Introduction of Processor Design for AI Applications

L05 – Retiming and Pipelining

Pengju Ren
Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University

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

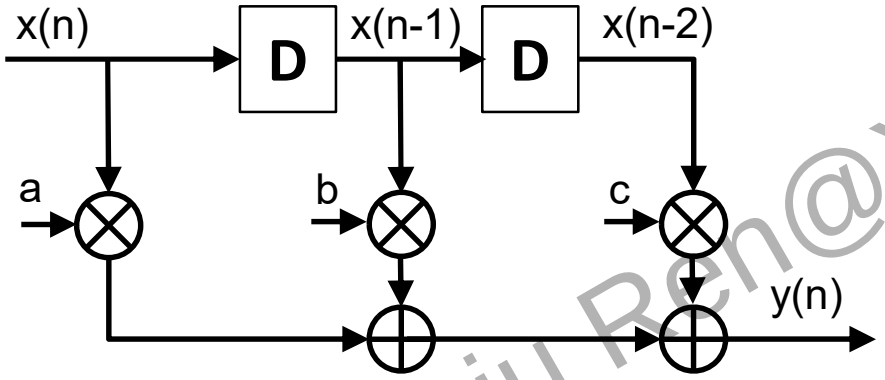
LTI Systems

- Linear systems
  - Assume x1 (n)->y1 (n) and x2 (n)->y2 (n), where "->" denotes "lead to". If ax1 (n)+bx2 (n) -> ay1 (n)+by2 (n), then the systems is referred to as a "Linear System."
  - Homogenous and additive properties
- Time-invariant (TI) systems
  - x(n-n0)->y(n-n0)
- LTI systems
  - y(n)=h(n)*x(n)
- Causal systems
  - y(n_0) depends only on x(n), where n <= n_0

Pipelining of FIR Digital Filters (1)

Leads to a reduction in the critical path

$$FIR: \quad y(n) = a * x(n) + b * x(n-1) + c * x(n-2)$$



T_M: *Execution time of Muliplication*

T_A: *Execution time of Addition*

The **critical path** (or the minimum time required for processing a new sample) is limited by 1 multiply and 2 add times. Thus, the "sample period" (or the "sample frequency") is given by:

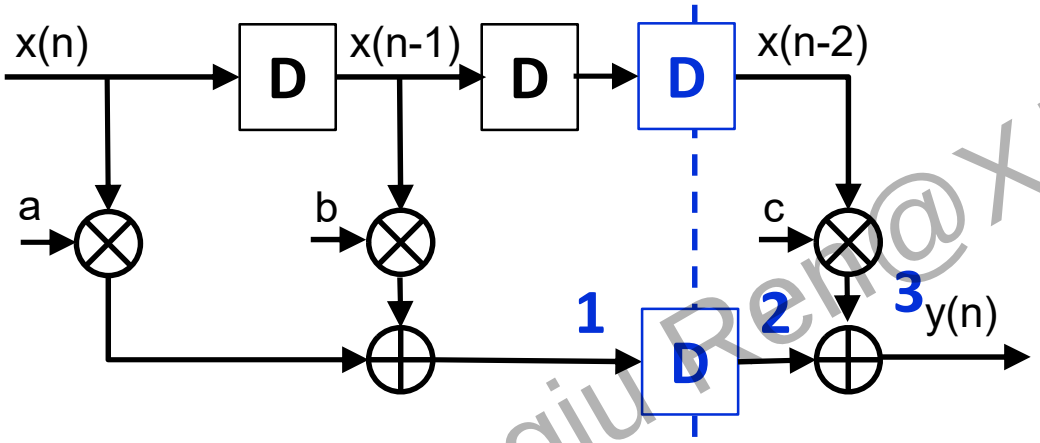
$$T_{sample} \geq T_M + 2T_A$$

$$f_{sample} \leq \frac{1}{T_M + 2T_A}$$

Pipelining of FIR Digital Filters (2)

Pipelining: reduce the effective critical path by introducing **latches** along the critical data path

$$FIR: \quad y(n) = ax(n) + bx(n-1) + cx(n-2)$$



$$T_{sample} \geq T_M + T_A$$

$$f_{sample} \leq \frac{1}{T_M + T_A}$$

Clock	Input	Node1	Node2	Node3	Output
0	$x(n)$	$ax(n) + bx(n-1)$	$ax(n-1) + bx(n-2)$	$cx(n-3)$	$y(n-2)$
1	$x(n+1)$	$ax(n+1) + bx(n)$	$ax(n) + bx(n-1)$	$cx(n-2)$	$y(n-1)$
2	$x(n+2)$	$ax(n+2) + bx(n+1)$	$ax(n+1) + bx(n)$	$cx(n-1)$	$y(n)$
3	$x(n+3)$	$ax(n+3) + bx(n+2)$	$ax(n+2) + bx(n+1)$	$cx(n)$	$y(n+1)$

Pipelining of FIR Digital Filters (3)

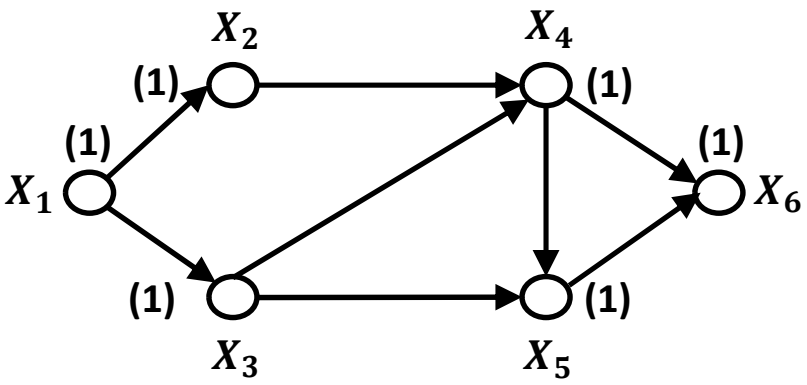
Pipelining reduces the critical path, but leads to a penalty in terms of an increased latency and the number of latches.

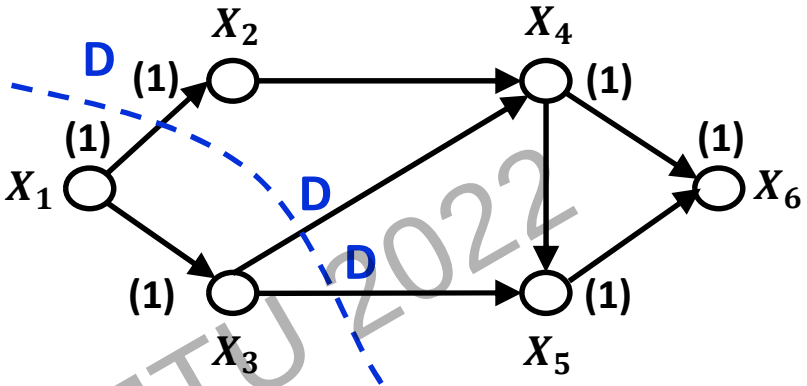
This longest path or the "critical path" can be reduced by suitably placing the pipelining latches in the DSP architecture – The pipelining latches can only be placed across any ***feed-forward cutset*** of the graph

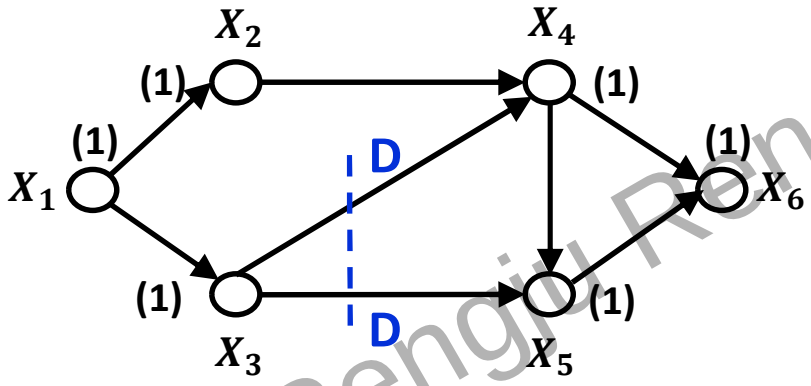
Cutset: a set of edges of a graph such that if these edges are removed from the graph, the graph becomes disjoint

