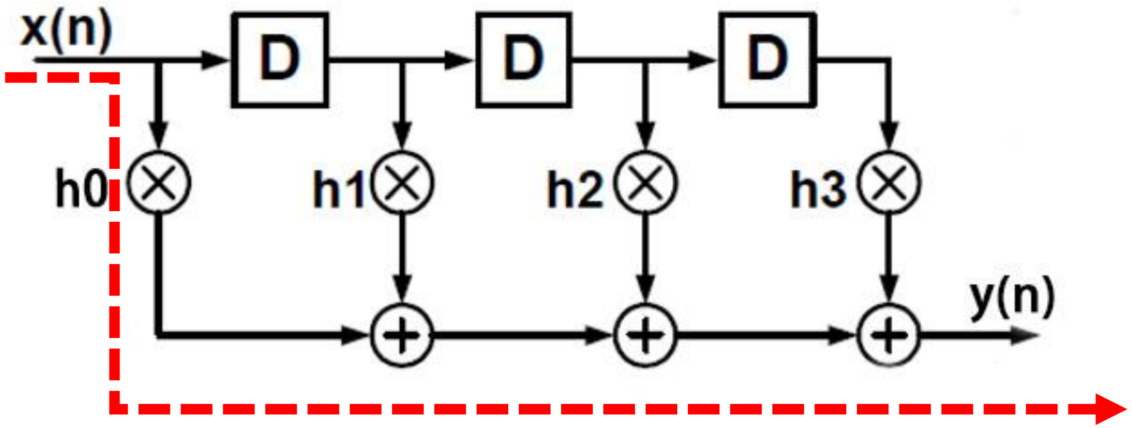
1: Reverse the direction of all edges

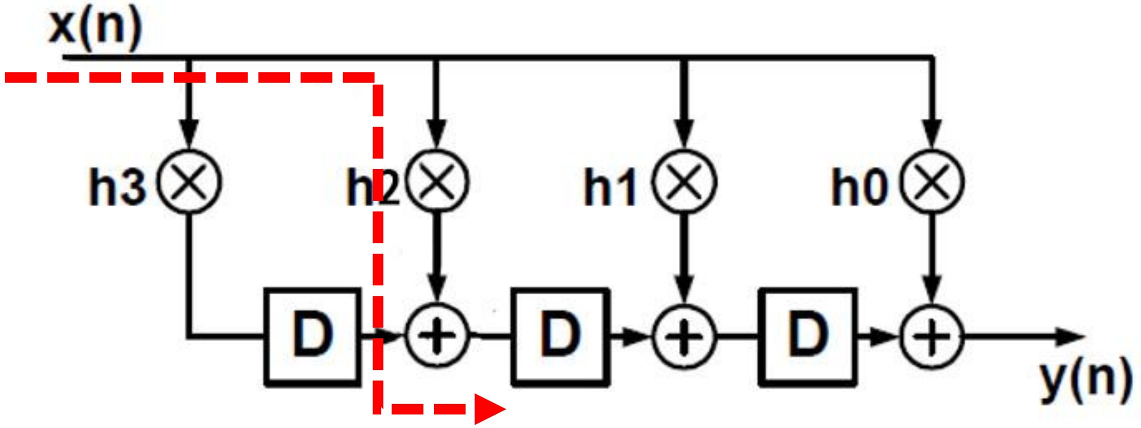
2:Exchange input and output



Transposed Form 4-tap FIR

Direct Form 4-tap FIR → **Transposed Form 4-tap FIR**





$T_{critical} = T_M + (N-1)T_A$

N = Nr. of Taps

→

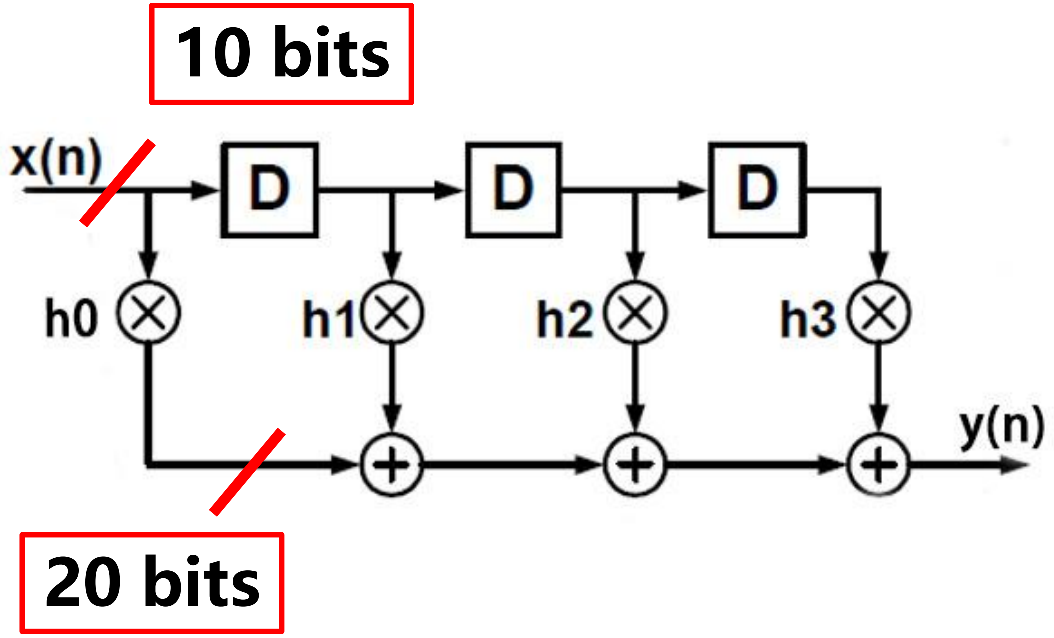
$T_{critical} = T_M + T_A$

N = Nr. of Taps

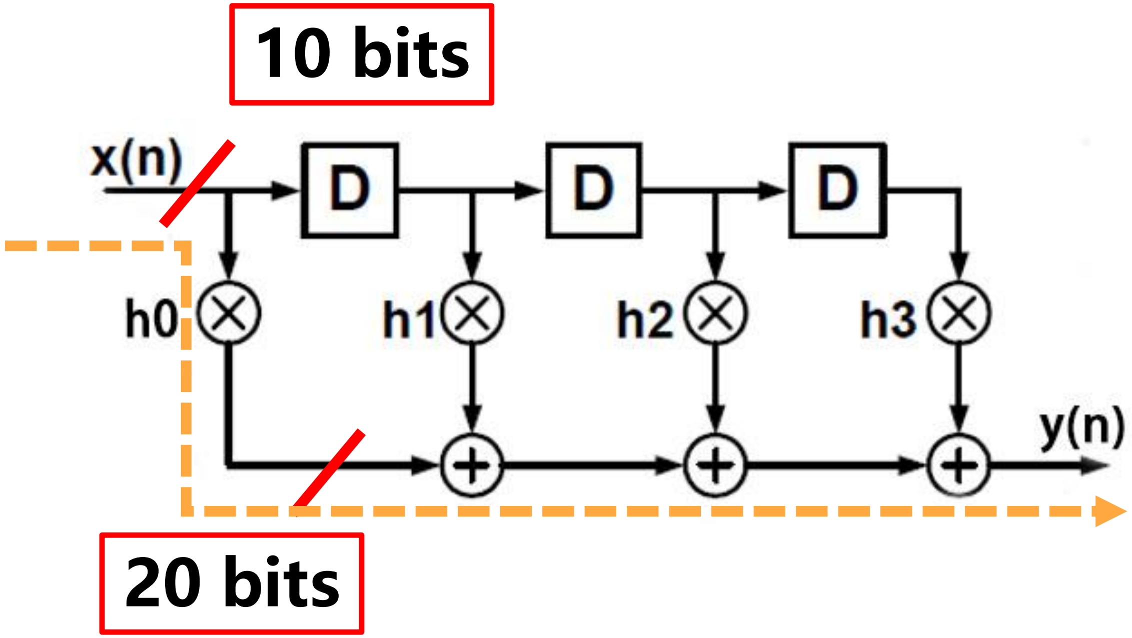
定点实现

字长： Wordlength is a crucial parameter for

- Algorithm Performance
- Power Consumption
- Clock Speed
