

# **Introduction of Processor Design for AI Applications**

## **L04 – Iteration Bound**

Pengju Ren

Institute of Artificial Intelligence and Robotics

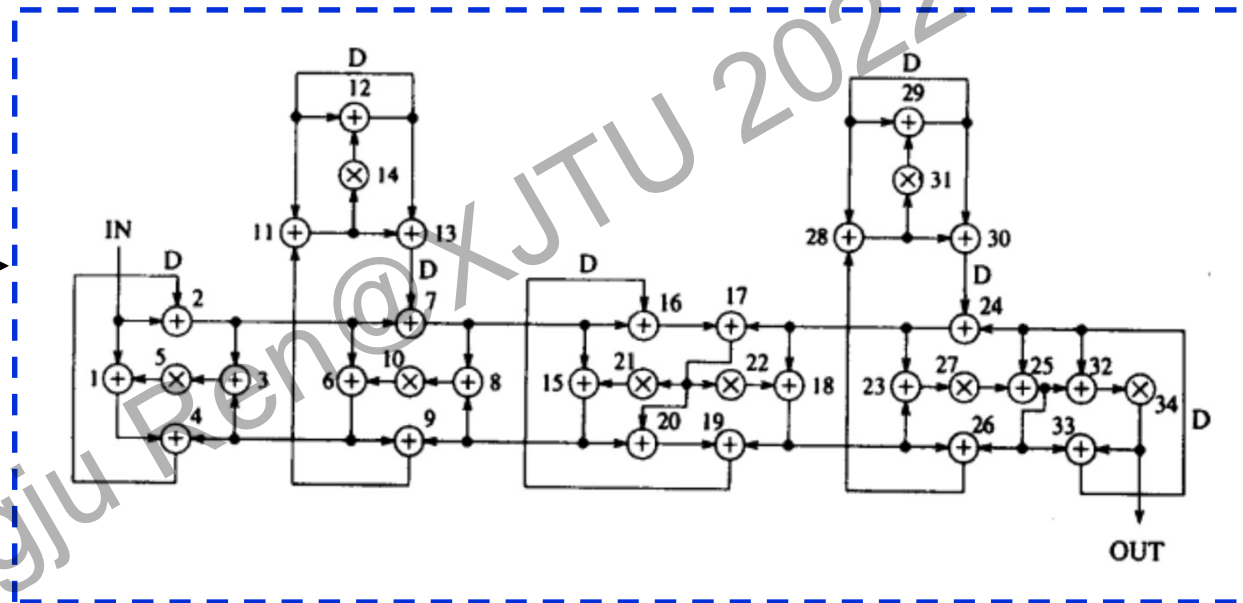
Xi'an Jiaotong University

<http://gr.xjtu.edu.cn/web/pengjuren>

# Why Iteration Bound Matters?

How to determine the feasible Sample rate ( iteration period) ?

Input Signal  $i(n)$



Data Stream Hardware Module

- Only for recursive algorithms which have *feedback loops*
- Impose an inherent fundamental **lower bound** on the achievable iteration or sample period

# Outline

## **Loop Bound and Iteration Bound**

**Important Definitions and Examples**

## **Compute the Iteration Bound**

**Longest Path Matrix Algorithm (LPM)**

**Minimum Cycle Mean Method (MCM)**

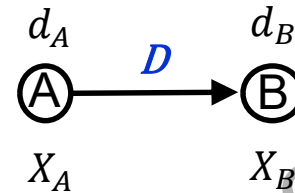
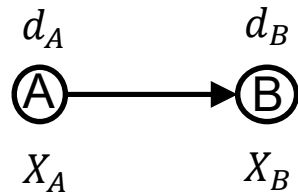
# Introduction

**Iteration period:** the time required for execution of one iteration of algorithm (same as **sample period**).

**Iteration rate:** the number of iterations executed per second

**Sample rate** = the number of samples processed in the system per second (a.k.a **throughput rate**)

# Preliminaries (Time constraints)



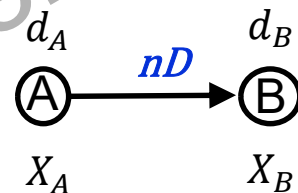
$d_A, d_B$ : Execution delays of A and B