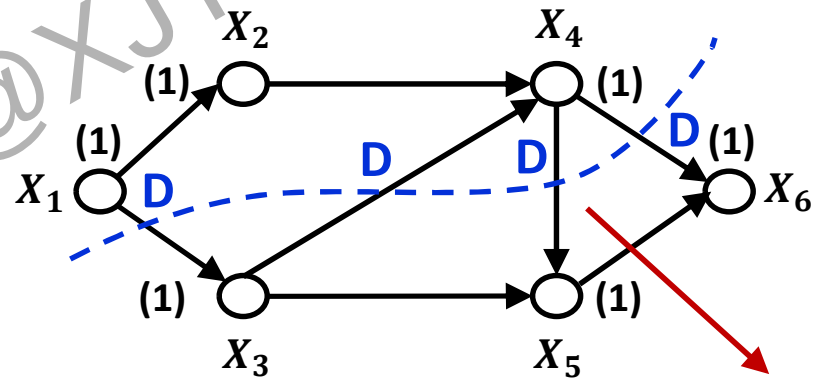
Feed-forward cutset: data move in the forward direction on all the edge of the cutset

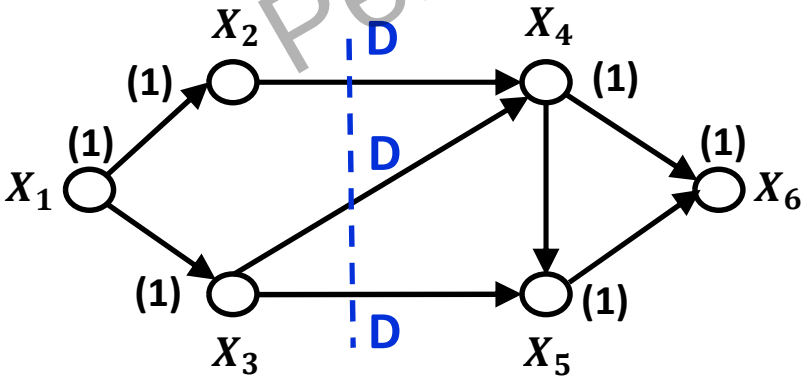
SFG representation of Cutset





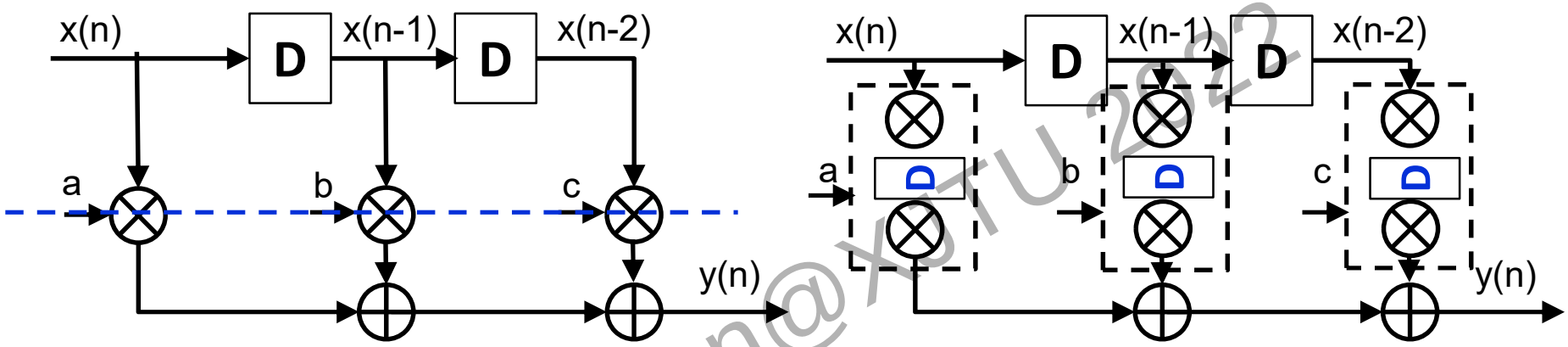






Invalid pipelining

Fine-grain Pipelining of FIR Filter

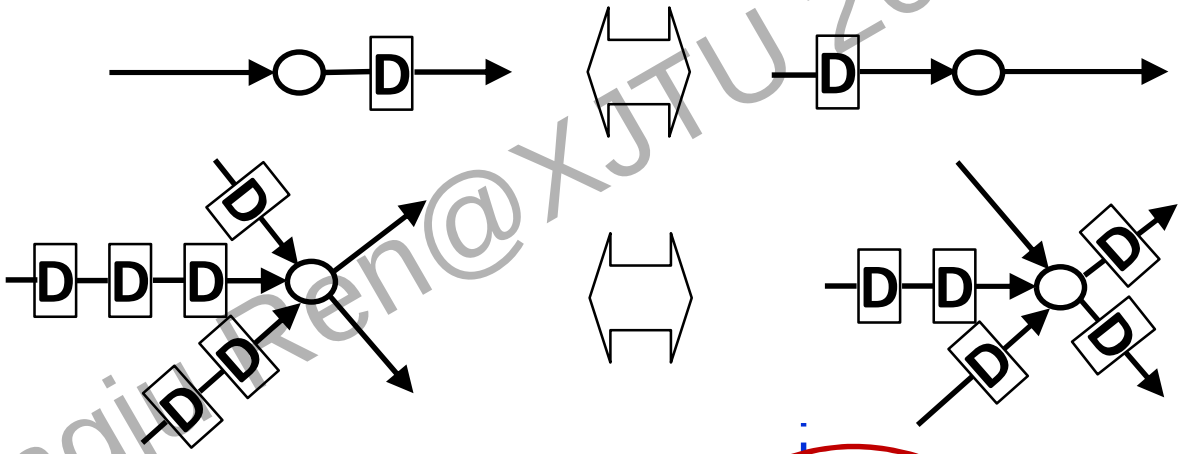


Let T_M =10 units and T_A =2 units. If the multiplier is broken into 2 smaller units (T_{M1} and T_{M2}) with processing times of 5 units, respectively (by placing the latches on the horizontal cutset across the multiplier), then the desired clock period can be achieved as 7 units

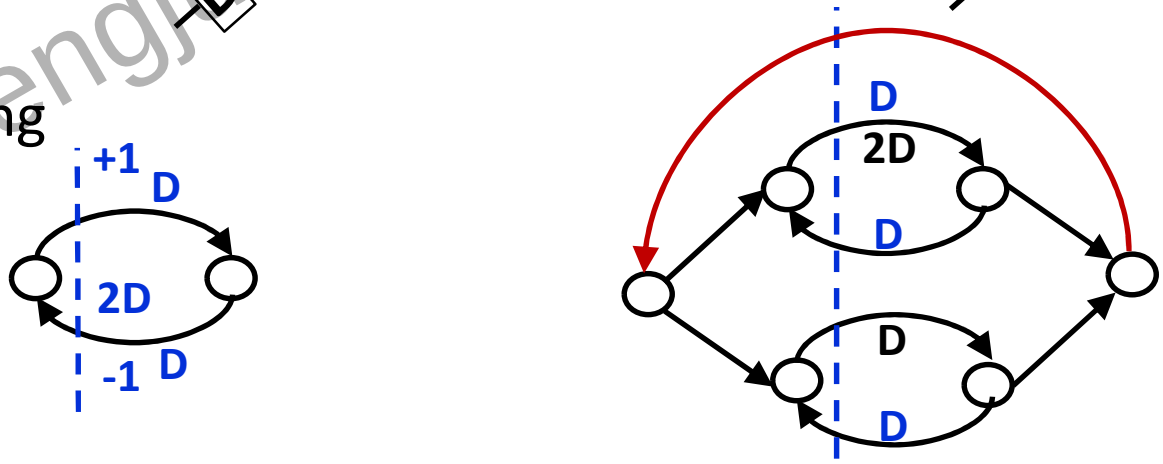
Retiming and Example (1)