- Area

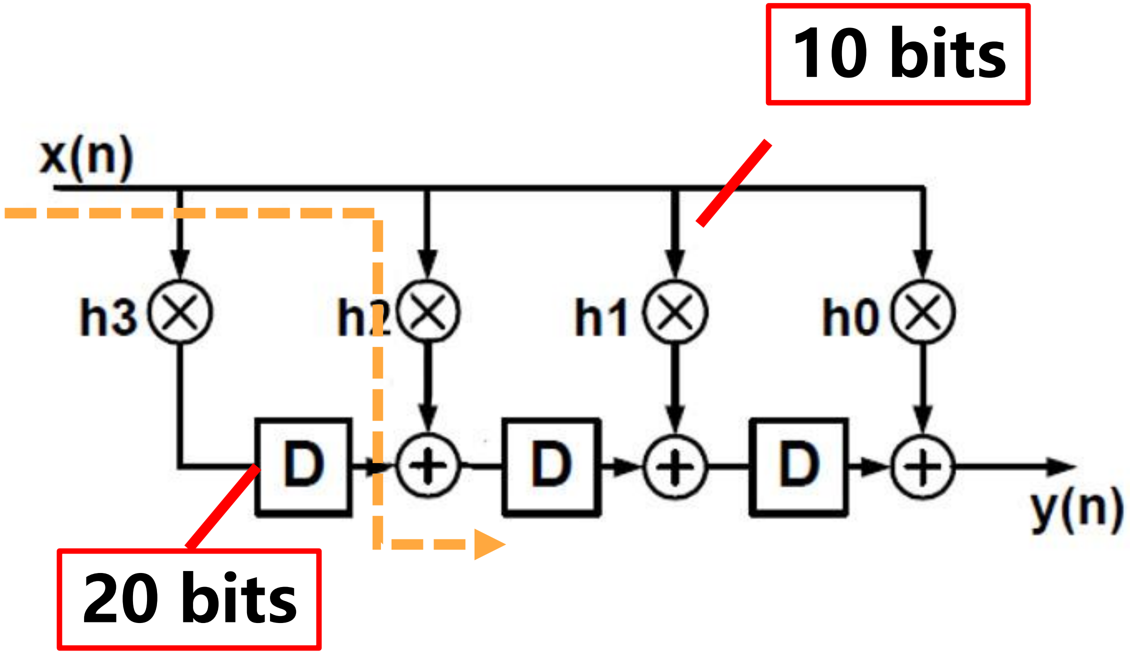


定点实现



Direct Form

Long Critical Path
Narrow Registers



Transposed Form

Short Critical Path
Wide Registers

图形表示方法3——数据流图

DFG(Data-Flow Graph) 只示出一次迭代过程

- **节点**

  -表示算法中计算（或功能）执行
  -包含关联的计算时间：（数字）

- **有向边**

  -表示节点间通信关系
  -包含关联的非负延迟Z^{-1}或D

- **每条边描述了两节点间执行的优先顺序约束**

  -边无延迟：D=0，描述迭代内优先顺序约束（$\rightarrow$）
  -边有延迟：描述迭代间优先顺序约束（$\Rightarrow$）

- **DFG中节点的粒度**

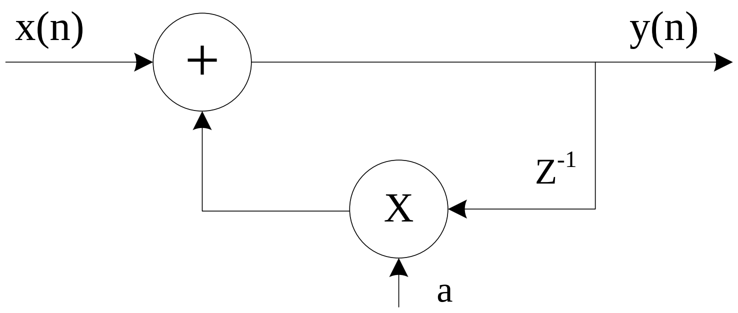
  -细粒度：节点简单到基本运算单元，如乘、加等
  -粗粒度：节点为子任务以上层次的复杂功能块，如滤波、FFT等