$X_A, X_B$ : Start time of “some” iteration of A and B

$T$ : Iteration period

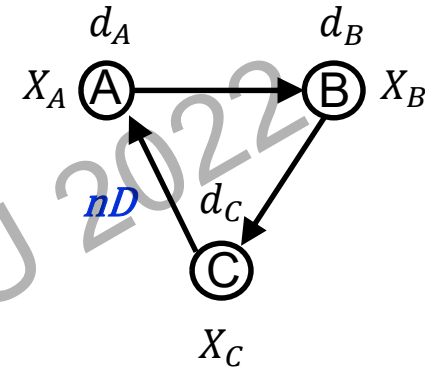
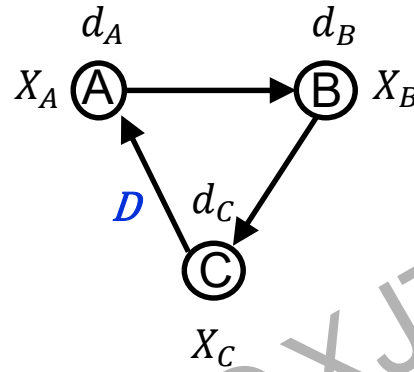
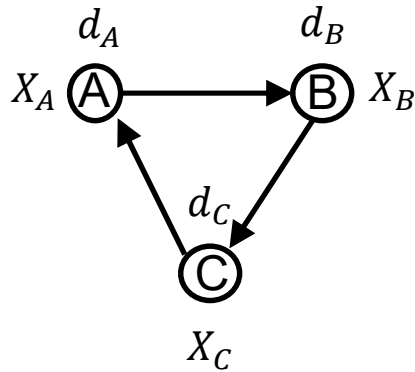
$$X_B \geq X_A + d_A$$

$$X_B \geq X_A + d_A - T$$



$$X_B \geq X_A + d_A - nT$$

# Preliminaries (Loop bound)



$d_A, d_B, d_C$  : Execution delays of A , B and C

$X_A, X_B, X_C$ : Start time of “some” iteration of A , B and C

$T$ : Iteration period

$$X_B \geq X_A + d_A$$

$$X_C \geq X_B + d_B$$

$$X_A \geq X_C + d_C$$

$$X_B \geq X_A + d_A$$

$$X_C \geq X_B + d_B$$

$$X_A \geq X_C + d_C - nT$$

$$X_A + X_B + X_C \geq X_A + X_B + X_C + d_A + d_B + d_C$$

$$0 \geq d_A + d_B + d_C$$

$$X_A + X_B + X_C \geq X_A + X_B + X_C + d_A + d_B + d_C + nT$$

$$nT \geq d_A + d_B + d_C$$

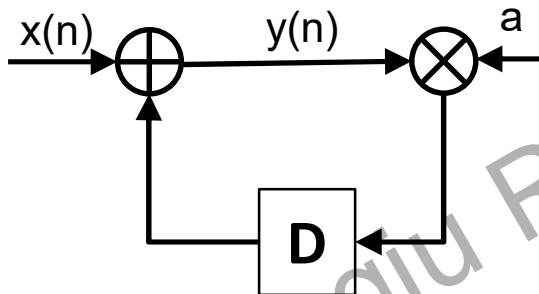
**Delay-free loops are non-computable**

## Example: DSP Program

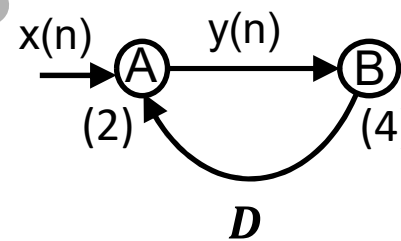
$$Y(n) = a y(n-1) + x(n)$$

Adder (A): 2 cycles, Multiplier (B) : 4 cycles

Block Diagram



Data-Flow Graph

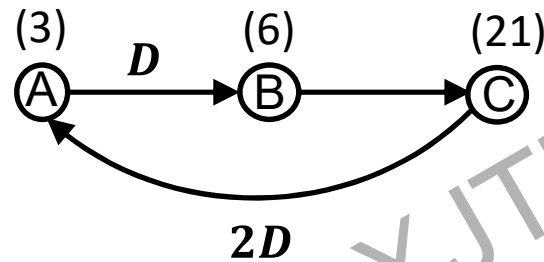


**Intra-Iteration Period** (edges with no delay elements) :  $A_K \rightarrow B_K$

**Inter-Iteration Period** (edges with delay elements):  $B_K \Rightarrow A_{K+1}$

$$A_i \rightarrow B_i \Rightarrow A_{i+1} \rightarrow B_{i+1} \Rightarrow A_{i+2} \rightarrow B_{i+2} \Rightarrow \dots$$

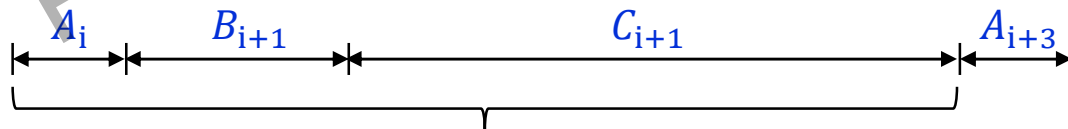
## Example: DSP Program



$$A_i \Rightarrow B_{i+1} \rightarrow C_{i+1} \Rightarrow A_{i+3} \Rightarrow B_{i+4} \rightarrow C_{i+4} \Rightarrow \dots$$

$$A_{i+1} \Rightarrow B_{i+2} \rightarrow C_{i+2} \Rightarrow A_{i+4} \Rightarrow B_{i+5} \rightarrow C_{i+5} \Rightarrow \dots$$

$$A_{i+2} \Rightarrow B_{i+3} \rightarrow C_{i+3} \Rightarrow A_{i+5} \Rightarrow B_{i+6} \rightarrow C_{i+6} \Rightarrow \dots$$

