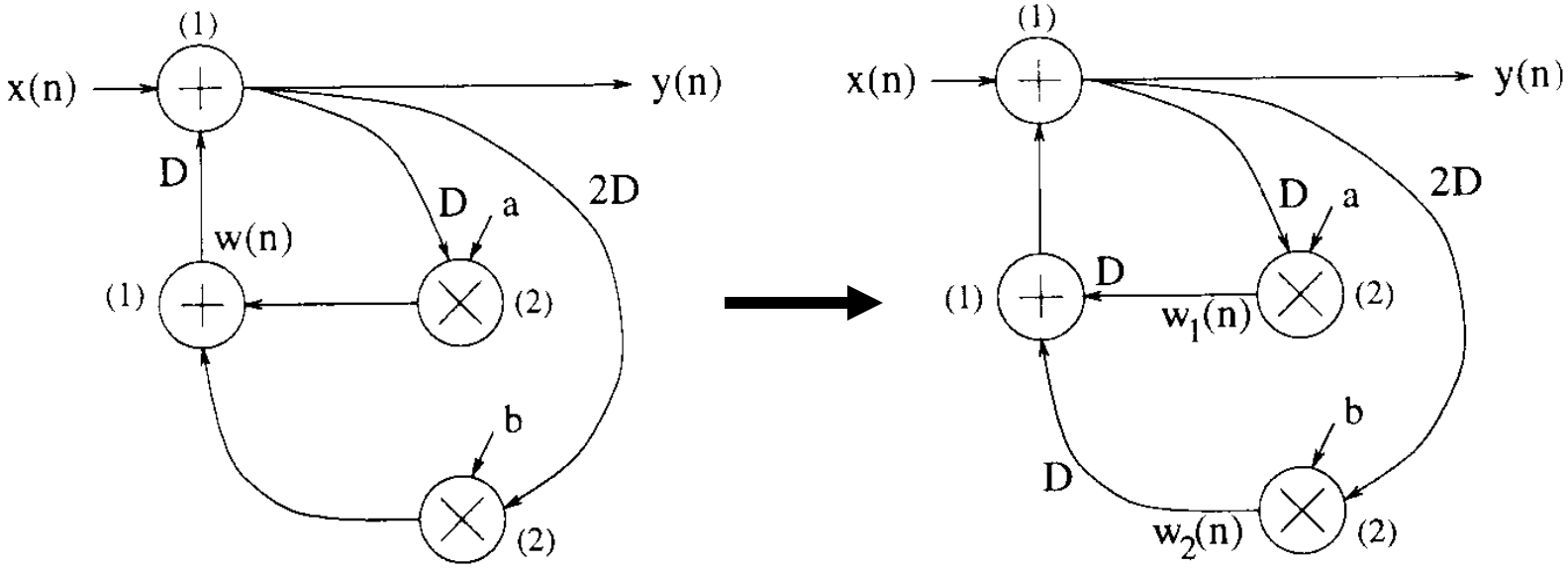
臺灣大學

Retiming

Shao-Yi Chien

Introduction (1/2)

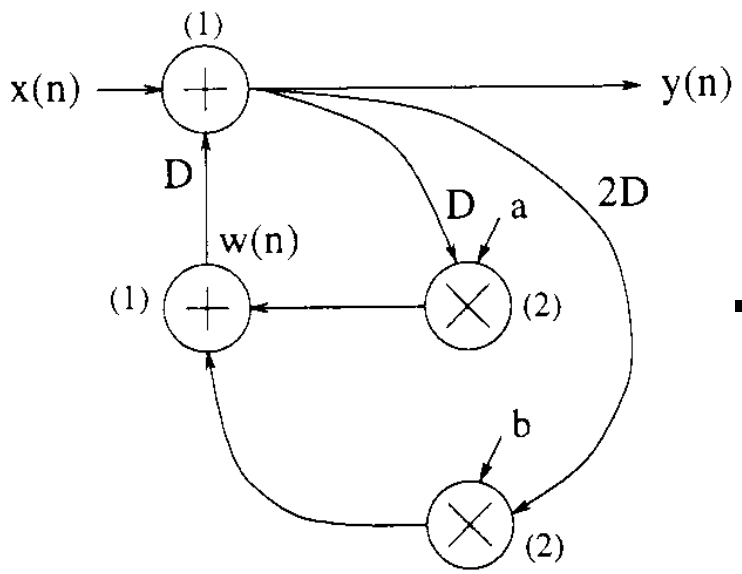
- Retiming
  - A transformation technique used to **change the locations of delay elements** in circuit without affecting the input/output characteristics

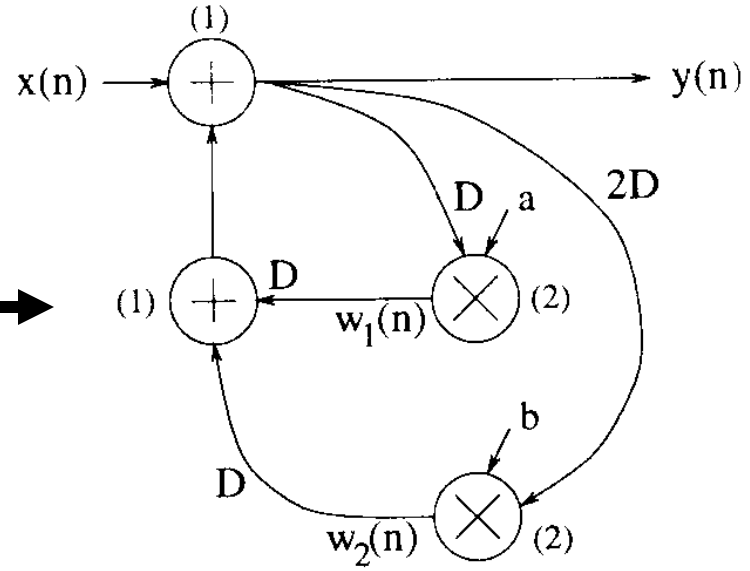


Introduction (2/2)

- Applications of retiming
  - Reducing the clock period
  - Reducing the number of registers
  - Reducing the power consumption
  - Logic synthesis