图形表示方法3——数据流图

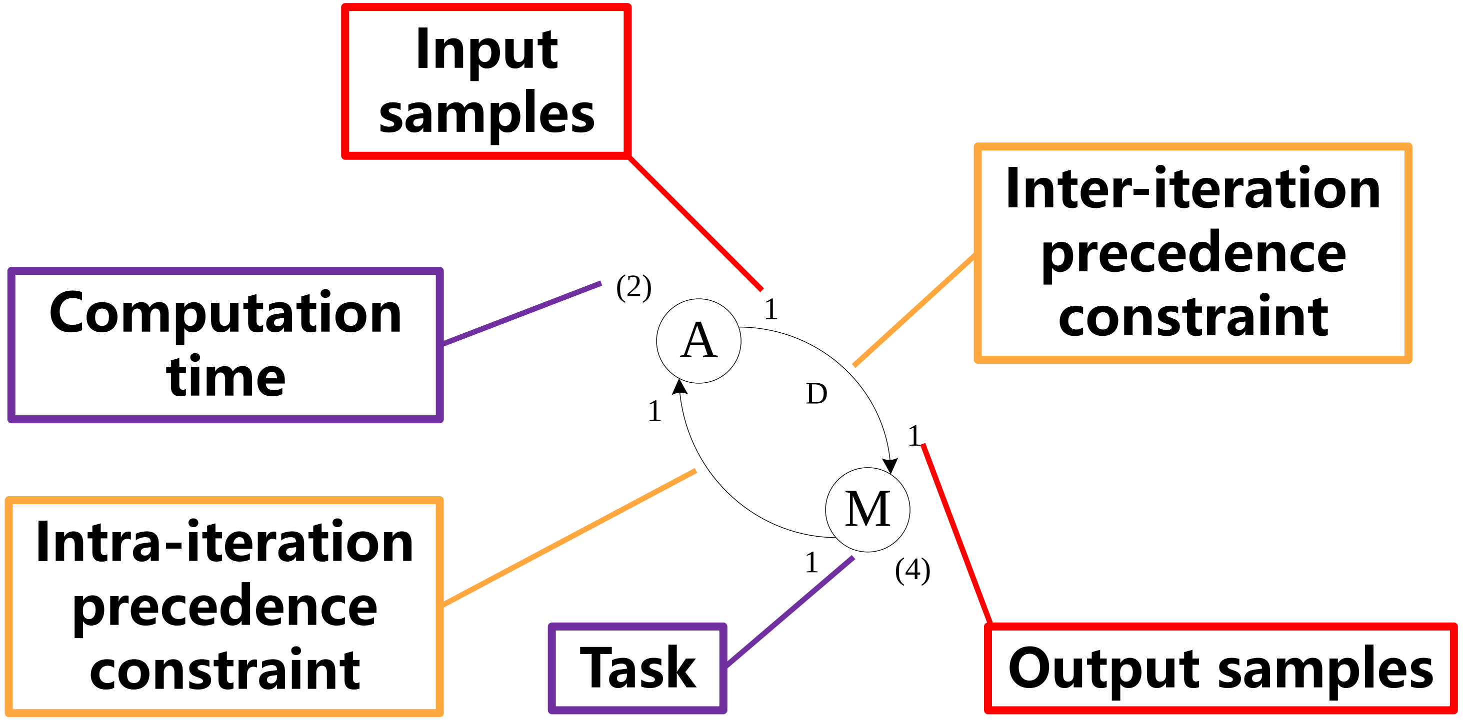
数据流图DFG：

捕获DSP算法的数据驱动性质，一旦所有输入数据准备好后节点就可启动执行

示例： $y(n)=ay(n-1)+x(n)$



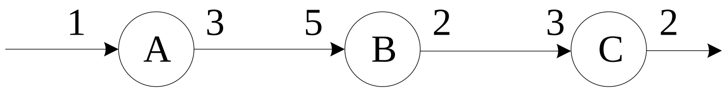
Block diagram description



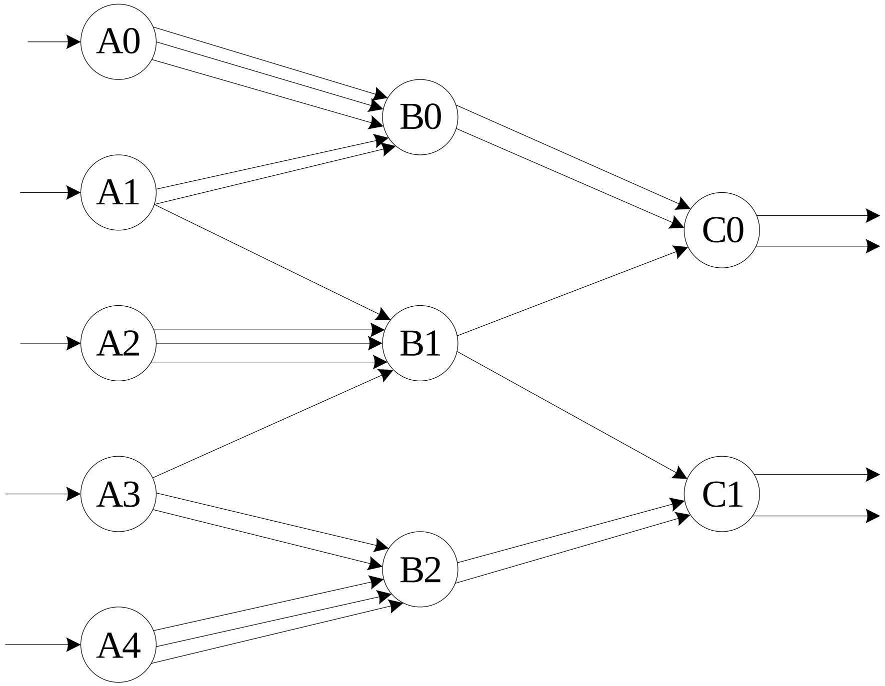
Synchronous DFG description

图形表示方法3——数据流图

Single-rate & Multi-rate



Multi-rate DFG



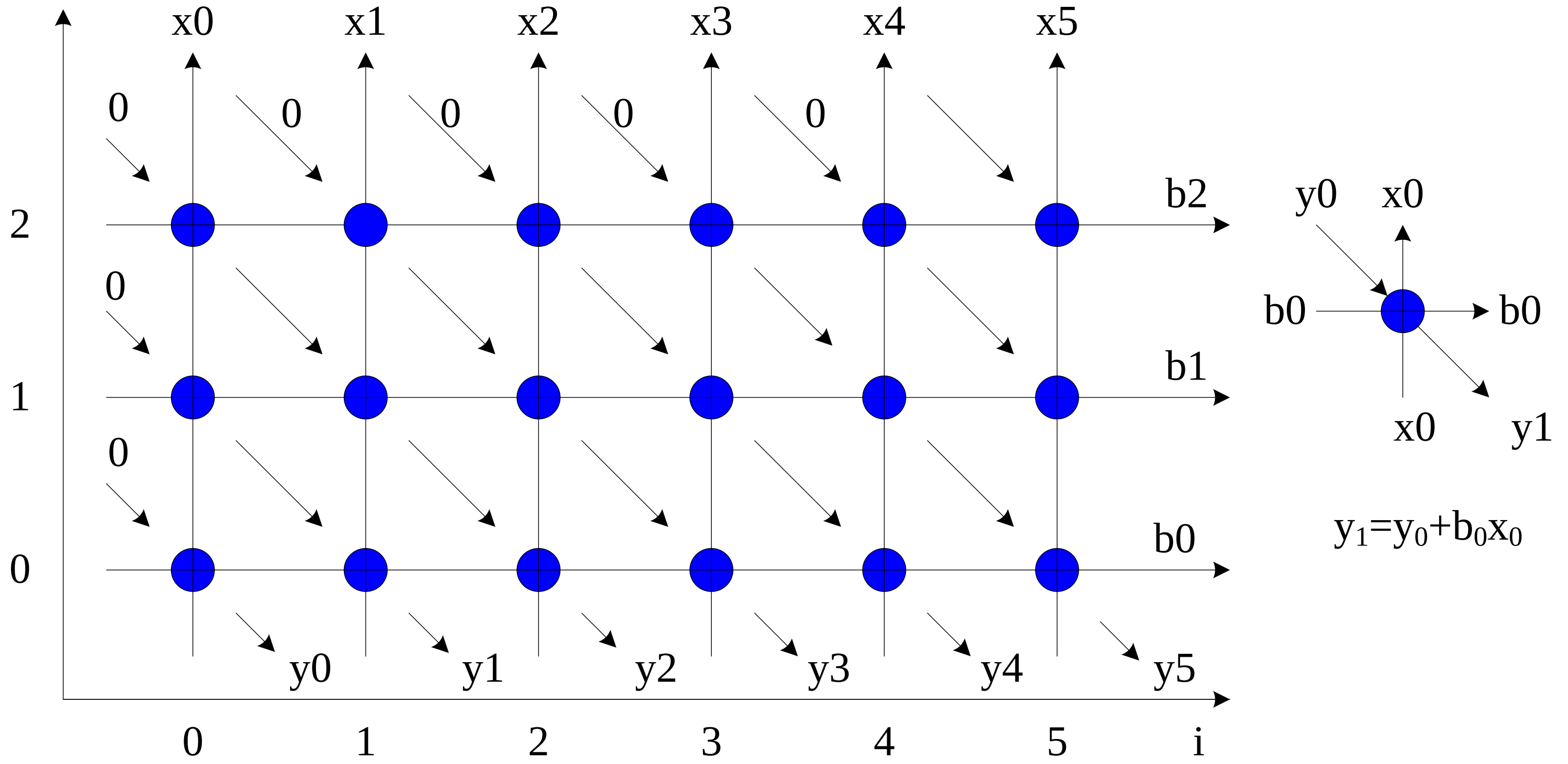
Single-rate DFG

图形表示方法4——依赖图

- **DG（Dependence Graph）：**