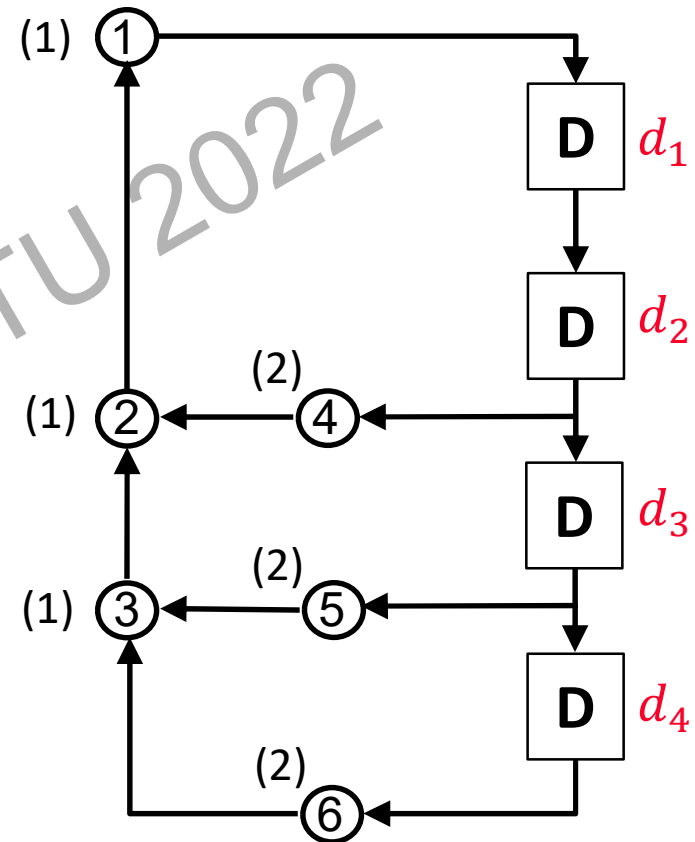


$$3T \geq d_A + d_B + d_C = 3 + 6 + 21 = 30$$

# Loop Bounds in DFG (Data flow Graph)

- **Critical Path:** The path with the longest computation time among all paths that contain *zero delays*
- **Loop:** Directed path that begins and ends at the same node
- **Loop Bound** of the *i-th* loop =  $t_i / w_i$ , where  $t_i$  is the loop computation time and  $w_i$  is the number of delays in the *i-th* loop
- **Critical Loop:** the loops in which has maximum loop bound.
- **Iteration Bound:** maximum loop bound,  

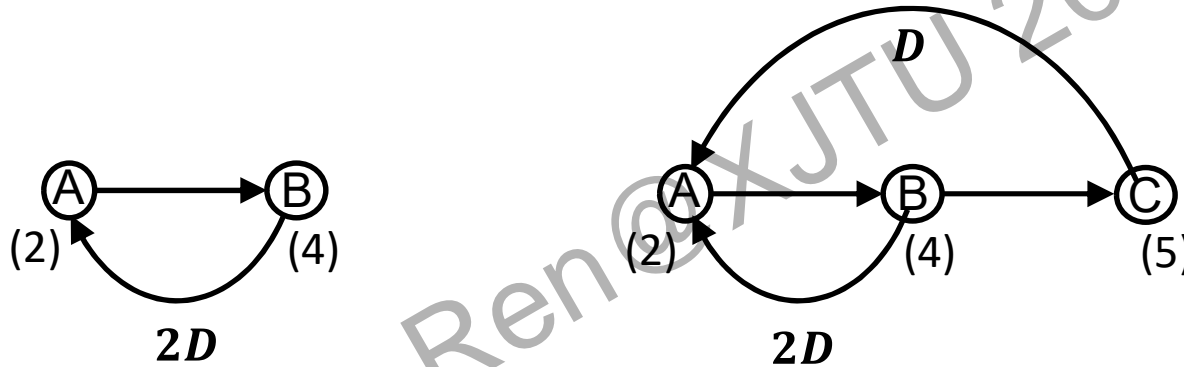
$$T_{\infty} = \max_{i \in L} \left\{ \frac{t_i}{w_i} \right\}$$
, where  $L$  is the set of loops in the DFG. i.e., a fundamental limit for recursive algorithms