A transformation technique used to **change the location of delay elements** in the circuit without affecting the input/output characteristics, changing 1) **the clock period** and 2) **the number of registers**

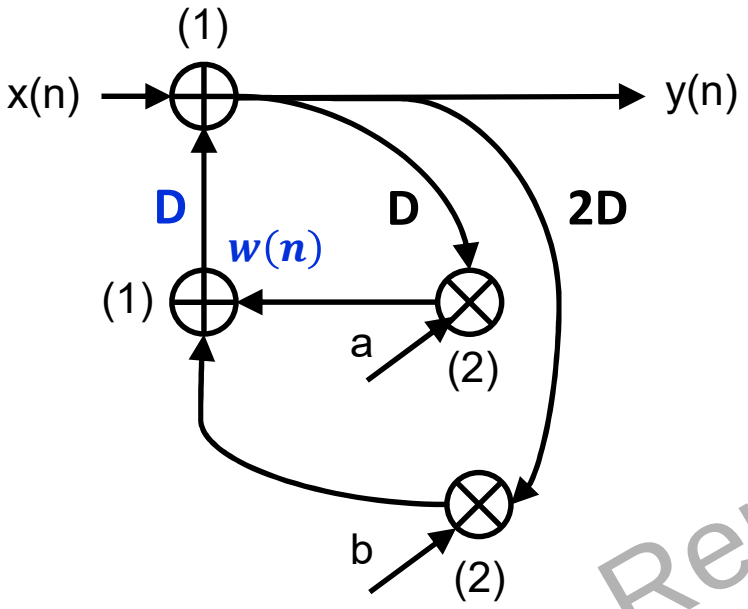
Node Retiming

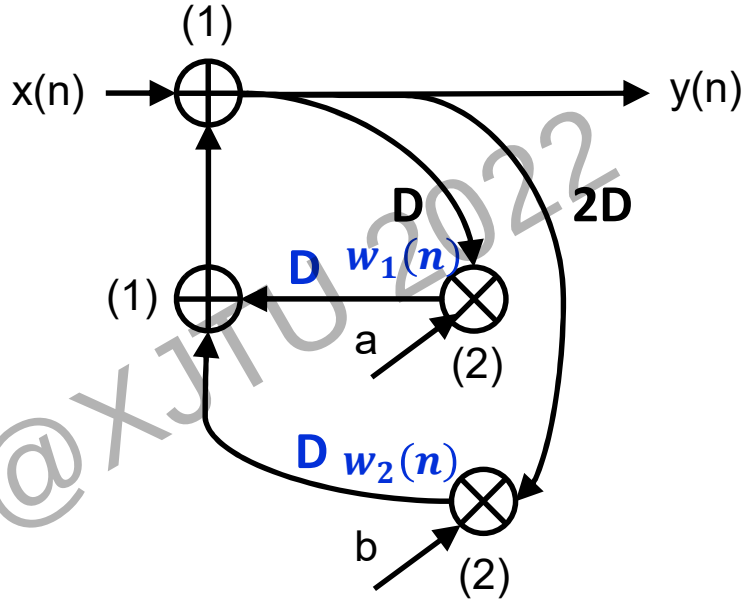


Cutset Retiming



Retiming and Example (2)





$$w(n) = ay(n-1) + by(n-2)$$
$$y(n) = w(n-1) + x(n)$$
$$= ay(n-2) + by(n-3) + x(n)$$

Critical Path：$T_M + T_A = 3$ u.t

Number of registers: 4

$$w_1(n) = ay(n-1)$$
$$w_2(n) = by(n-2)$$
$$y(n) = w_1(n-1) + w_2(n-1) + x(n)$$
$$= ay(n-2) + by(n-3) + x(n)$$

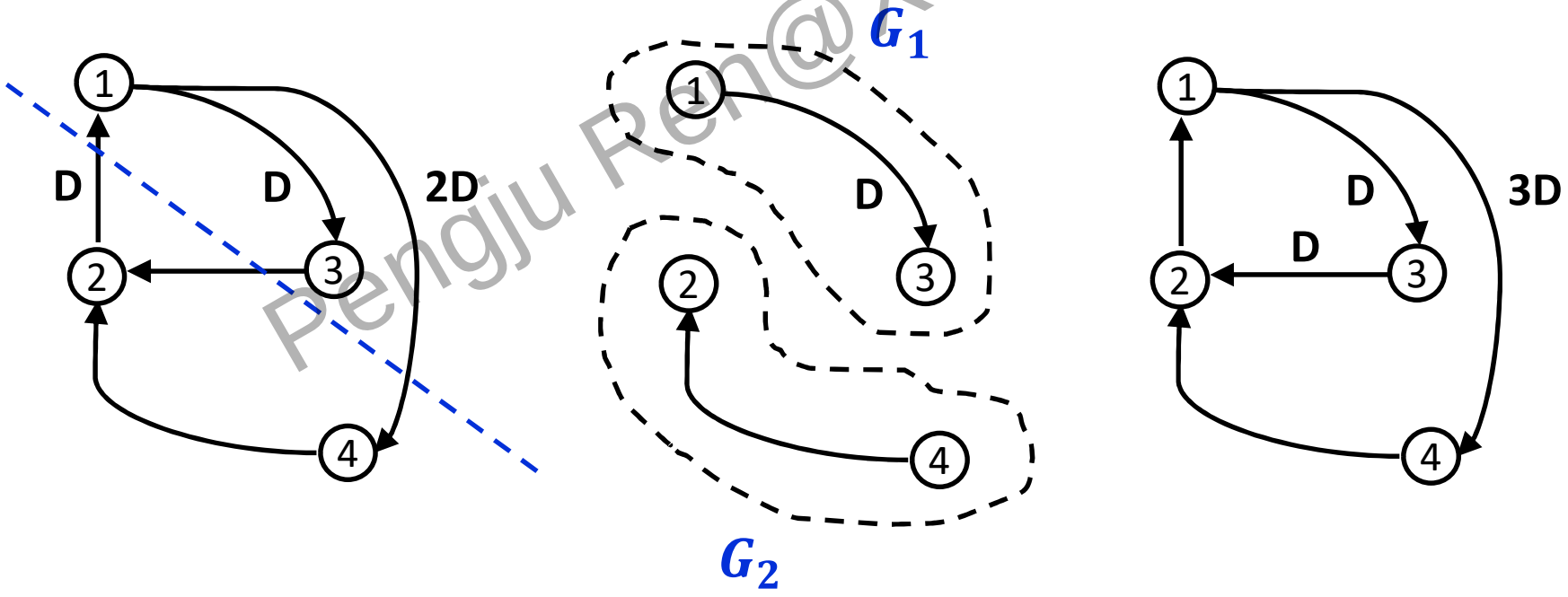
Critical Path: $2T_A = 2\ u.t$

Number of registers: 5

Retiming Formulation (1)

Cutset retiming only affects the weights of edges in the cutset.

If the 2 disconnected subgraphs are labeled G_1 and G_2, then cutset retiming consists of adding k delays to each edge from G_1 to G_2, and removing k delays to each edge from G_2 to G_1.