依赖图是一种有向图，表示算法计算间的依赖关系（脉动阵列用）

- **示例：** $y(n)=b_0x(n)+b_1x(n-1)+b_2x(n-2)$



目录

路径与关键路径

路径与关键路径：

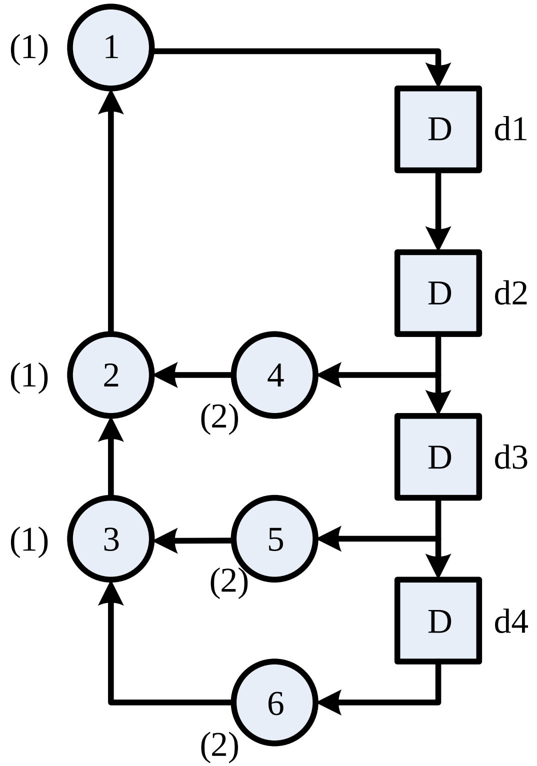
- **路径：** 数据在任意两节点间经有向边和中间节点的通路
- **关键路径：** DFG中在不包含延迟单元的路径中执行计算时间最长的路径T_C

示例：

3条无延迟路径

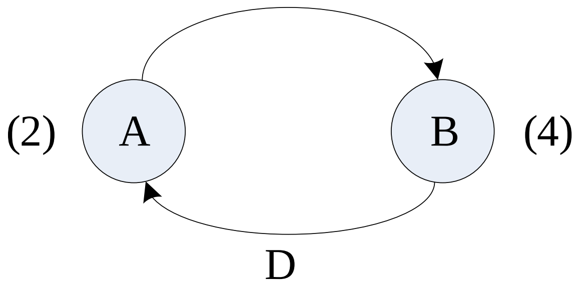
- $4 \rightarrow 2 \rightarrow 1$, $4.u.t.$
- $5 \rightarrow 3 \rightarrow 2 \rightarrow 1$, $5u.t.$
- $6 \rightarrow 3 \rightarrow 2 \rightarrow 1$, $5u.t.$

关键路径 T_C=5u.t.