**Loop bounds: 4/2 u.t., 5/3 u.t., 5/4 u.t.**

# Iteration Bound

Definition:  $T_{\infty} = \max_{i \in L} \left\{ \frac{t_i}{w_i} \right\}$



*Loop bounds: 6/2 u.t.*

*Iteration bound = 3 u.t.*

*Loop bounds: 6/2 u.t., 11/1 u.t.*

*Critical loop : A->B->C->A*

*Iteration bound = 11 u.t.*

# Longest Path Matrix (LPM) Algorithm

A series of matrices ( $L^{(1)} \sim L^{(m)}$ ,  $m=1, 2, 3, \dots, d$ ) are constructed, and the iteration bound is found by examining the **diagonal elements** of the matrices.

$d$  : # of delays in the DFG

$L_{i,j}^{(m)}$  : Element of Matrix  $L^{(m)}$ , which is the **longest computation time** of all paths from delay element  $di$  to delay element  $dj$  that pass through exactly  $m-1$  delays, if no path exists, then the value of  $L_{i,j}^{(m)}$  equals **-1**.

Usually,  $L_{i,j}^{(1)}$  is computed using DFG, The higher order matrices are computed recursively:

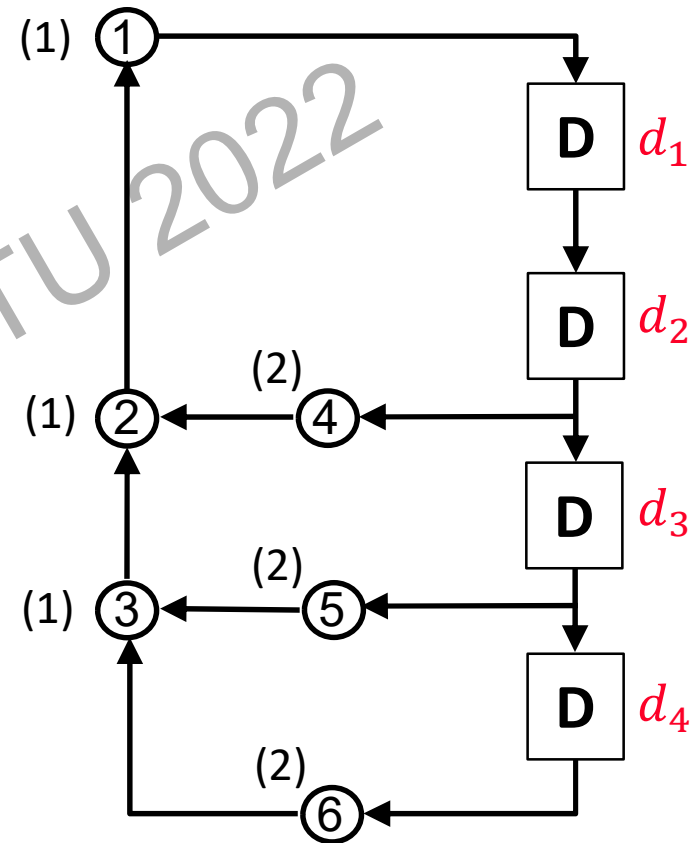
$$L_{i,j}^{(m)} = \max_{k \in K} (-1, L_{i,k}^{(1)} + L_{k,j}^{(m-1)})$$

The **iteration bound** is given by:

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

## A DFG with Three Loops Using LPM

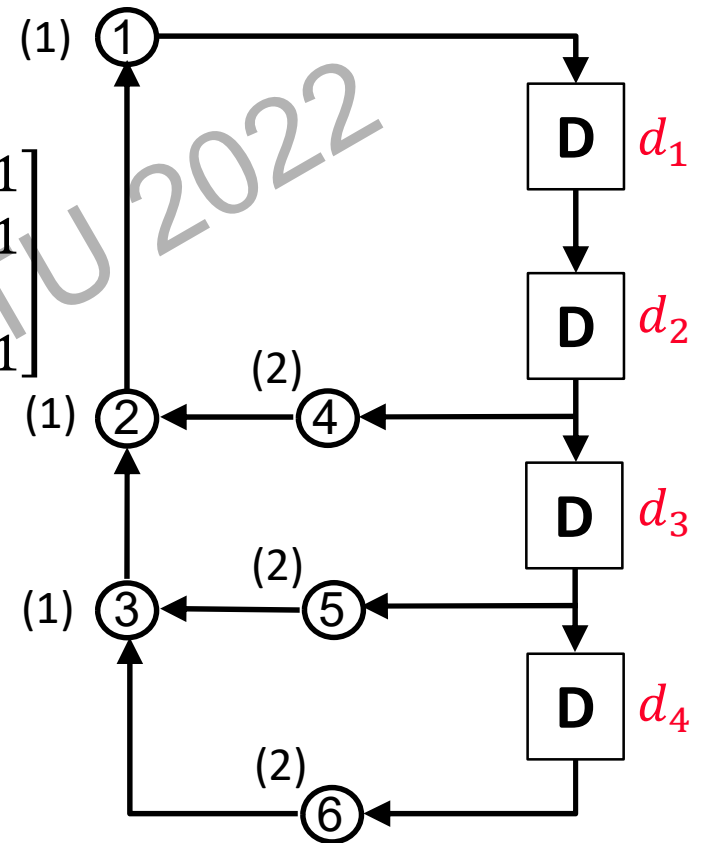
$$L^{(1)} = \begin{bmatrix} -1 & 0 & -1 & -1 \\ 4 & -1 & 0 & -1 \\ 5 & -1 & -1 & 0 \\ 5 & -1 & -1 & -1 \end{bmatrix}$$