Reducing the Clock Period

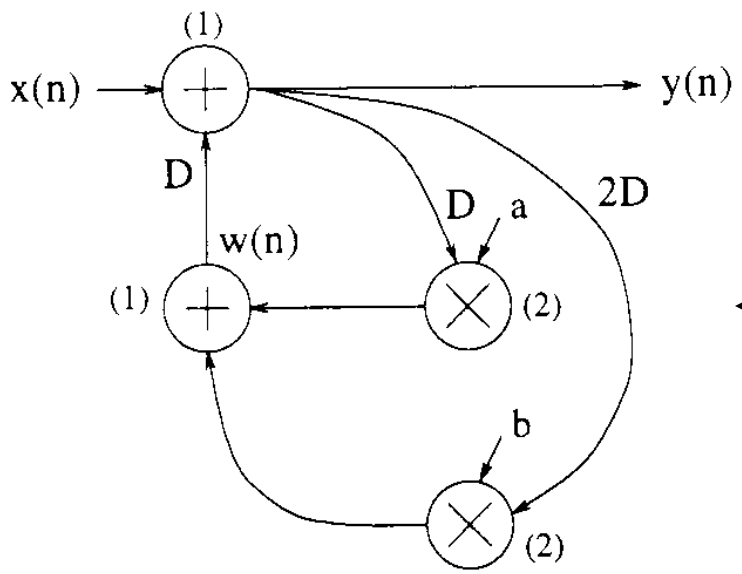


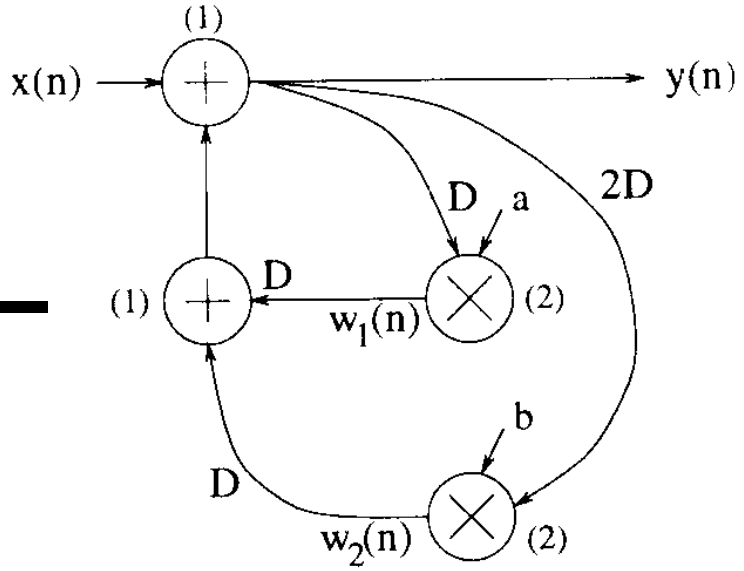


Critical path=3u.t.
Min. clock period=3u.t.

Critical path=2u.t.
Min. clock period=2u.t.

Reducing the Number of Registers



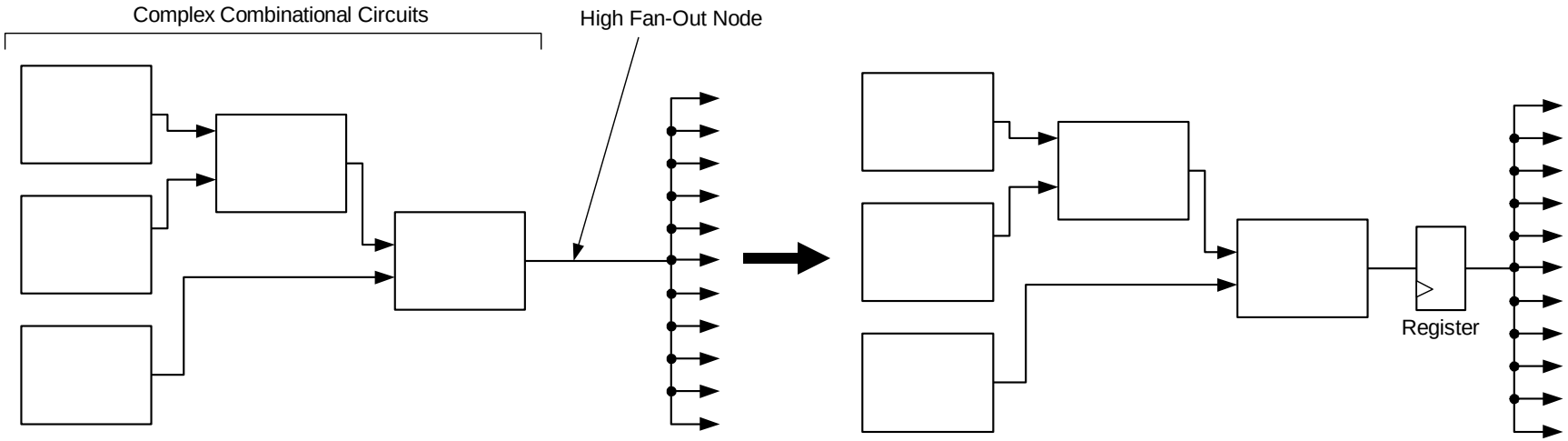


Number of registers: 4

Number of registers: 5

Reducing the Power Consumption

- Placing registers at the inputs of nodes with large capacitances can reduce the switching activities at these nodes



Large power consumption due to glitch

Lower power consumption

Quantitative Description of Retiming

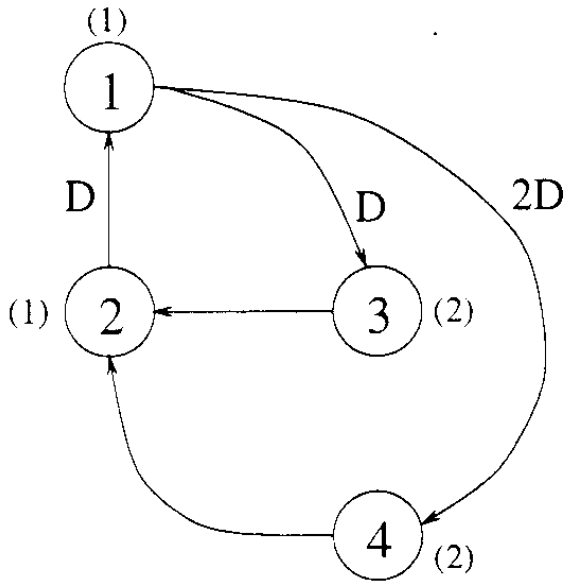
- Map circuit G→G_r
- Retiming can be presented with r(V), V is one of the nodes in the circuit
- For an edge $U \xrightarrow{e} V$

Destination Source

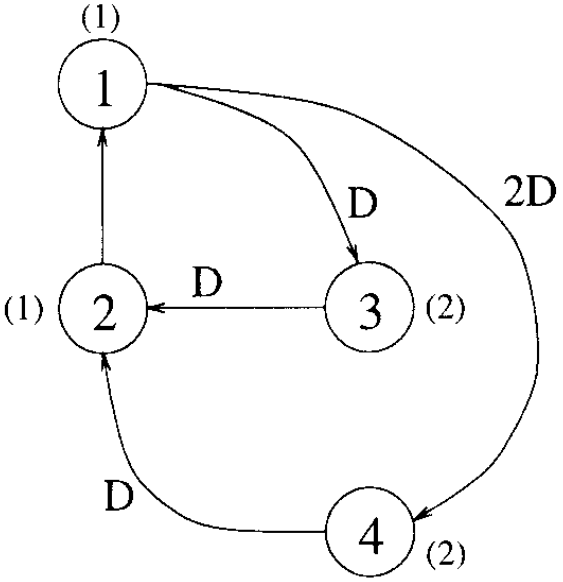
$$w_r(e) = w(e) + r(V) - r(U)$$

  - w(e): weight (delay) of the edge e in the origin circuit
  - w_r(e): weight of the edge e in the retimed circuit