基本概念

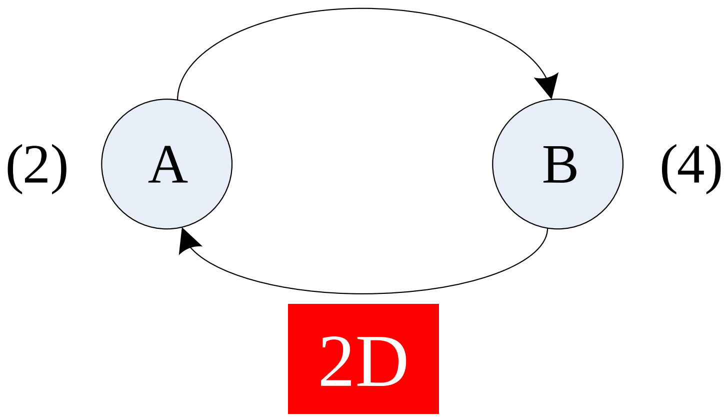
- **迭代：DFG（数据流图）中所有节点执行一次**
- **迭代周期 T_{it}：是处理一个输入样点并输出一个结果所需时间**
- **时钟周期 T_{clock}：系统按拍工作的周期，由关键路径 T_C 决定。**
  - 系统时钟频率 f 则为 T_{clock} 的倒数
- **环路：开始与结束于同一节点的有向路径**
  - $A_0 \rightarrow B_0 \Rightarrow A_1 \rightarrow B_1 \Rightarrow A_2 \rightarrow B_2$。。。下标表示迭代编号
  - 一次迭代时间下限为6u.t.。决定环路每次迭代最低执行时间的因素之一



环路边界

定义:

第 L 个环路的环路边界 $T_{LoopBond}$ 是指 T_L / w_L ，其中 T_L 是环路运行时间， w_L 是环路中延迟数



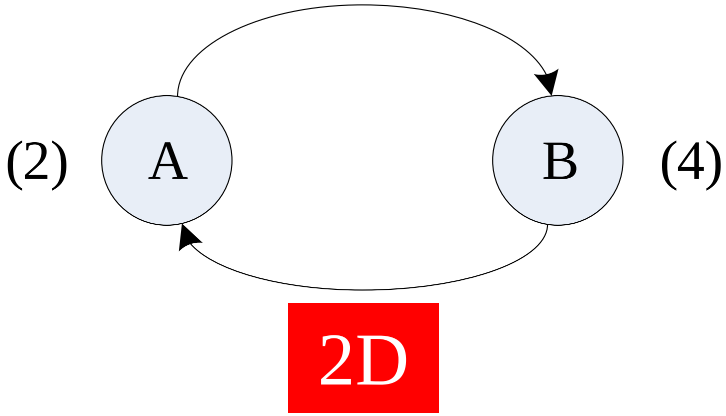
环路边界

$$T_{LoopBond} = (2+4)/1 = 6$$

环路边界

$$T_{LoopBond} = (2+4)/2 = 3$$

环路边界



环路边界

$$T_{LoopBond} = (2+4)/2 = 3$$

系统能够设置两套硬件并行

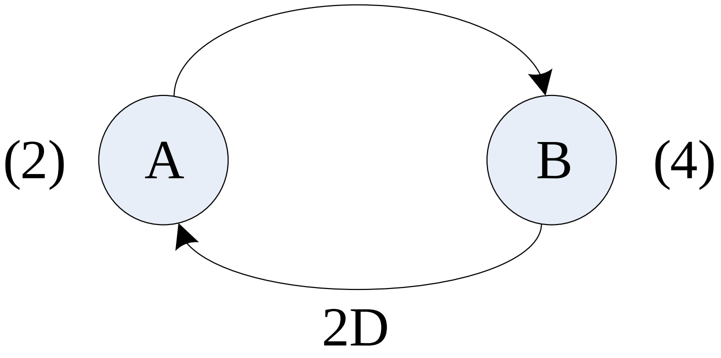
为什么要除以延迟数?

存在两组独立的优先顺序约束，

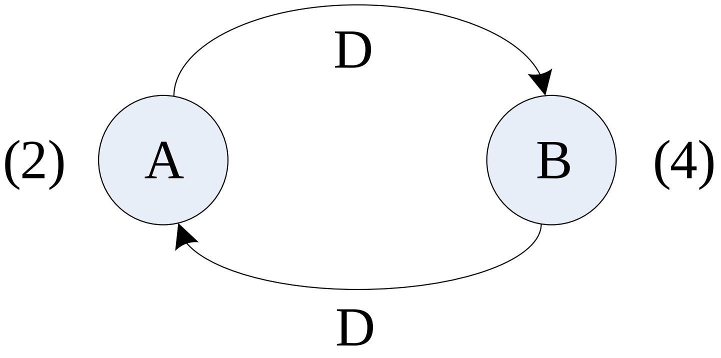
一组偶迭代， $A_0 \rightarrow B_0 \Rightarrow A_2 \rightarrow B_2 \Rightarrow A_4 \rightarrow B_4 \Rightarrow A_6 \rightarrow \cdots$

一组奇迭代， $A_1 \rightarrow B_1 \Rightarrow A_3 \rightarrow B_3 \Rightarrow A_5 \rightarrow B_5 \Rightarrow A_7 \rightarrow \cdots$

环路边界



Critical Path = 6



Critical Path = 4

两者环路边界相同，存在两组独立的优先顺序约束，每组A与B迭代编号交错

$$A_0 \Rightarrow B_1 \Rightarrow A_2 \Rightarrow B_3 \Rightarrow A_4 \Rightarrow B_5 \cdots$$