Retiming Formulation (2)

Map circuit $G \text{->} G_r$ using retiming

Retiming can be presented with $r(X)$, X is one of the nodes in the circuit

For an edge U$\xrightarrow{e}$ V (U is Source, V is the destination)

$$w'(e) = w(e) + r(V) - r(U)$$

$w(e)$: weight (delay) of the edge U$\xrightarrow{e}$ V in the origin circuit

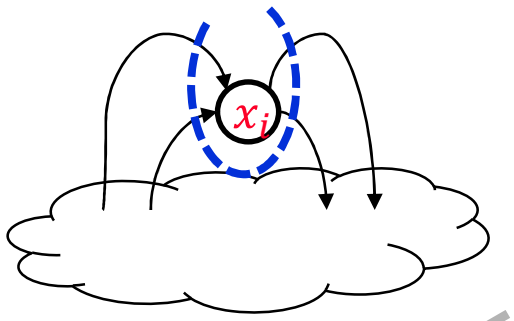
$w'(e)$: weight of the edge U$\xrightarrow{e}$ V in the retimed circuit

A retiming solution is feasible if :$\boldsymbol{w'(e) \geq 0, \forall e \epsilon G}$

$$w(e) + r(V) - r(U) \geq \mathbf{0} \Rightarrow r(U) - r(V) \leq w(e)$$

One inequality per edge due to the causality of the system

Retiming Formulation: cutset contains 1 node



If cutset = $\{x_i\}$: we set r(x_i) = $\begin{cases} \mathrm{k} \ if \ x = x_i \\ 0, \quad others \end{cases}$ x *are nodes in* G

Therefore, for an edge U $\xrightarrow{e}$ V in G (U is Source, V is destination)

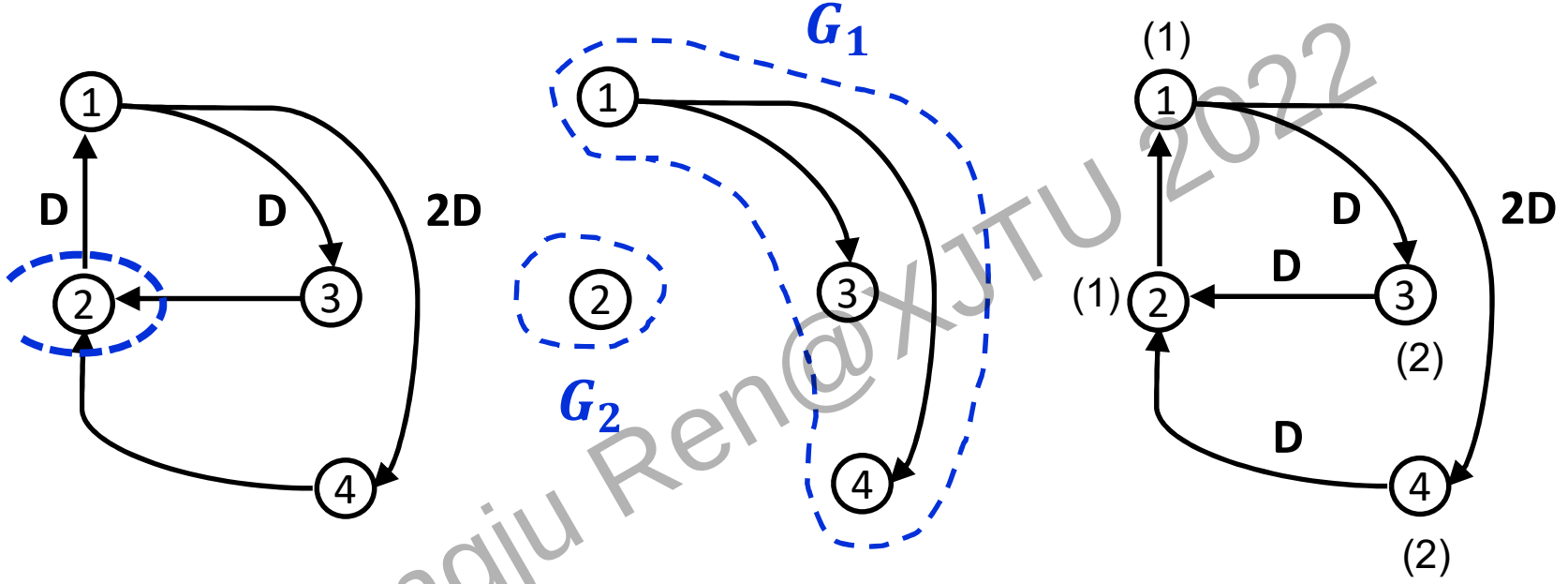
$$r(V) - r(U) = \begin{cases} k, & if \ \mathrm{V} = x_i \\ -k, & if \ \mathrm{U} = x_i \\ 0, & others \end{cases}$$

$$w'(e) = w(e) + r(V) - r(U) = \begin{cases} w(e) + k, & if \ \mathrm{V} = x_i \\ w(e) - k, & if \ \mathrm{U} = x_i \\ w(e), & others \end{cases}$$

In-degree edges of x_i

Out-degree edges of x_i

Retiming Formulation: cutset contains one node

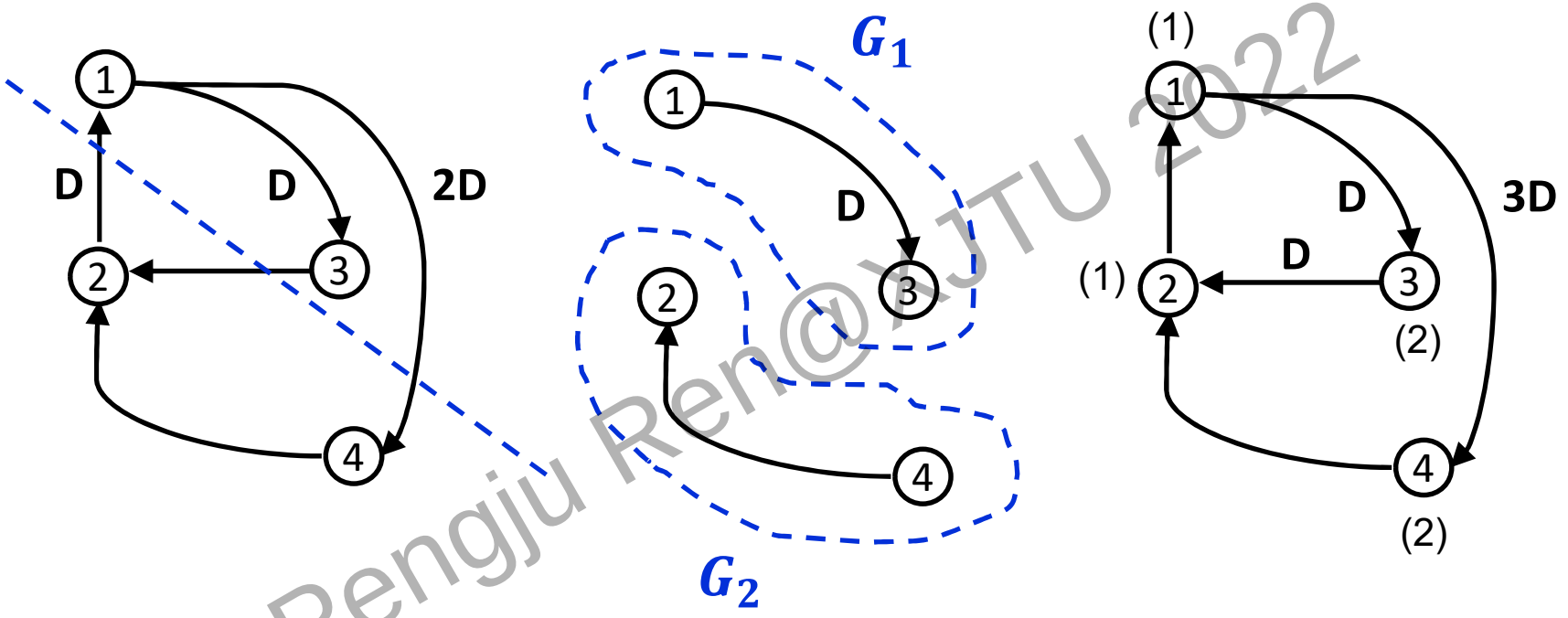


$$r(x_i) = \begin{cases} 0, if\ x_i \in \mathrm{G}_1 \\ k, if\ x_i \in \mathrm{G}_2 \end{cases} \Rightarrow \begin{matrix} r(1) = 0 \\ r(2) = k \\ r(3) = 0 \\ r(4) = 0 \end{matrix} \Rightarrow$$

$$w'(1 \to 3) = 1 + 0 - 0 = 1$$
$$w'(1 \to 4) = 2 + 0 - 0 = 2$$
$$w'(2 \to 1) = 1 + 0 - k = 0 \rightarrow \boldsymbol{G_2}\text{-> }\boldsymbol{G_1}: \boldsymbol{-1}$$
$$w'(3 \to 2) = 0 + k - 0 = 1$$
$$w'(4 \to 2) = 0 + k - 0 = 1 \Big\} \boldsymbol{G_1}\text{-> }\boldsymbol{G_2}: \boldsymbol{+1}$$