An Example (1/2)



Origin DFG



Retimed DFG with
r(1)=0
r(2)=1
r(3)=0
r(4)=0

An Example (2/2)

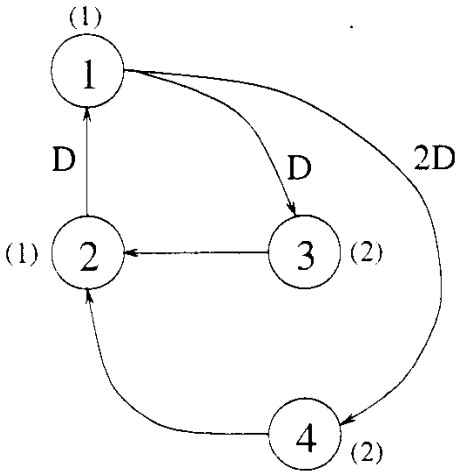
$2 \to 1: \ 1 + r(1) - r(2) \geq 0$

$1 \to 3: \ 1 + r(3) - r(1) \geq 0$

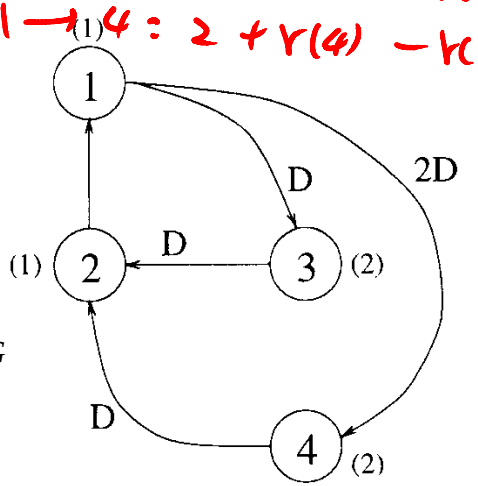
$3 \to 2: \ 0 + r(2) - r(3) \geq 0$

$4 \to 2: \ 0 + r(2) - r(4) \geq 0$

$1 \to 4: \ 2 + r(4) - r(1) \geq 0$



$w_r(e) \geq 0 \quad \forall e \in G$



$$w_r(2 \xrightarrow{e} 1) = w(2 \xrightarrow{e} 1) + r(1) - r(2)$$
$$= 1 + 0 - 1 = 0$$

- A retiming solution is feasible if

$$w_r(e) \geq 0 \quad \forall e \in G$$

Properties of Retiming (1/2)

- The weight of the retimed path

$p = V_0 \xrightarrow{e_0} V_1 \xrightarrow{e_1} \cdots \xrightarrow{e_{k-1}} V_k$ is given by

$$w_r(p) = w(p) + r(V_k) - r(V_0)$$

  - Prof:

$$\begin{aligned} w_r(p) &= \sum_{i=0}^{k-1} w_r(e_i) \\ &= \sum_{i=0}^{k-1} (w(e_i) + r(V_{i+1}) - r(V_i)) \\ &= \sum_{i=0}^{k-1} w(e_i) + \left(\sum_{i=0}^{k-1} r(V_{i+1}) - \sum_{i=0}^{k-1} r(V_i) \right) \\ &= w(p) + r(V_k) - r(V_0). \end{aligned}$$

Properties of Retiming (2/2)

- 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

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

Solving Systems of Inequalities (1/3)

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

Solving Systems of Inequalities (2/3)

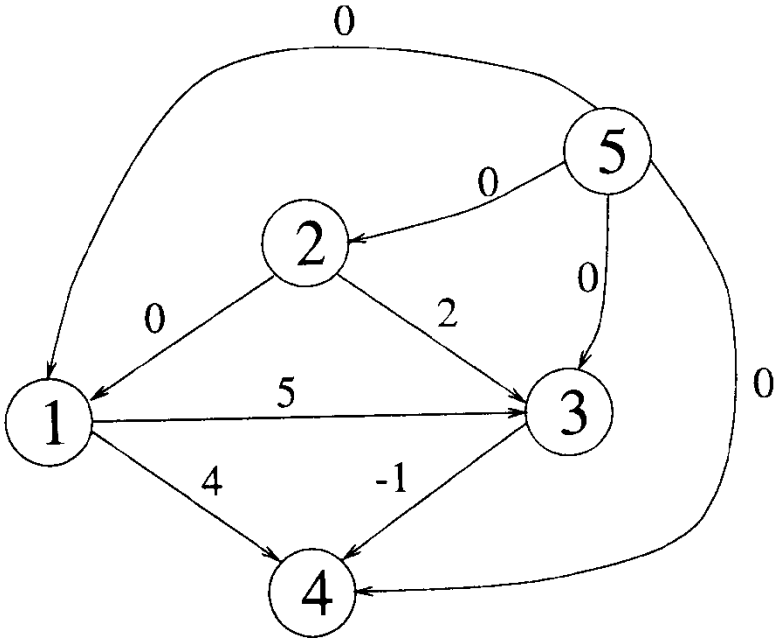
- Step 1: draw a constraint graph
  - Draw the node i for each of the N variables r_i, i=1,2,…,N
  - Draw the node N+1
  - For each inequality $r_i - r_j <= k$, draw the edge j→i from the node j to node i with length k
  - For each node i, i=1,2,…n, draw the edge N+1→i from the node N+1 to the node i with length 0

Solving Systems of Inequalities (3/3)

- 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 r_i is the minimum-length path from the node N+1 to the node i

Example

$$
\begin{aligned}
r_1 - r_2 &\leq 0 \\
r_3 - r_1 &\leq 5 \\
r_4 - r_1 &\leq 4 \\
r_4 - r_3 &\leq -1 \\
r_3 - r_2 &\leq 2.
\end{aligned}
$$



Bellman-Ford shortest path algorithm:

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

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

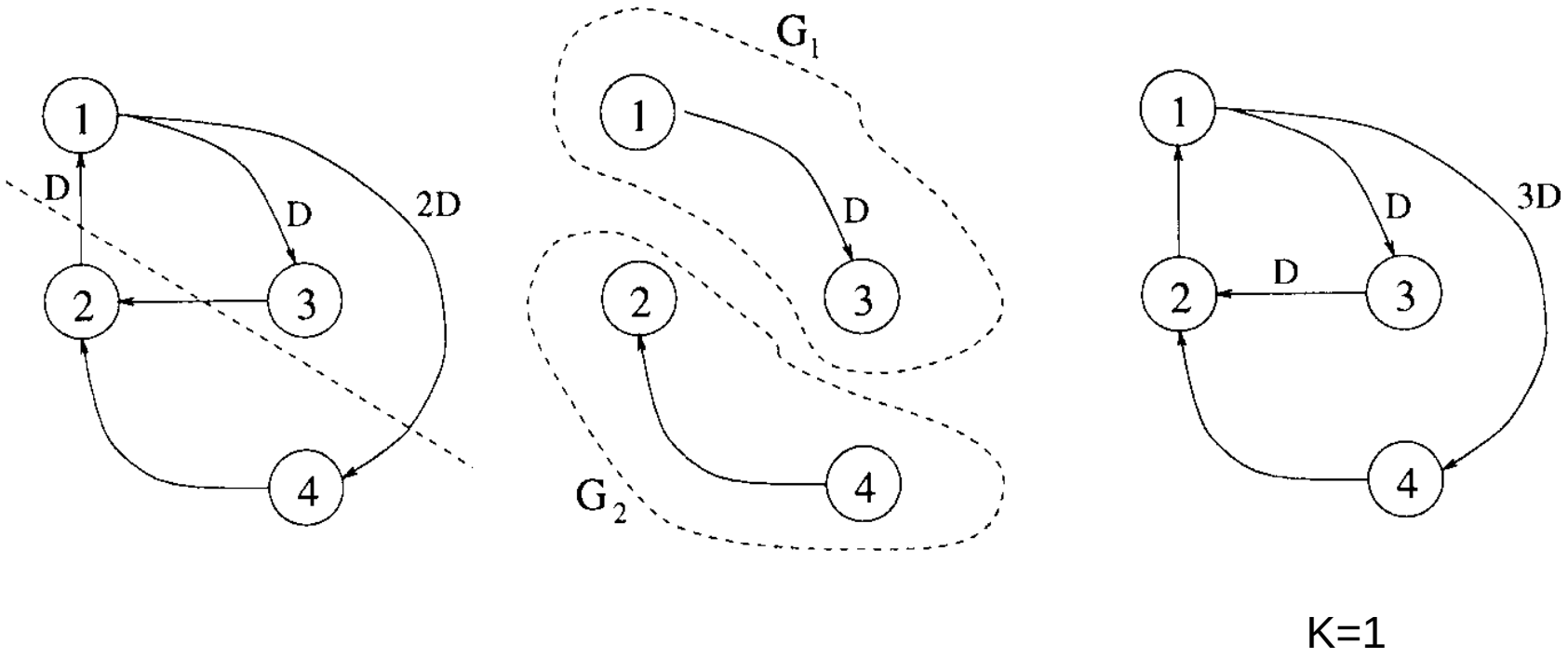
Retiming Techniques

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

Cutset Retiming

- A special case of retiming that only affects the weights of the edges in the cutset
- For the disconnected subgraph G1 and G2
  - Adding k delays to each edge from G1 to G2
  - Removing k delays from each edge from G2 to G1

An Example of Cutset Retiming



Feasibility of Cutset Retiming

- For each edge from G1 to G2

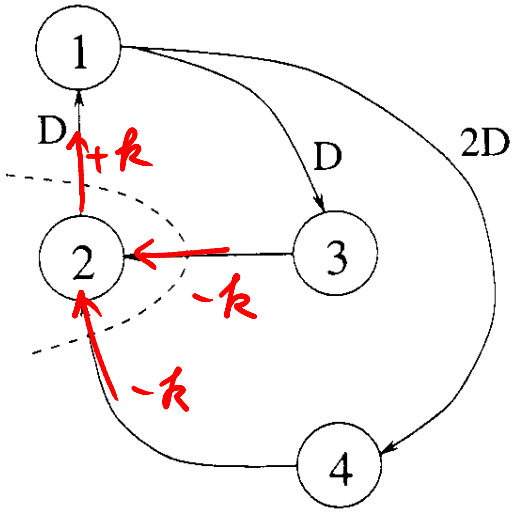
$$w_r(e_{1,2}) \geq 0 \Rightarrow w(e_{1,2}) + k \geq 0$$

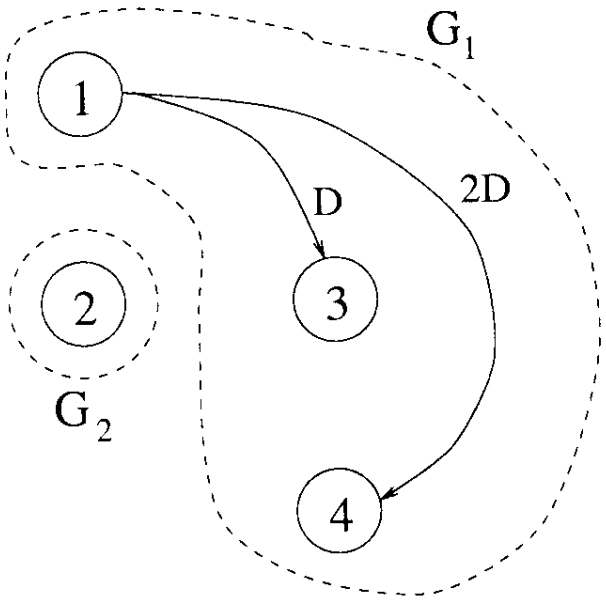
- For each edge from G2 to G1

$$w_r(e_{2,1}) \geq 0 \Rightarrow w(e_{2,1}) - k \geq 0$$

$$-\min_{G_1 \xrightarrow{e} G_2} \{w(e)\} \leq k \leq \min_{G_2 \xrightarrow{e} G_1} \{w(e)\}$$