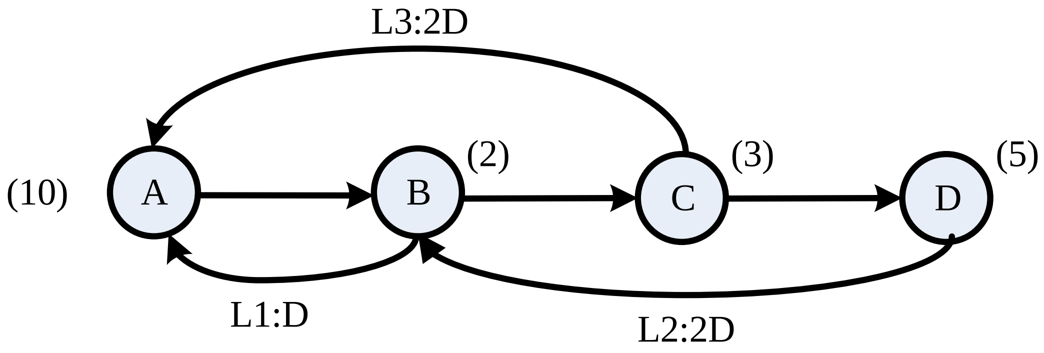
$$A_1 \Rightarrow B_2 \Rightarrow A_3 \Rightarrow B_4 \Rightarrow A_5 \Rightarrow B_6 \cdots$$

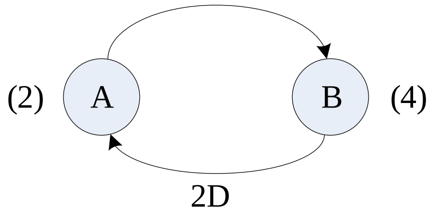
环路边界相同 $T_{LoopBond} = (2+4)/2 = 3$

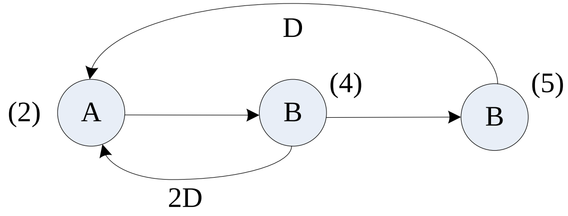
目录

迭代边界

- **迭代边界：关键环路的环路边界 T_∞，** $T_\infty = \max\limits_{l \in L}\left\{\frac{t_l}{w_l}\right\}$ ，**L是DSP系统一组环的集合**
- **关键环路：具有最大环路边界的环路**
- **3个示例：**







$$T_{L1} = (10+2)/1 = 12$$

$$T_{L2} = (2+3+5)/2 = 5$$

$$T_{L3} = (2+3+10)/2 = 7.5$$

$$T_\infty = T_{LoopBond} = 3$$

$$T_\infty = max\{6/2, 11/1\} = 11$$

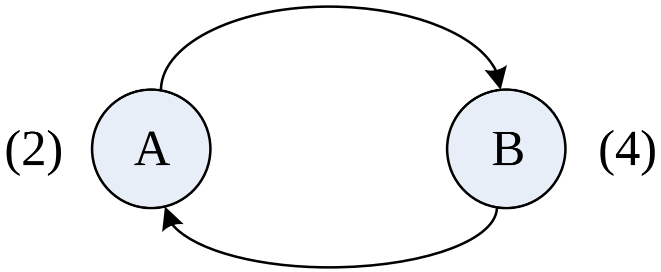
$$T_\infty = \max_{l \in L}\{12, 5, 7.5\} = 12$$

迭代边界

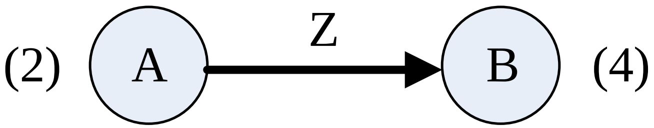
迭代边界的特点（1）：

- **环路必须有延迟元件**

  -迭代若环路延迟数 $w_L = 0$，则 $T_L / 0 = \infty$



- **必须是因果系统，非因果系统无法硬件实现**



$$\left\{ \begin{array}{ll} B = A \cdot Z & \text{非因果} \\ A = B \cdot Z^{-1} & \text{因果} \end{array} \right\}$$

迭代边界

迭代边界的特点（2）：

- 迭代边界：关键环路的环路边界T_∞
- 给出DFG所有环路迭代周期的下限
- 决定带反馈环路DSP算法性能的重要参数，反映了硬件实现DSP程序能有多快
- 即使DSP系统无限提高计算能力，迭代周期≥迭代边界

迭代边界

迭代边界的计算方法(1)：

最长路径矩阵 **longest path matrix algorithm(LPM)**

- 建立一系列矩阵，利用矩阵对角线元素求迭代边界
- 矩阵$\mathrm{L}^{(m)}, m = 1, 2, \cdots, d$，按照下面的方法构建：
  - 从延时单元d_i到d_j中经过正好m-1个延时（不包含d_i和d_j）的所有路径中，用$l_{i,j}^{(m)}$表示最长的运算时间。
  - 如果这样的路径不存在，那么$l_{i,j}^{(m)} = -1$

$$L^{(m)} = \begin{bmatrix} l_{11}^{(m)} & l_{12}^{(m)} & \cdots & l_{1d}^{(m)} \\ l_{21}^{(m)} & l_{22}^{(m)} & \cdots & l_{2d}^{(m)} \\ & & \cdots & \\ l_{d1}^{(m)} & l_{d2}^{(m)} & \cdots & l_{dd}^{(m)} \end{bmatrix}$$