Feasible solution of cutset retiming: $-\min(w(e), e \in \boldsymbol{G_1}\text{-> }\boldsymbol{G_2}) \le \mathrm{k} \le \min(w(e), e \in \boldsymbol{G_2}\text{-> }\boldsymbol{G_1})$

Retiming Formulation: cutset contains many nodes



$$r(x_i) = \begin{cases} 0, if\ x_i \in G_1 \\ k, if\ x_i \in G_2 \end{cases} \Rightarrow \begin{matrix} r(1) = 0 \\ r(2) = k \\ r(3) = 0 \\ r(4) = k \end{matrix} \Rightarrow$$

$$w'(1 \to 3) = 1 + k - k = 1$$
$$w'(2 \to 1) = 1 + 0 - k = 0 \quad \rightarrow \ \boldsymbol{G_2}\text{-> }\boldsymbol{G_1}: -1$$
$$w'(1 \to 4) = 2 + k - 0 = 3$$
$$w'(3 \to 2) = 0 + k - 0 = 1 \quad \} \ \boldsymbol{G_1}\text{-> }\boldsymbol{G_2}: +1$$
$$w'(4 \to 2) = 0 + k - k = 0$$

Feasible solution of cutset retiming: $-\min(w(e), e \in \boldsymbol{G_1}\text{-> }\boldsymbol{G_2}) \leq \mathrm{k} \leq \min(w(e), e \in \boldsymbol{G_2}\text{-> }\boldsymbol{G_1})$

Properties of Retiming

The weight of the retimed path:

$p = \mathrm{V}_0 \xrightarrow{e_0} \mathrm{V}_1 \xrightarrow{e_1} \ldots \xrightarrow{e_{m-1}} \mathrm{V}_m$ is given by $w'(p) = w(p) + r(\mathrm{V}_m) - r(\mathrm{V}_0)$

Proof: $w'(p) = \sum_{i=0}^{m-1} w'(e_i)$

$$= \sum_{i=0}^{m-1} (w(e_i) + r(\mathrm{V}_{i+1}) - r(\mathrm{V}_i))$$
$$= \sum_{i=0}^{m-1} w(e_i) + (\sum_{i=0}^{m-1} r(\mathrm{V}_{i+1}) - \sum_{i=0}^{m-1} r(\mathrm{V}_i))$$
$$= w(p) + r(\mathrm{V}_m) - r(\mathrm{V}_0)$$

- Retiming does not change the number of delays in a cycle
- Retiming does not alter the *iteration bound* in a DFG
- Adding the constant value j to the retiming value of each node does not change the mapping from G *to* G_r

$$w'(e) = w(e) + (r(V) + j) - (r(U) + j)) = w(e) + r(V) - r(U)$$

Solving Systems of Inequalities (1)

- Given a set of M equalities in N variables, use ***shortest path algorithm*** to solve the results

Step 1: draw a constraint graph

- Draw the node x_i for each of the N variables x_i , i=1,2,…,N
- Draw the node x_{N+1}
- For each inequality $\mathrm{r}(x_i) - \mathrm{r}(x_j) \leq k$, draw the edge $x_j \rightarrow x_i$ from the node x_j to node x_i with length k
- For each node x_i , i=1,2,…n, draw the edge $x_{N+1} \rightarrow x_i$ from the node x_{N+1} to the node x_i with length 0

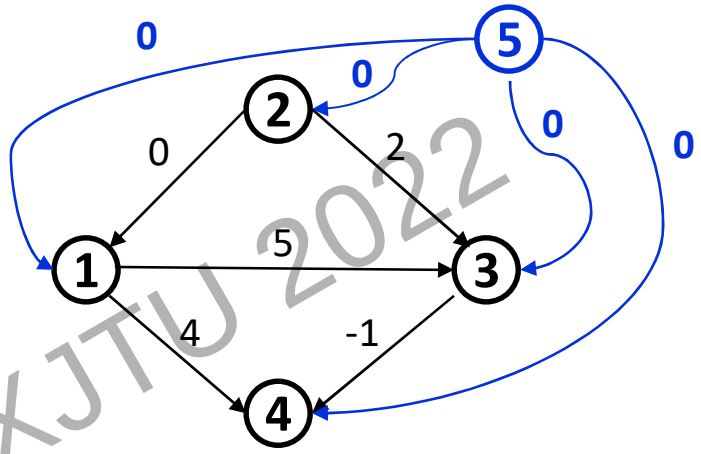
Solving Systems of Inequalities (2)

Step 2: Solve using a shortest path algorithm

- The system of inequalities has a solution if and only if the constraint graph contains *no negative cycles*
- If a solution exists, one solution is where $\mathrm{r}(x_i)$ is the minimum-length path from the node x_{N+1} to the node x_i

Example

$$r_1 - r_2 \le 0$$
$$r_3 - r_1 \le 5$$
$$r_4 - r_1 \le 4$$
$$r_4 - r_3 \le -1$$
$$r_3 - r_2 \le 2$$