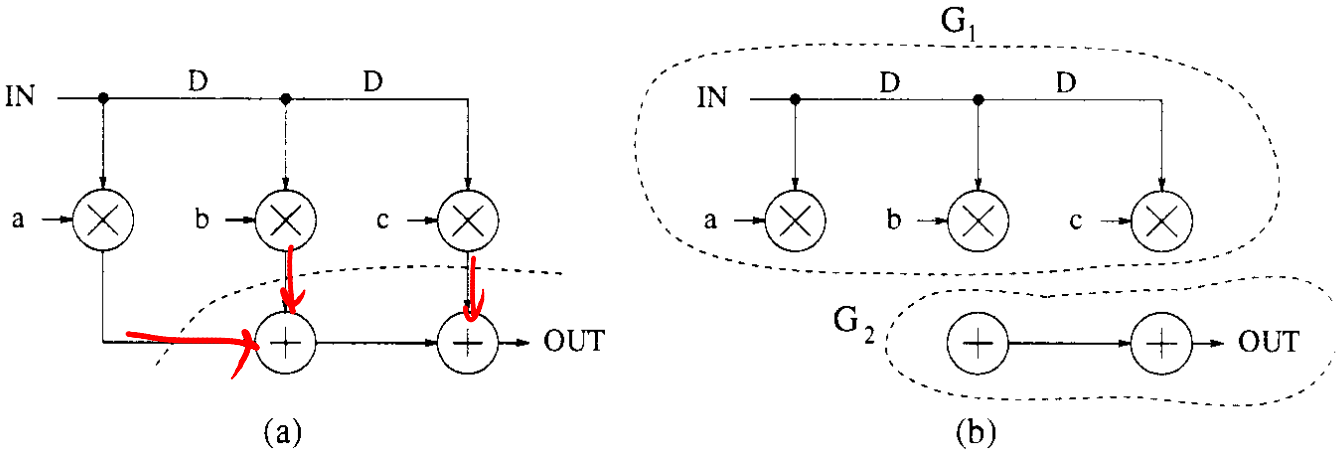
Special Case of Cutset Retiming: Single Node Cutset

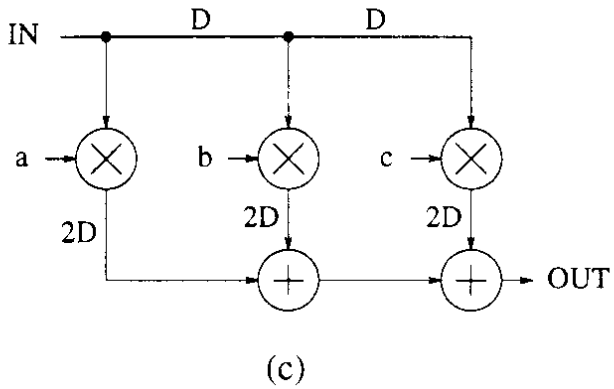




- Choose a node as a cutset
- Substract one delay from each edge outgoing from the node
- Add one delay from each edge incident into the node

Special Case of Cutset Retiming: Pipelining



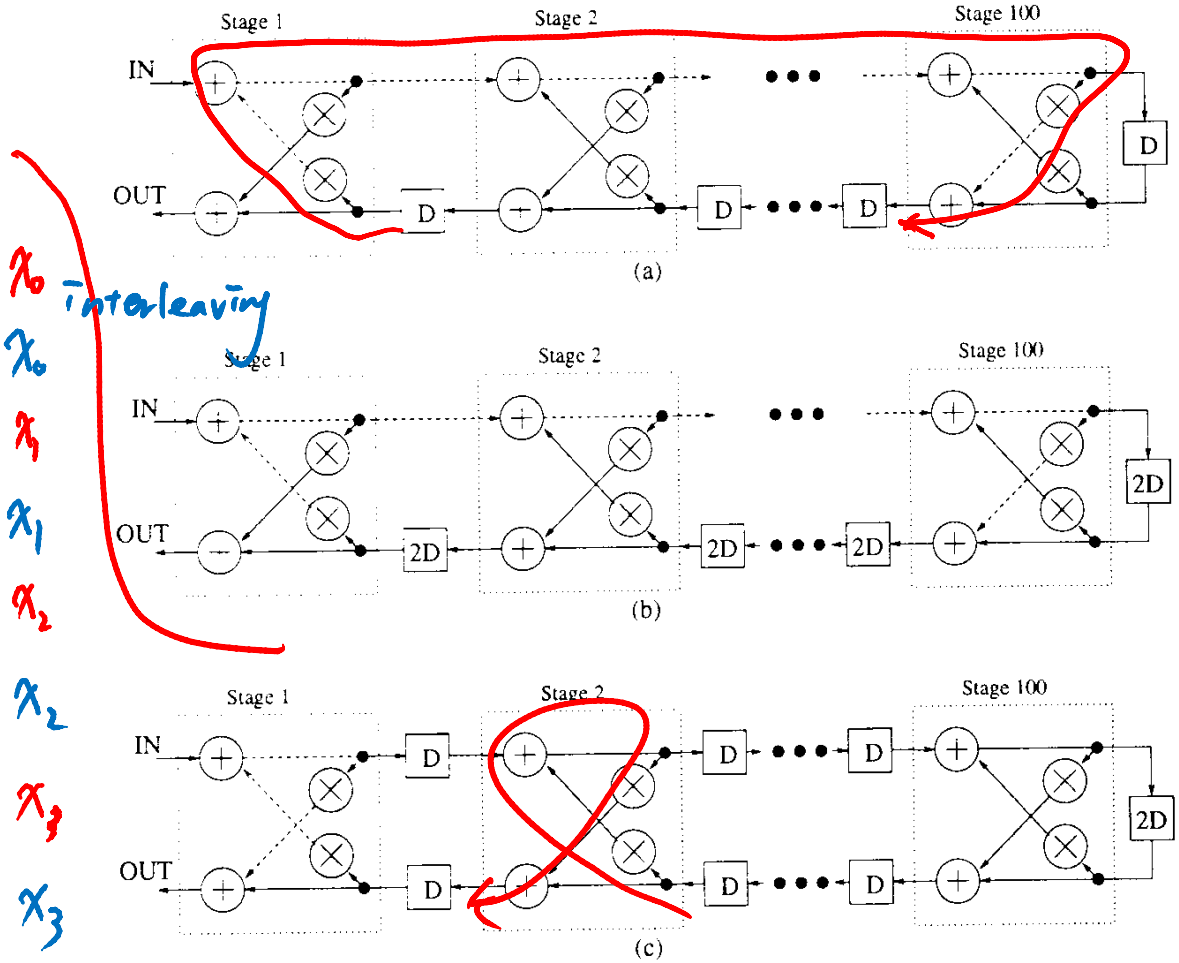


K=2

Special Case: Combining with Slow-Down (1/2)

- Create N-slow version of the DFG first
  - Replace each delay element with N delays
- In an N-slow system, N-1 null operations (or 0 samples) must be interleaved after each useful signal sample to preserve the functionality

Special Case: Combining with Slow-Down (2/2)