Longest Computation time

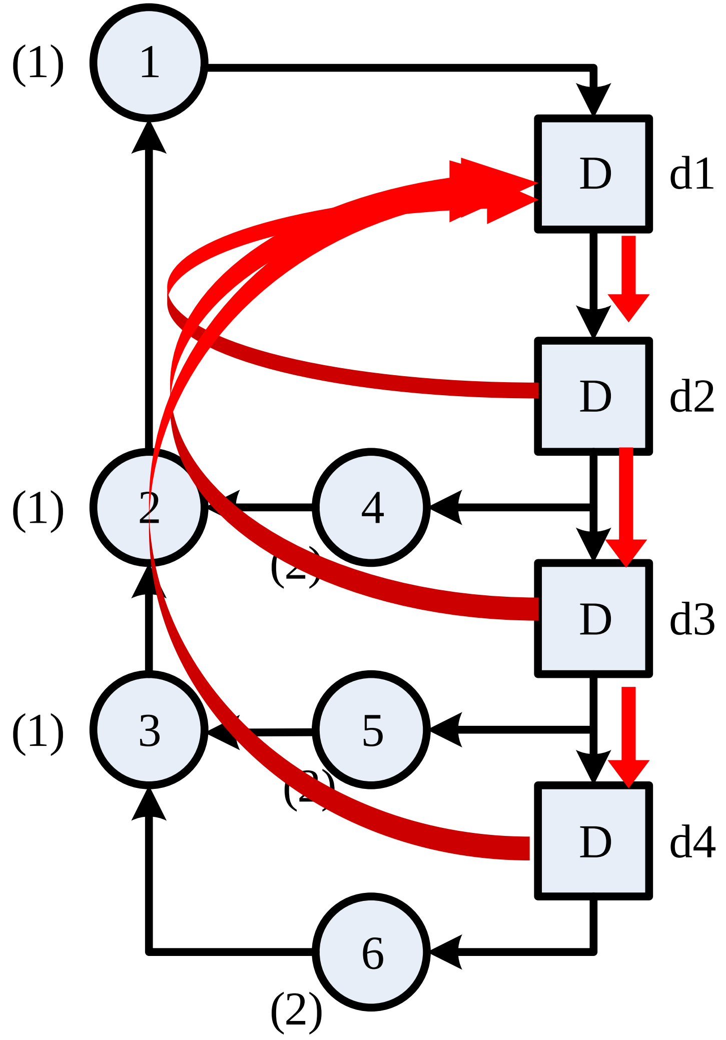
From i to j

Delay=m-1

迭代边界

最长路径矩阵：

$$\mathbf{L}^{(1)}=\begin{bmatrix}-1 & \boxed{0} & -1 & -1\\ \boxed{4} & -1 & \boxed{0} & -1\\ \boxed{5} & -1 & -1 & \boxed{0}\\ \boxed{5} & -1 & -1 & -1\end{bmatrix}$$

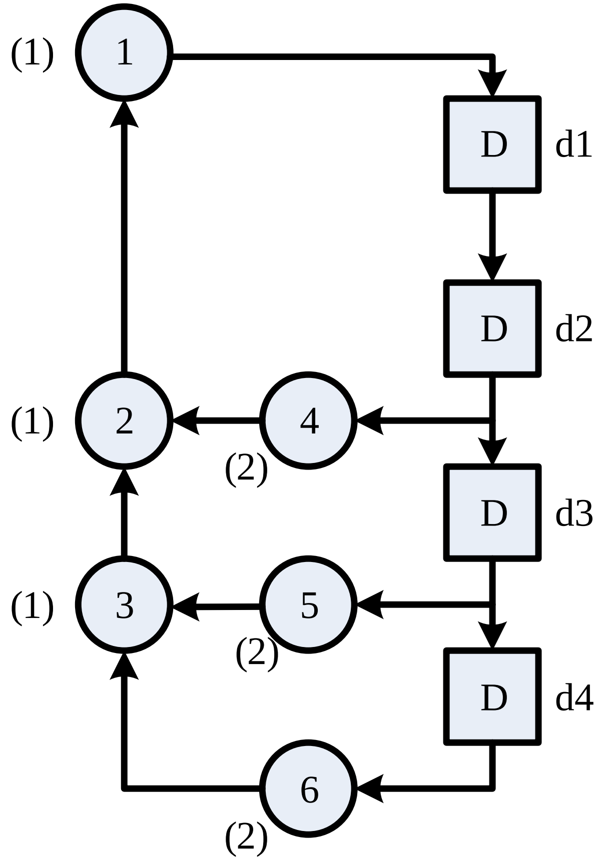


迭代边界

最长路径矩阵： $l_{i,j}^{(m+1)} = \max_{k \in K}(-1, l_{i,k}^{(1)} + l_{k,j}^{(m)})$

$$\mathbf{L}^{(2)} = \begin{bmatrix} -1 & 0 & -1 & -1 \\ 4 & -1 & 0 & -1 \\ 5 & -1 & -1 & 0 \\ 5 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} -1 & 0 & -1 & -1 \\ 4 & -1 & 0 & -1 \\ 5 & -1 & -1 & 0 \\ 5 & -1 & -1 & -1 \end{bmatrix}$$

$$L^{(2)} = \begin{bmatrix} 4 & -1 & 0 & -1 \\ 5 & 4 & -1 & 0 \\ 5 & 5 & -1 & -1 \\ -1 & 5 & -1 & -1 \end{bmatrix}$$



迭代边界

最长路径矩阵：

$$L^{(3)}=\begin{bmatrix}-1 & 0 & -1 & -1\\ 4 & -1 & 0 & -1\\ 5 & -1 & -1 & 0\\ 5 & -1 & -1 & -1\end{bmatrix}\begin{bmatrix}4 & -1 & 0 & -1\\ 5 & 4 & -1 & 0\\ 5 & 5 & -1 & -1\\ -1 & 5 & -1 & -1\end{bmatrix} \Rightarrow L^{(3)}=\begin{bmatrix}5 & 4 & -1 & 0\\ 8 & 5 & 4 & -1\\ 9 & 5 & 5 & -1\\ 9 & -1 & 5 & -1\end{bmatrix}$$

$$L^{(4)}=\begin{bmatrix}-1 & 0 & -1 & -1\\ 4 & -1 & 0 & -1\\ 5 & -1 & -1 & 0\\ 5 & -1 & -1 & -1\end{bmatrix}\begin{bmatrix}5 & 4 & -1 & 0\\ 8 & 5 & 4 & -1\\ 9 & 5 & 5 & -1\\ 9 & -1 & 5 & -1\end{bmatrix} \Rightarrow L^{(4)}=\begin{bmatrix}8 & 5 & 4 & -1\\ 9 & 8 & 5 & 4\\ 10 & 9 & 5 & 5\\ 10 & 9 & -1 & 5\end{bmatrix}$$