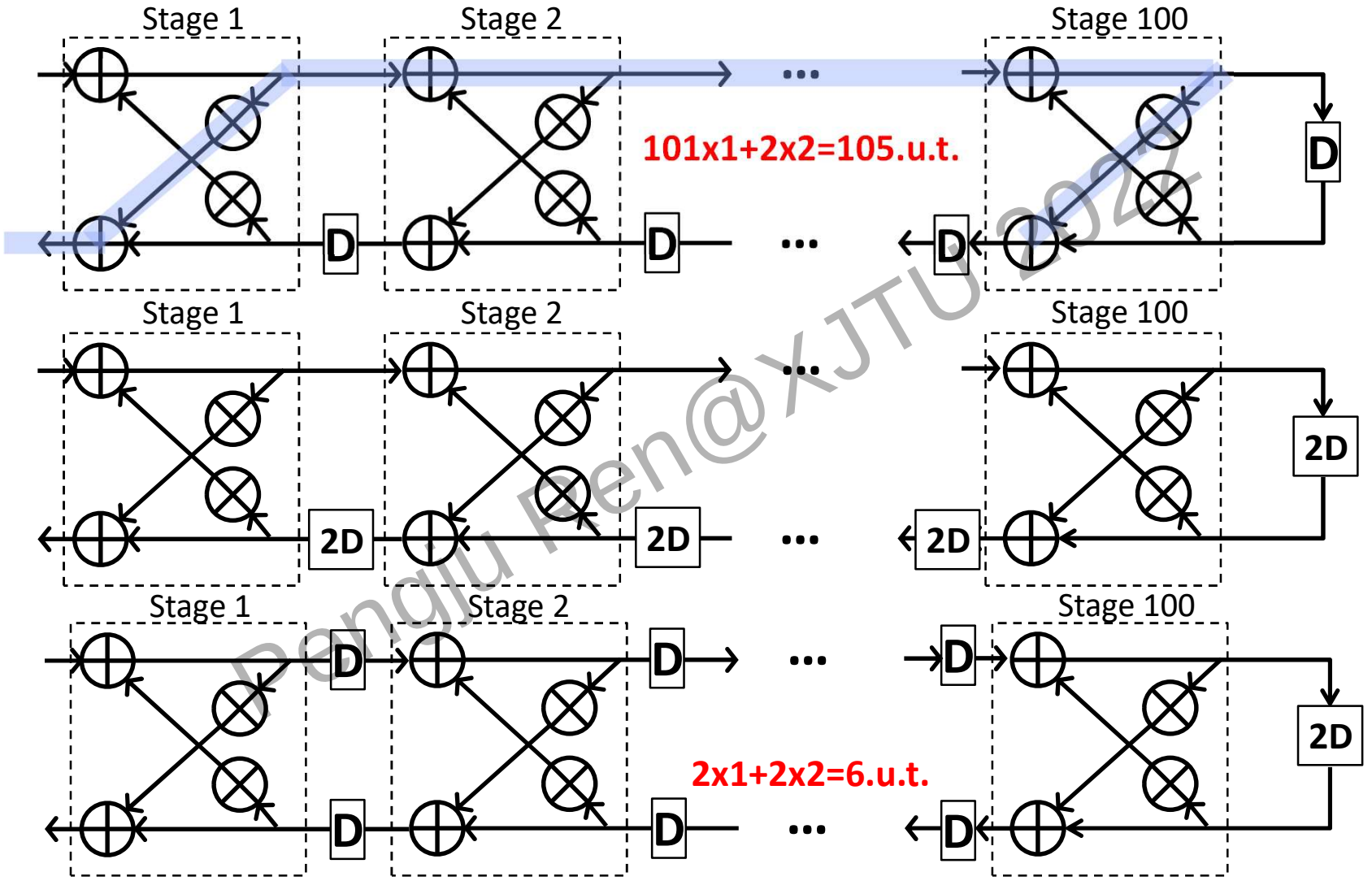


Bellman-Ford shortest path algorithm:

$$R^{(6)} = \begin{bmatrix} \infty & \infty & 5 & 4 & \infty \\ 0 & \infty & 2 & 1 & \infty \\ \infty & \infty & \infty & -1 & \infty \\ \infty & \infty & \infty & \infty & \infty \\ 0 & 0 & 0 & -1 & \infty \end{bmatrix}$$

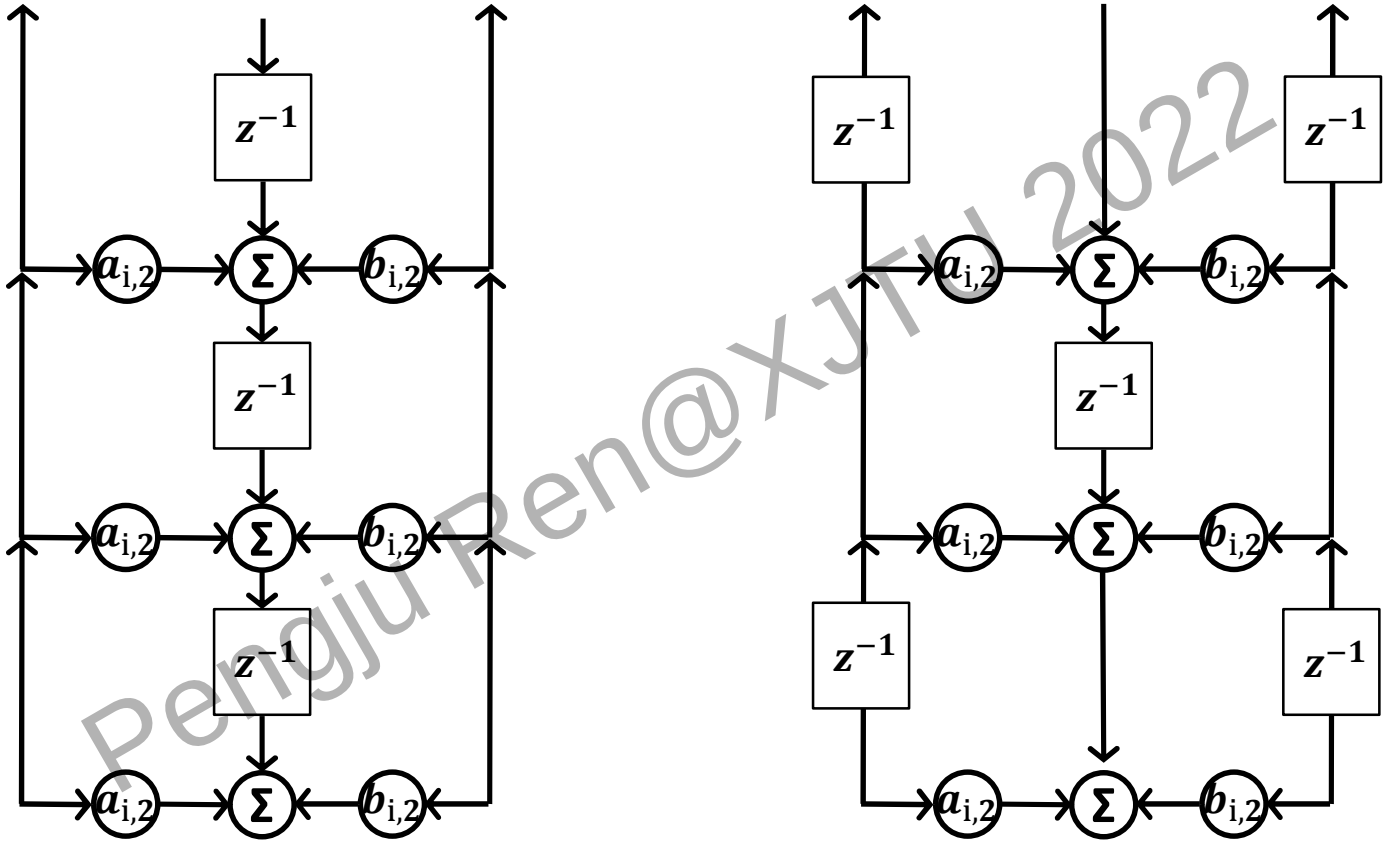
$$r_1 = 0, r_2 = 0, r_3 = 0, r_4 = -1$$

Cutset Retiming and Slow-down (Lattice Filter)



1) Replace each delay with **N** delays to create **N-slow** DFG
2) Perform cutset retiming on the **N-slow** DFG

Systolic Transformation



Source: L. D. Van, "A new 2-D systolic digital filter architecture **without global broadcast**," IEEE Trans. VLSI Systs., vol. 10, pp. 477-486, Aug. 2002

The Usage of Retiming Techniques

- Cutset retiming and pipelining
- Retiming for **clock period minimization**
- Retiming for **register minimization**

Retiming for clock period minimization (1)

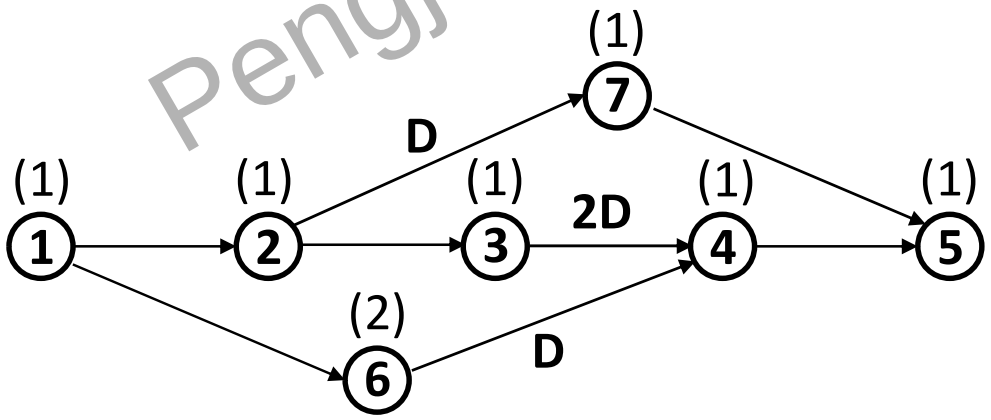
- Minimum feasible clock period or critical path :

$$\Phi(G) = max\{t(\boldsymbol{p}): U \xrightarrow{\boldsymbol{p}} V, \mathrm{w}(\boldsymbol{p}) = 0\}$$

Introduce two quantities $W(U,V)$ and $D(U,V)$ to determine if there is a retiming solution that can achieve a desired clock period :