# A DFG with Three Loops Using LPM

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

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



$$L_{i,j}^{(m)} = \max_{k \in K} (-1, L_{i,k}^{(1)} + L_{k,j}^{(m-1)})$$

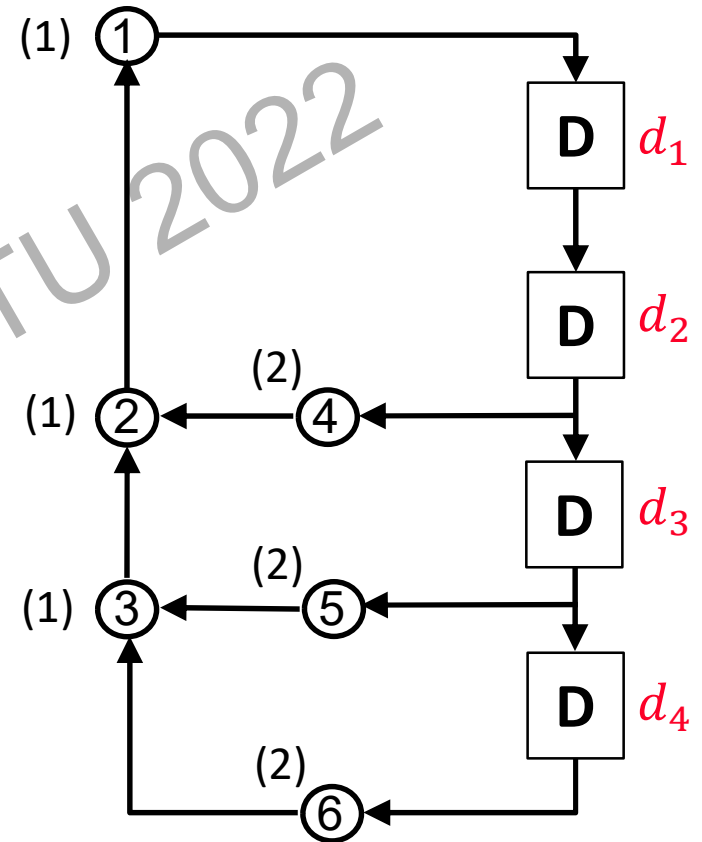
# A DFG with Three Loops Using LPM

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

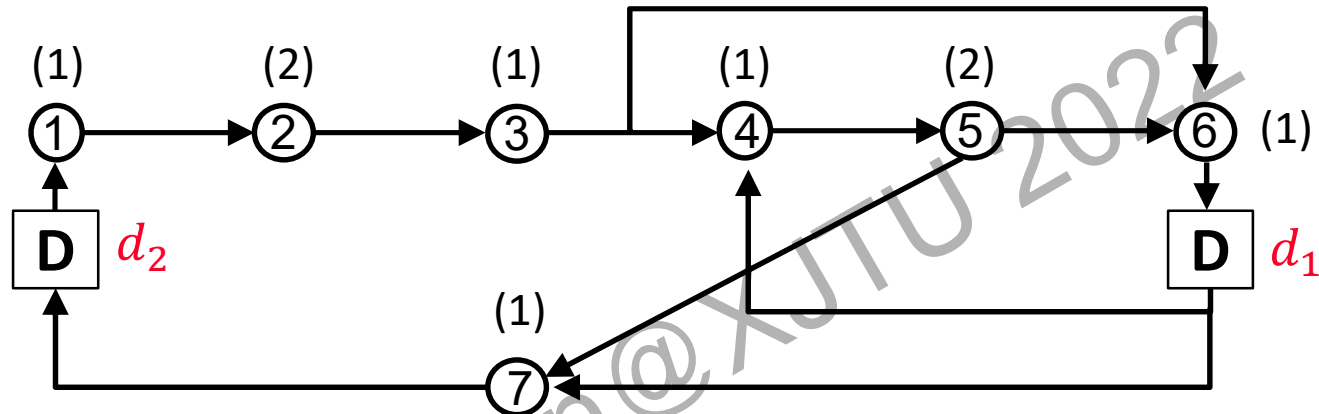
$$L^{(4)} = \begin{bmatrix} 8 & 5 & 4 & -1 \\ 9 & 8 & 5 & 4 \\ 10 & 9 & 5 & 5 \\ 10 & 9 & -1 & 5 \end{bmatrix}$$