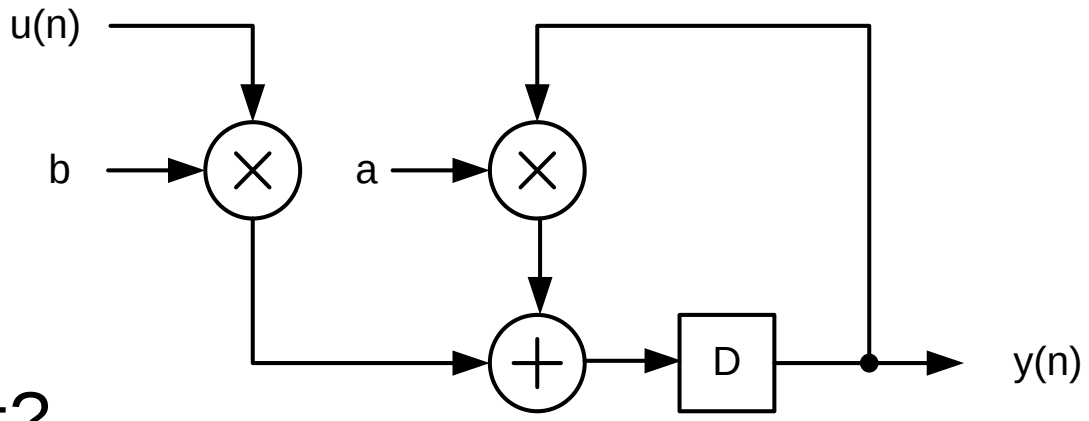
Assume addition: 1 u.t., multiplication: 2 u.t. Critical path is 105 u.t. Minimum sample period is 105 u.t.

2-slow version

Retimed version. The critical path is 6 u.t. The minumum sample period is 12 u.t.

Example: Reduce the Critical Path of a Recursive DFG

- For the IIR filter y(n+1)=ay(n)+bu(n)
  T_M=3u.t., T_A=1u.t.

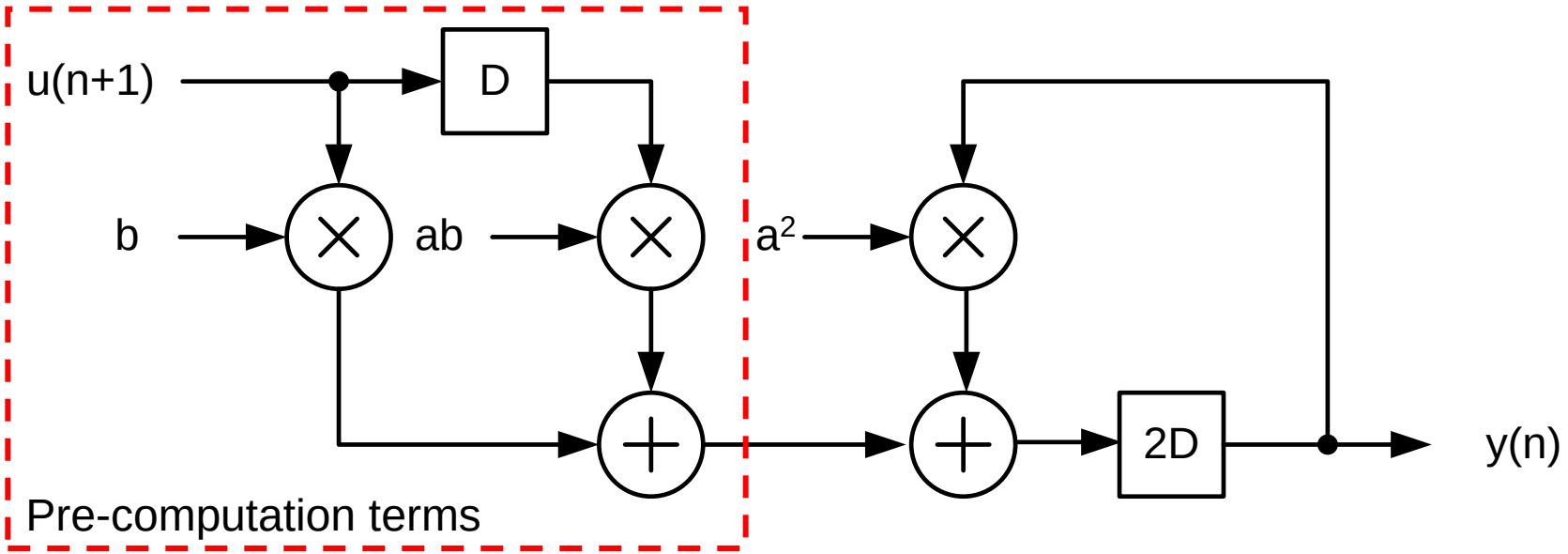


- Critical path=?
- Iteration bound=?
- Can we reduce the sampling period to 2u.t.?

Example: Reduce the Critical Path of a Recursive DFG

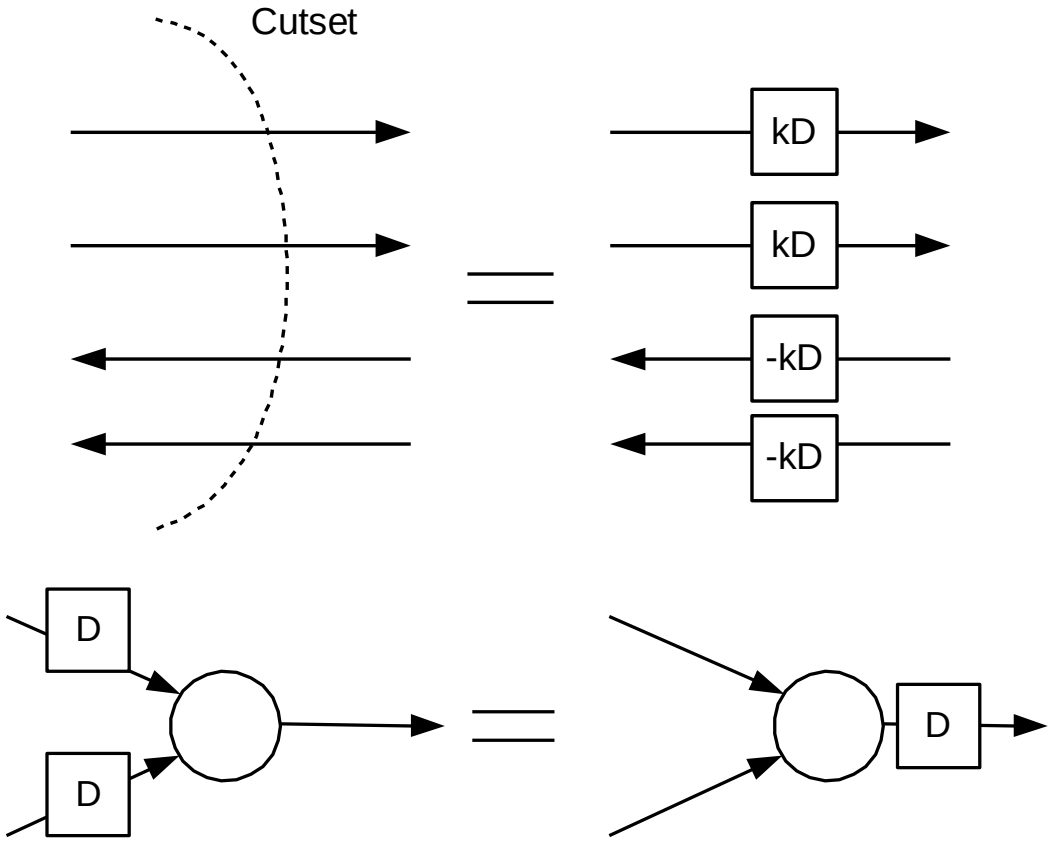
- Employ **look-ahead transformation**
  - Consider more than one iterations
  - $y(n+2) = ay(n+1) + bu(n+1)$
    $= a[ay(n)+bu(n)] + bu(n+1)$
    $= a^2y(n)+abu(n)+bu(n+1)$

Example: Reduce the Critical Path of a Recursive DFG



- Critical path=?
- Iteration bound=?
- Can we reduce the sampling period to 2u.t.?