- Minimum number of registers of $U \xrightarrow{\boldsymbol{P}} V$: $W(U,V) = min\{w(\boldsymbol{p}): U \xrightarrow{\boldsymbol{P}} V\}$
- Maximum computation time of $U \xrightarrow{\boldsymbol{P}} V$ with minimum number of registers:

$$D(U,V) = max\{t(\boldsymbol{p}): U \xrightarrow{\boldsymbol{P}} V \; and \; w(\boldsymbol{p}) = W(U,V)\}$$



$W(1,5) = \mathbf{1}$

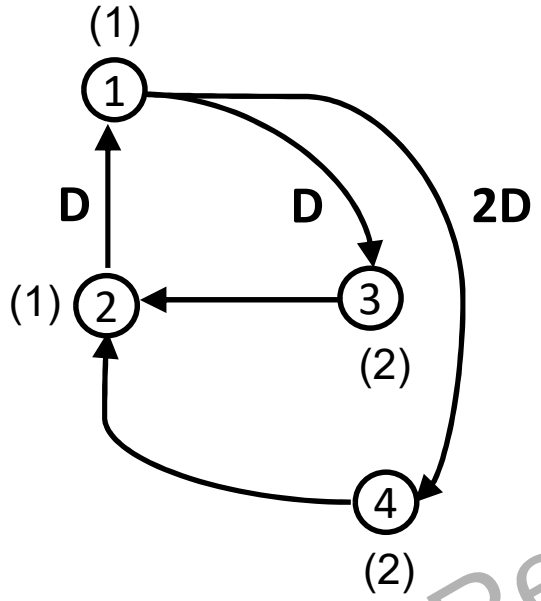
$D(1,5) = \mathbf{5}$

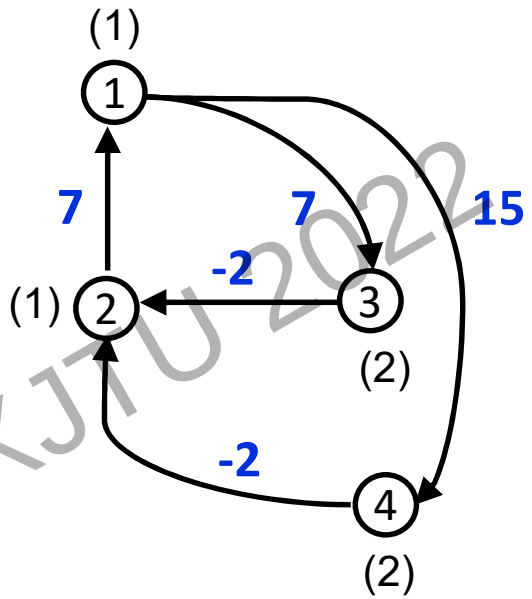
Retiming for clock period minimization (2)

Algorithm to compute $W(U,V)$ and $D(U,V)$

1. Let $M = t_{max}n$, where t_{max} is the maximum computation time of the nodes in G and n is the $\#$ of nodes in G.
2. Form a new graph G' which is the same as G except the edge weights are replaced by $w'(e) = Mw(e) - t(U)$ for all edges $U \xrightarrow{P} V$.
3. Solve for all pair shortest path problem on G' by using ***Floyd Warshall algorithm***. Let S'_{UV} be the ***shortest path*** form $U \xrightarrow{P} V$.
4. If $U \neq V$, then $W(U,V) = \lceil S'_{UV}/M \rceil$ and $D(U,V) = MW(U,V) - S'_{UV} + t(V)$. If $U = V$, then $W(U,V) = 0$ and $D(U,V) = t(U)$

Retiming for clock period minimization (3)





Step1: $M = t_{max}n = 2 \times 4 = 8$

Step2: New G' with $w'(e) = Mw(e) - t(U)$

Step3: pair shortest path problem on G'

Step4: If $U \neq V$, then $W(U,V) = \lceil S'_{UV}/M \rceil$ and $D(U,V) = MW(U,V) - S'_{UV} + t(V)$. If $U = V$, then $W(U,V) = 0$ and $D(U,V) = t(U)$

S'_{UV}	1	2	3	4
1	12	5	7	15
2	7	12	14	22
3	5	-2	12	20
4	5	-2	12	20

$W(U,V)$	1	2	3	4
1	0	1	1	2
2	1	0	2	3
3	1	0	0	3
4	1	0	2	0

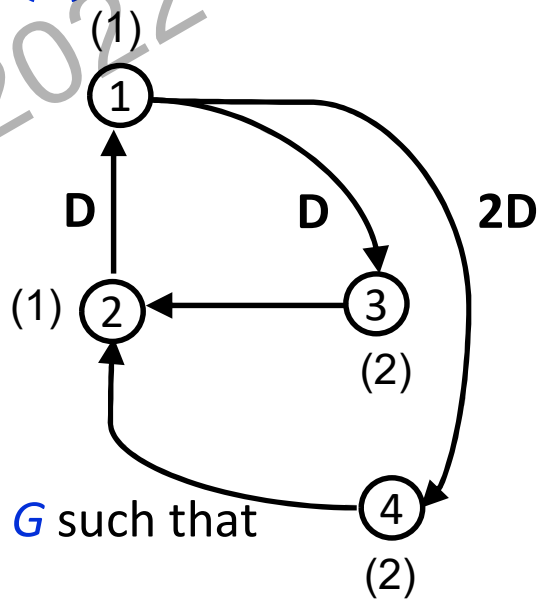
$D(U,V)$	1	2	3	4
1	1	4	3	3
2	2	1	4	4
3	4	3	2	6
4	4	3	6	2

Retiming for clock period minimization (4)

■ Feasibility constraints

$w'(e) \geq 0$ => Causality of the system => $w(e) \geq r(U) - r(V)$, $\forall e\ U \xrightarrow{e} V$

$$r(1) - r(3) \leq 1$$
$$r(1) - r(4) \leq 2$$
$$r(2) - r(1) \leq 1$$
$$r(3) - r(2) \leq 0$$
$$r(4) - r(2) \leq 0$$



■ Critical Path constraints

$r(U) - r(V) \leq W(U,V) - 1$ for all vertices *U* and *V* in *G* such that $D(U,V) > c$, where c = **target clock period** *(if $c = 3$)*

$$r(1) - r(2) \leq 0$$
$$r(2) - r(3) \leq 1$$
$$r(2) - r(4) \leq 2$$
$$r(3) - r(1) \leq 0$$
$$r(3) - r(4) \leq 2$$
$$r(4) - r(1) \leq 0$$
$$r(4) - r(3) \leq 1$$

Retiming for clock period minimization (c=3)

Using $W(U,V)$ and $D(U,V)$ the feasibility and critical path constraints are formulated to give certain inequalities.

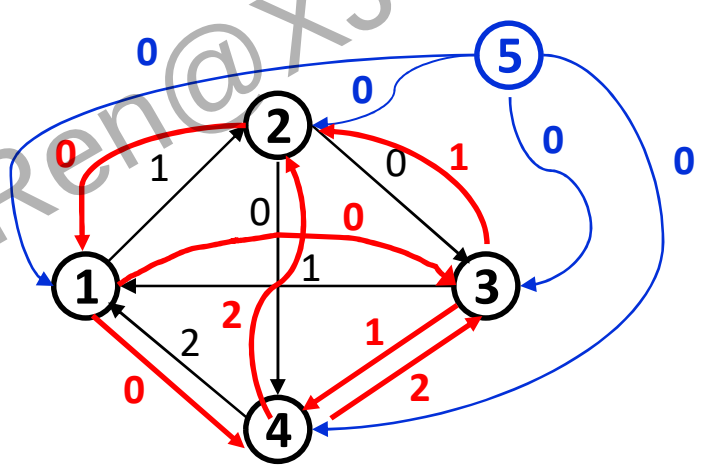
The inequalities are solved using ***constraint graphs*** and if a feasible solution is obtained then the circuit can be clocked with a period 'c'

■Feasibility constraints

$$r(1) - r(3) \leq 1$$
$$r(1) - r(4) \leq 2$$
$$r(2) - r(1) \leq 1$$
$$r(3) - r(2) \leq 0$$
$$r(4) - r(2) \leq 0$$