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

$$T_{\infty} = \max_{i,m \in \{1,2,\dots,4\}} \left\{ \frac{4}{2}, \frac{4}{2}, \frac{5}{3}, \frac{5}{3}, \frac{5}{3}, \frac{8}{4}, \frac{8}{4}, \frac{5}{4}, \frac{5}{4} \right\} = 2$$



## A Filter Using LPM



$$L^{(1)} = \begin{bmatrix} 4 & 4 \\ 8 & 8 \end{bmatrix}$$

$$L^{(2)} = \begin{bmatrix} 12 & 12 \\ 16 & 16 \end{bmatrix}$$

$$T_{\infty} = \max \left\{ \frac{4}{1}, \frac{8}{1}, \frac{12}{2}, \frac{16}{2} \right\} = 8$$

# Minimum Cycle Mean (MCM) Method

## ■ Step 1: Construct the new graph $G_d$ and $\overline{G_d}$

- Transform from original DFG  $G$ .
- Decide the # of nodes from the # of delays in  $G$ .
- Decide the weight of each edge

## ■ Step 2: Compute the minimum cycle mean

- Construct the series of  $d+1$  vectors  $f^{(m)}$ ,  $m=0, 1, 2, \dots, d$  the dimension of  $f^{(m)}$  is  $d \times 1$

An arbitrary reference node is chosen in  $G_d$  (called this node  $s$ ). The initial vector  $f^{(0)}$  is formed by setting  $f^{(0)}(s) = 0$  and setting the remaining nodes of  $f^{(0)}$  to infinity ( $\infty$ ).

- The remaining vectors  $f^{(m)}$ ,  $m=1, 2, \dots, d$  are recursively computed:

$$f^{(m)}(j) = \min_{i \in I} (f^{(m-1)}(i) + \bar{w}(i, j))$$

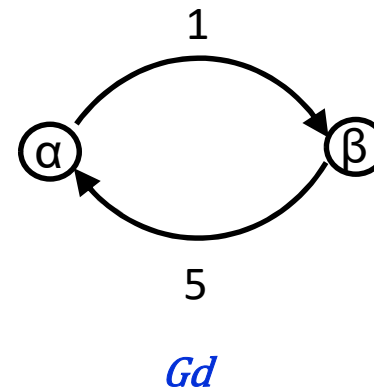
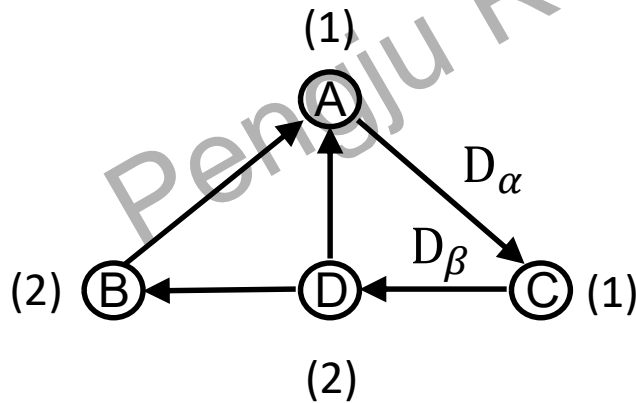
$I$  is the set of nodes such that there exists an edge from node  $i$  to node  $j$

- find the min cycle mean

$$T_{\infty} = - \min_{i \in \{1, 2, \dots, d\}} \left( \max_{m \in \{0, 1, 2, \dots, d-1\}} \left( \frac{f^{(d)}(i) - f^{(m)}(i)}{d - m} \right) \right)$$