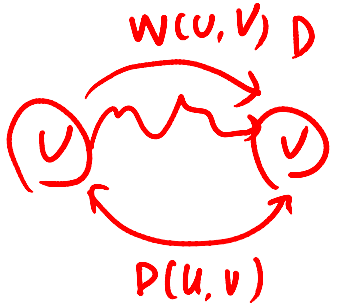
Remarks



Retiming for Clock Period Minimization (1/7)

- Minimum feasible clock period or critical path

$$\Phi(G) = \max\{t(p) : w(p) = 0\}.$$



- Define two quantities, U→V
  - Minimum number of registers of U→V

$$W(U,V) = \min\{w(p) : U \overset{p}{\rightsquigarrow} V\}$$

  - Maximum computation time of U→V

$$D(U,V) = \max\{t(p) : U \overset{p}{\rightsquigarrow} V \text{ and } w(p) = W(U,V)\}$$

Retiming for Clock Period Minimization (2/7)

$S_{uv}' = \min\{\sum w'(e)\} = \min\{\sum(Mw(e) - t(u))\}$

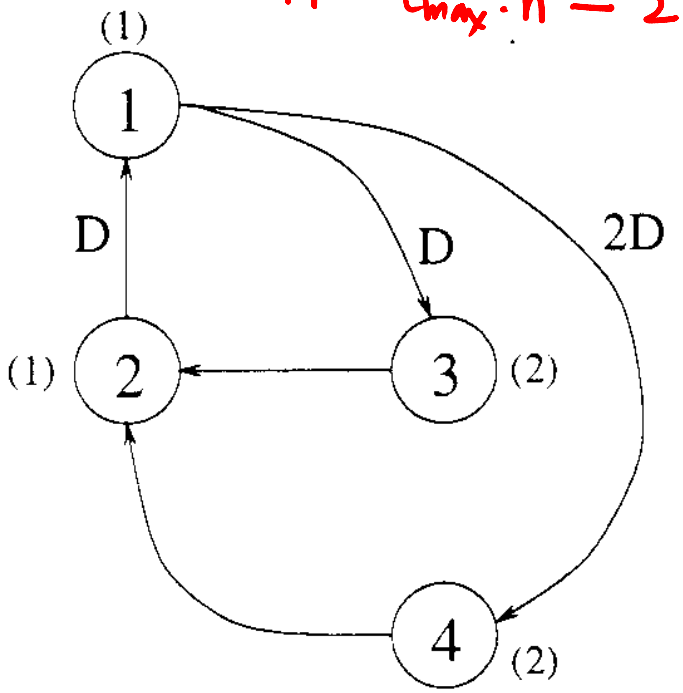
- Method to compute W(U,V) and D(U,V)

1. Let $M = t_{max}n$, where t_{max} is the maximum computation time of the nodes in G and n is the number 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 $U \xrightarrow{e} V$.
3. Solve the all-pairs shortest path problem on G'. Let S'_{UV} be the shortest path from U to V.
4. If $U \neq V$, then $W(U,V) = \left\lceil \frac{S'_{UV}}{M} \right\rceil$ and $D(U,V) = MW(U,V) - S'_{UV} + t(V)$. If $U = V$, then $W(U,V) = 0$ and $D(U,V) = t(U)$.

$\left\lceil \frac{S_{uv}}{M} \right\rceil = \left\lceil \frac{\sum(Mw(e) - t(u))}{M} \right\rceil = \left\lceil \sum\left(w(e) - \frac{t(u)}{M}\right) \right\rceil = \left\lceil \sum w(e) - \sum \frac{t(u)}{M} \right\rceil = \sum w(e)$

Retiming for Clock Period Minimization (3/7)

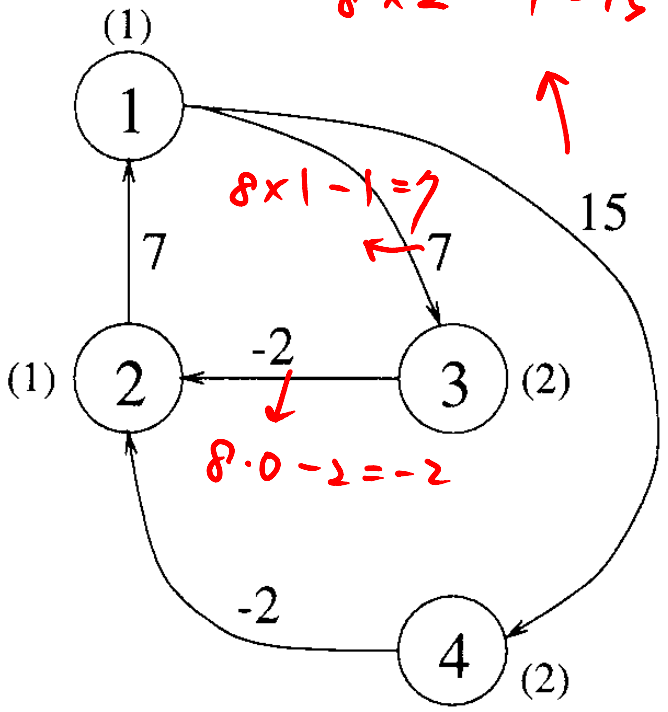
$M = t_{max} \cdot n = 2 \cdot 4 = 8$



G

$M w(e) - t(u)$

$8 \times 2 - 1 = 15$



G'

Retiming for Clock Period Minimization (4/7)