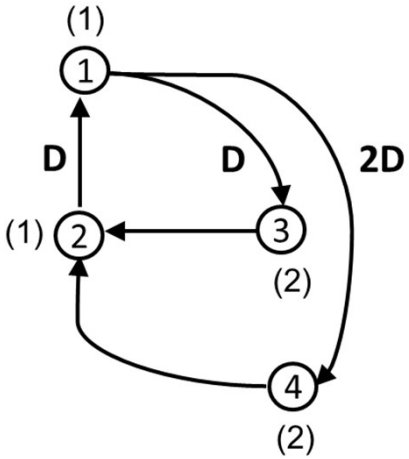
■ Critical Path constraints **(c=3)**

$$r(1) - r(2) \leq 0$$
$$r(2) - r(3) \leq 1$$
$$r(2) - r(4) \leq 2$$
$$r(3) - r(1) \leq 0$$
$$r(3) - r(4) \leq 2$$
$$r(4) - r(1) \leq 0$$
$$r(4) - r(3) \leq 1$$



Constraint graph

$$r(1) = r(2) = r(3) = r(4) = 0$$



Retiming for clock period minimization (c=2)

Using $W(U,V)$ and $D(U,V)$ the feasibility and critical path constraints are formulated to give certain inequalities.

The inequalities are solved using ***constraint graphs*** and if a feasible solution is obtained then the circuit can be clocked with a period 'c'

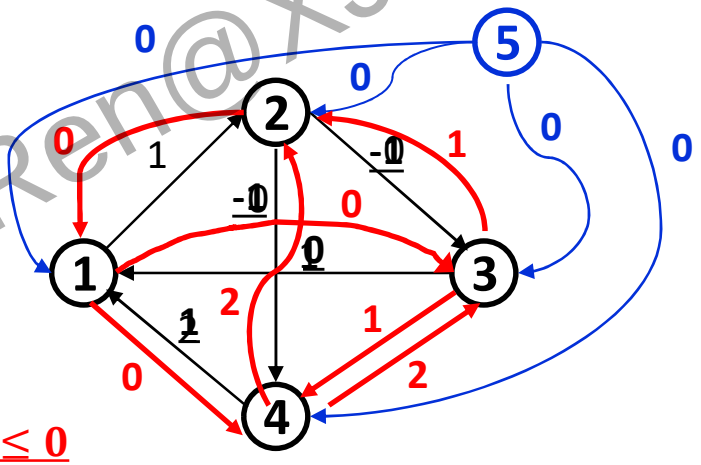
■ Feasibility constraints

$r(1) - r(3) \leq 1$
$r(1) - r(4) \leq 2$
$r(2) - r(1) \leq 1$
$r(3) - r(2) \leq 0$
$r(4) - r(2) \leq 0$

■ Critical Path constraints (c=2)

$r(1) - r(2) \leq 0$
$r(2) - r(3) \leq 1$
$r(2) - r(4) \leq 2$
$r(3) - r(1) \leq 0$
$r(3) - r(4) \leq 2$
$r(4) - r(1) \leq 0$
$r(4) - r(3) \leq 1$

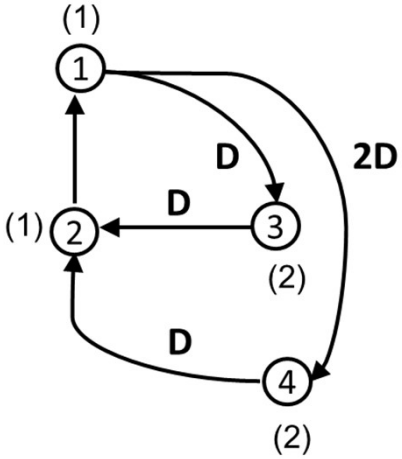
$r(1) - r(3) \leq 0$
$r(1) - r(4) \leq 1$
$r(3) - r(2) \leq -1$
$r(4) - r(2) \leq -1$



Constraint graph

$r(1) = r(3) = r(4) = -1, r(2) = 0$

$r(1) = r(3) = r(4) = 0, r(2) = 1$

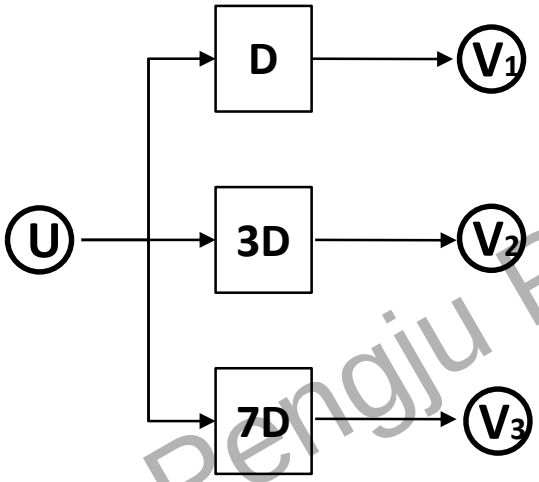


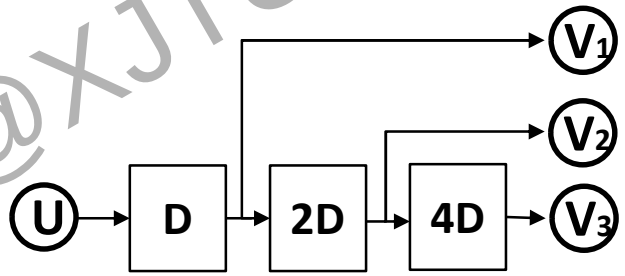
General Approach（Retiming for clock period minimization）

1. Compute $W(U,V)$ and $D(U,V)$
2. Sort the values of $D(U,V)$
3. Perform *binary search* on these values to find the retiming solution with the minimum clock period.

Retiming for Register Minimization(1)

If a node has several output edges carrying the same signal, the number of registers required to implement these edges is the maximum number of registers on any one of the edges.





$$R_U = \max_{U \to ?} \{w_r(e)\}$$

$$R_U = 7$$

Retiming for Register Minimization(2)

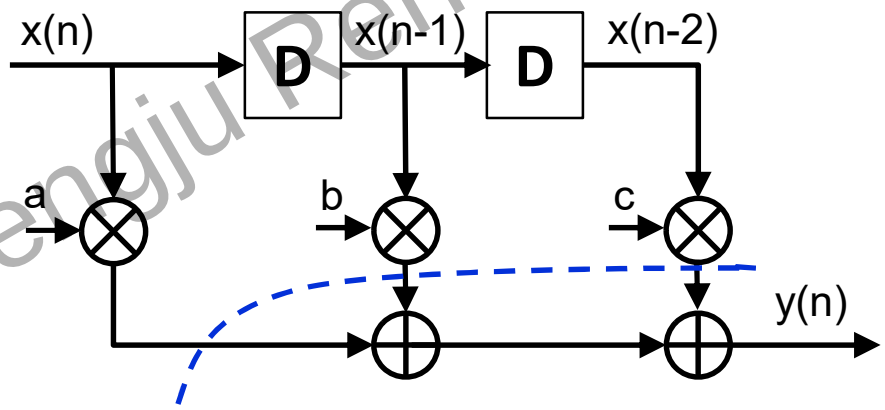
Minimize $COST = \sum R_X$ subject to

1. (fanout constraint) $R_X \geq w_r(e)$ for all X and all edges $X \xrightarrow{e} ?$
2. (feasibility constraint) $r(U) - r(V) \leq w(e)$ for every edge $U \xrightarrow{e} V$
3. (clock period constraint) $r(U) - r(V) \leq W(U,V) - 1$ for all vertices U, V such that $D(U,V) > c$.

Retiming and Pipelining

Pipelining is a special case of cutset retiming where there are no edges in the cutset from G2 to G1, i.e., pipelining applies to graphs without loops

Pipelining is Equivalent to Introducing Many delays at the Input (or output) followed by Retiming



Next Lecture：Parallel Architecture &

Resource Sharing