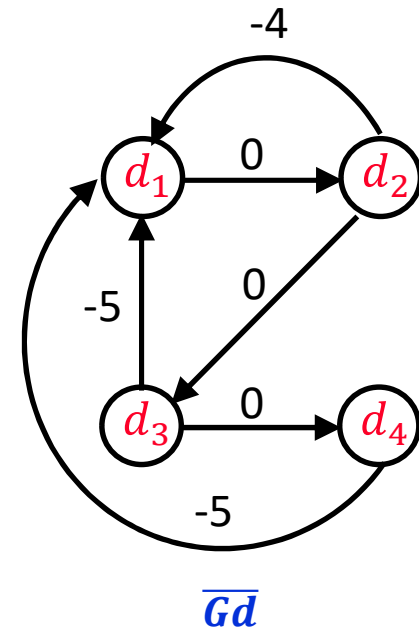
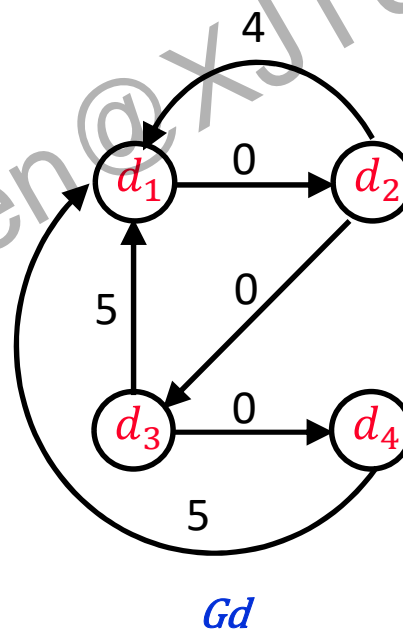
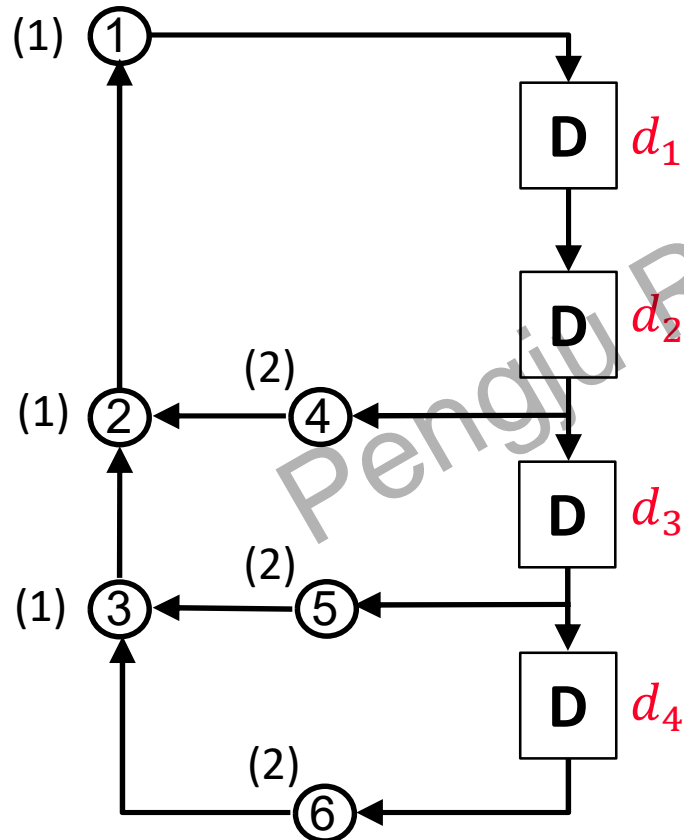
# A DFG with Three Loops Using MCM

- Cycle mean = Average length of the edge in  $c$  (Cycle = Loop)
- Longest path length
  - Path that passes through no delays
  - Longest: two loops that contain  $D_\alpha$  and  $D_\beta$ 
    - $\text{Max}\{6, 4\} = 6$
    - Cycle mean =  $6/2 = 3$



# A DFG with Three Loops Using MCM

- delay => node
- longest path length (computation time) => weight  $w(i,j)$
- If no zero-delay path exists from delay  $d_i$  to delay  $d_j$ , then the edge  $i \rightarrow j$  does not exist in  $G_d$ .



# A DFG with Three Loops Using MCM

- We will find  $d+1$  vectors,  $f^{(m)} \quad m = 0, 1, \dots, d$

$$f^{(0)} = \begin{bmatrix} 0 \\ \infty \\ \infty \\ \infty \end{bmatrix} \quad f^{(1)} = \begin{bmatrix} \infty \\ 0 \\ \infty \\ \infty \end{bmatrix} \quad f^{(2)} = \begin{bmatrix} -4 \\ \infty \\ 0 \\ \infty \end{bmatrix} \quad f^{(3)} = \begin{bmatrix} -5 \\ -4 \\ \infty \\ 0 \end{bmatrix} \quad f^{(4)} = \begin{bmatrix} -8 \\ -5 \\ -4 \\ \infty \end{bmatrix}$$

$$f^{(3)}(j) = \min_{i \in I} (f^{(m-1)}(i) + \bar{w}(i, j))$$

$\bar{w}(i, j)$  is the weight of the edge  $i \rightarrow j$  in and  $I$  is the set of nodes in such that there exists an edge from node  $i$  to node  $j$  ( $i \rightarrow j$ ).