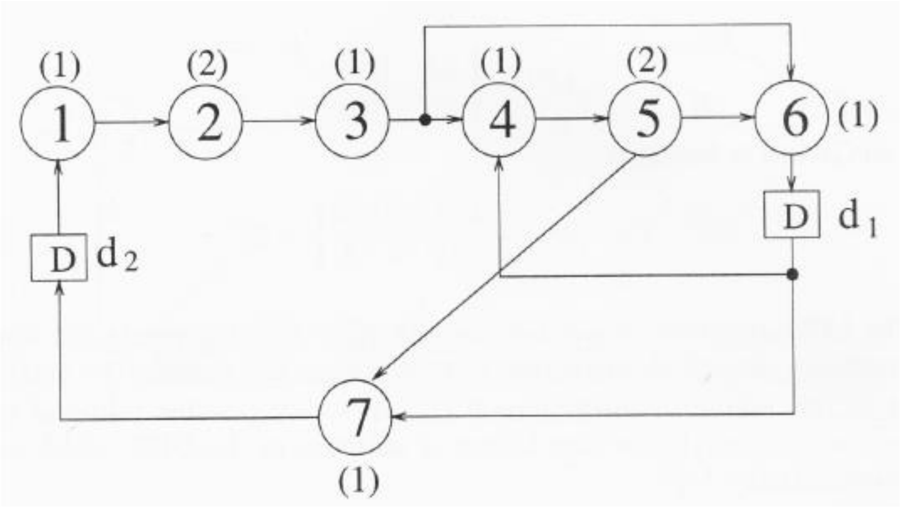
$$T_{\infty}=\max_{i,m\in\{1,2,\ldots,d\}}\left\{\frac{l_{i,i}^{(m)}}{m}\right\}$$

$$T_{\infty}=\max\{\frac{4}{2},\frac{4}{2},\frac{5}{3},\frac{5}{3},\frac{5}{3},\frac{8}{4},\frac{8}{4},\frac{5}{4},\frac{5}{4}\}=2$$

迭代边界

最长路径矩阵 LPM ：

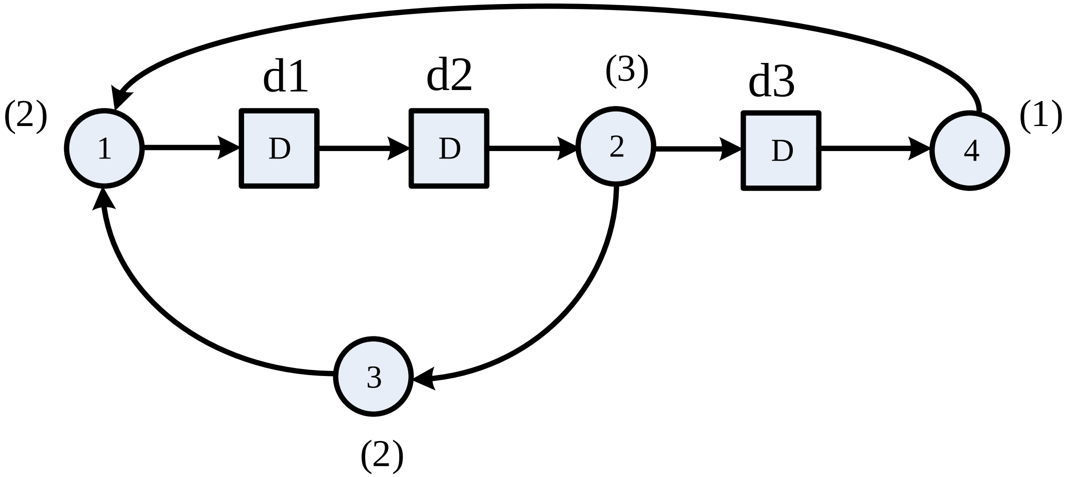
- $l^{(m)}$ 描述了环路 $i \to i$ 具有 m 延迟的计算时间
- $\dfrac{l_{i,i}^{(m)}}{m}$ 为环路边界
- $max(\dfrac{l_{i,i}^{(m)}}{m})$ 为迭代边界



迭代边界

EXERCISE :

- **For the DFG shown, the computation times of nodes are shown in parentheses. Compute the iteration bound of this DFG using the LPM algorithm.**



$$L^{(1)} = \begin{bmatrix} -1 & 0 & -1 \\ 7 & -1 & 3 \\ 3 & -1 & -1 \end{bmatrix}, \quad L^{(2)} = \begin{bmatrix} 7 & -1 & 3 \\ 6 & 7 & -1 \\ -1 & 3 & -1 \end{bmatrix}, \quad L^{(3)} = \begin{bmatrix} 6 & 7 & -1 \\ 14 & 6 & 10 \\ 10 & -1 & 6 \end{bmatrix}$$

$$T_\infty = \max\left\{ 7/2, 6/3 \right\} = 3.5$$

目录

基本概念

采样周期：

- 输入信号样点间隔的时间
- 取决于应用需要：语音、图像等各不相同

迭代周期：

- 完成一次迭代的时间
- 取决于时钟周期和产生输出样点数

时钟周期：

- DSP系统工作所用的时钟周期
- 取决于DSP的关键路径

关键路径：

- DFG中执行计算时间最长的无延迟路径

迭代周期、采样周期、时钟周期之间联系

实时处理：

- 要求迭代周期 = 采样周期

根据情况：

- 流水线：时钟周期=迭代周期，即时钟周期=采样周期
- 并行处理：时钟周期(慢)≠迭代周期，时钟周期≠采样周期
- 折叠：时钟周期(快)≠迭代周期，时钟周期≠采样周期

本章总结

- 描述关键路径与迭代边界的概念
- 能够找出电路中的关键路径
- 运用LPM方法求电路的迭代边界

谢谢！