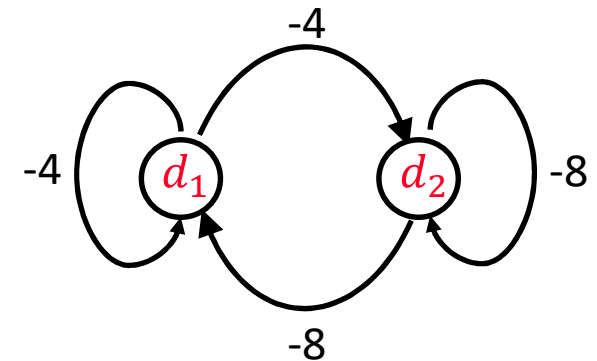
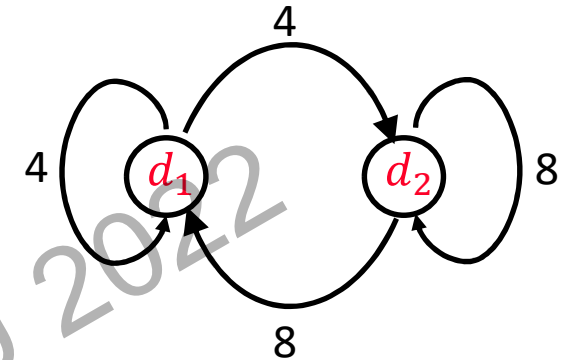
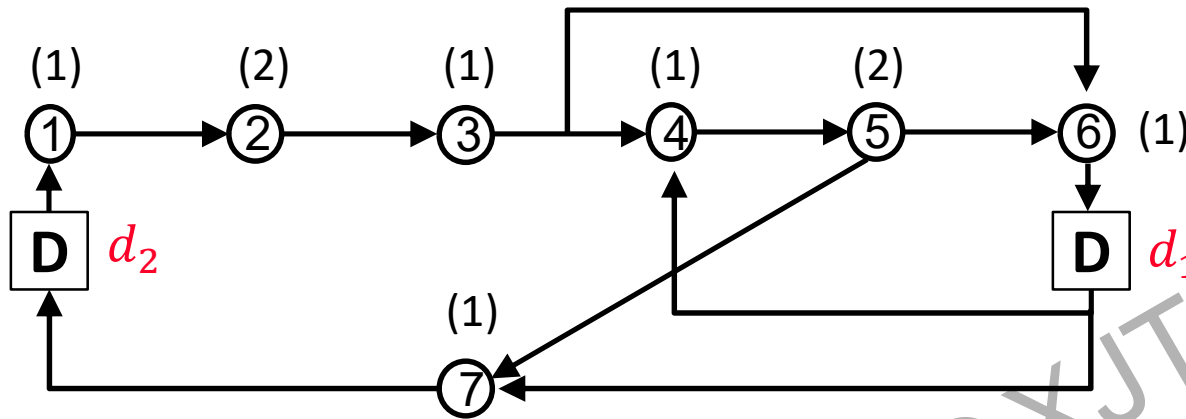
## A DFG with Three Loops Using MCM

$$T_{\infty} = - \min_{i \in \{1, 2, \dots, d\}} \left( \max_{m \in \{0, 1, 2, \dots, d-1\}} \left( \frac{f^{(d)}(i) - f^{(m)}(i)}{d - m} \right) \right)$$

	m=0	m=1	m=2	m=3	$\max_{m \in \{0, 1, 2, \dots, 3\}} \left( \frac{f^{(4)}(i) - f^{(m)}(i)}{4 - m} \right)$
i=1	-2	$-\infty$	-2	-3	-2
i=2	$-\infty$	-5/3	$-\infty$	-1	-1
i=3	$-\infty$	$-\infty$	-2	$-\infty$	-2
i=4	$\infty - \infty$	$\infty - \infty$	$\infty - \infty$	$\infty$	$\infty$

$$T_{\infty} = -\min(-2, -1, -2, \infty) = 2$$

# A Filter Using MCM



	m=0	m=1	$\max_{m \in \{0,1\}} \left( \frac{f^{(2)}(i) - f^{(m)}(i)}{2 - m} \right)$
i=1	-12/2	-8/1	-6
i=2	$-\infty$	-8/1	-8

$$f^{(0)} = \begin{bmatrix} 0 \\ \infty \end{bmatrix} f^{(1)} = \begin{bmatrix} -4 \\ -4 \end{bmatrix} f^{(2)} = \begin{bmatrix} -12 \\ -12 \end{bmatrix}$$

$$T_{\infty} = -\min(-8, -6) = 8$$

## Conclusion

When the DFG is recursive, the **iteration bound** is the fundamental limit on the minimum **sample period** of a hardware implementation of the Data-stream program.

Two algorithms to compute iteration bound, **LPM** and **MCM** are explored.

*Next Lecture : Retiming & Pipelining*

Pengju Ren@KUTU 2022