$W(U,V) = \left\lceil \frac{S'_{UV}}{M} \right\rceil$

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

$\left\lceil \frac{15}{8} \right\rceil = 2$

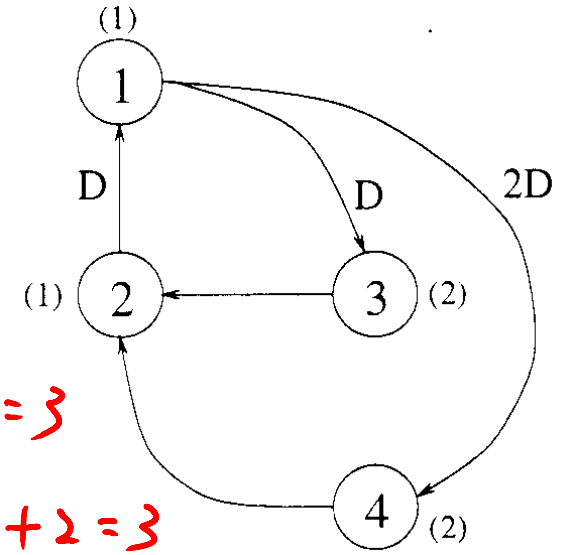
$D(U,V)$	1	2	3	4
1	1	4	3	3
2	2	1	4	4
3	4	3	2	6
4	4	3	6	2

$D(U,V) = M\,W(U,V) - S'_{UV} + t(V)$

$\left\lceil \frac{20}{8} \right\rceil = 3$

$8 \times 2 - 15 + 2 = 3$

$8 \times 3 - 20 + 2 = 6$



Retiming for Clock Period Minimization (5/7)

- Constraints

If the desired clock period is *c*

$w_r(e) \geq 0$

1. (feasibility constraint) $r(U) - r(V) \leq w(e)$ for every edge $U \xrightarrow{e} V$ of G. and

2. (critical path constraint) $r(U) - r(V) \leq W(U,V) - 1$ for all vertices U, V in G such that $D(U,V) > c$.

$W(u,v) + r(v) - r(u) \geq 1$

Retiming for Clock Period Minimization (6/7)

- If c=3

$$
\begin{aligned}
r(1) - r(3) &\leq 1 \\
r(1) - r(4) &\leq 2 \\
r(2) - r(1) &\leq 1 \\
r(3) - r(2) &\leq 0 \\
r(4) - r(2) &\leq 0,
\end{aligned}
$$

Feasibility constraints

$$
\begin{aligned}
r(1) - r(2) &\leq 0 \\
r(2) - r(3) &\leq 1 \\
r(2) - r(4) &\leq 2 \\
r(3) - r(1) &\leq 0 \\
r(3) - r(4) &\leq 2 \\
r(4) - r(1) &\leq 0 \\
r(4) - r(3) &\leq 1.
\end{aligned}
$$

Critical path constraints

C = 2 ?

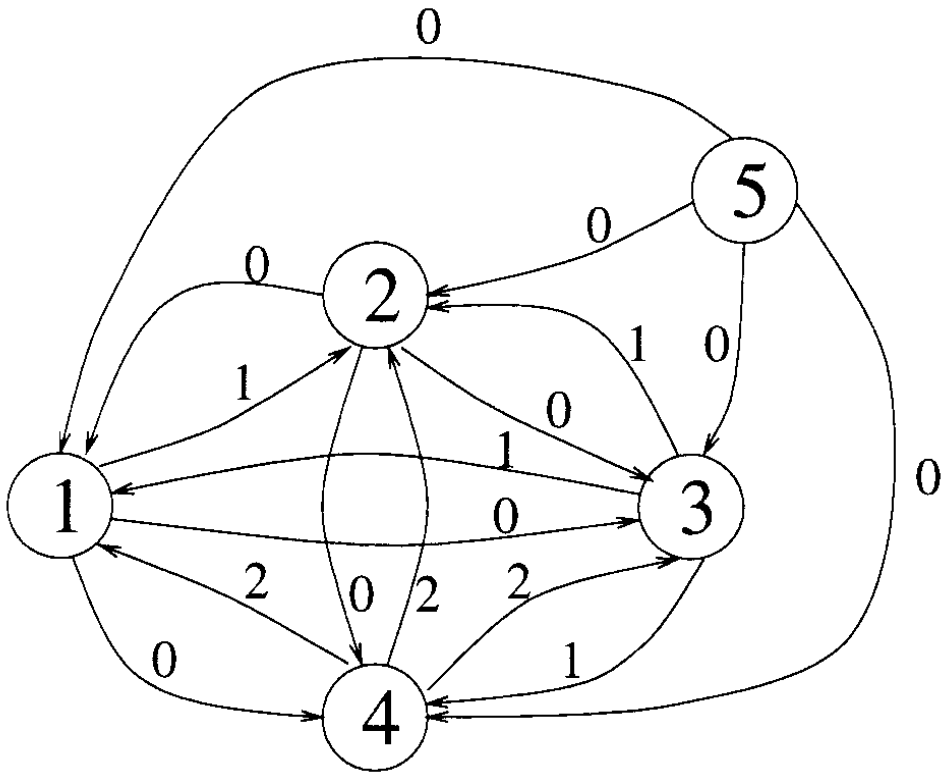
$r(1) - r(3) \leq 0$

$r(1) - r(4) \leq 1$

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

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

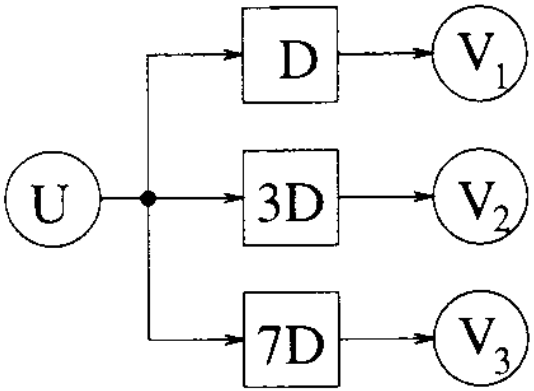
Retiming for Clock Period Minimization (7/7)

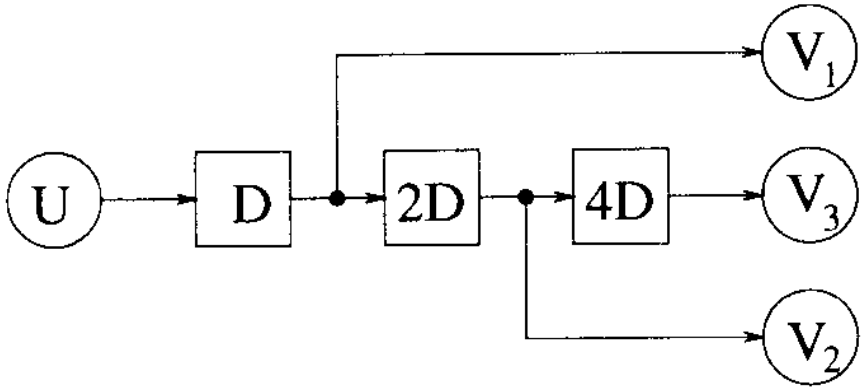


How about c=2?

Constraint graph → r(1)=r(2)=r(3)=r(4)=0

Retiming for Register Minimization (1/2)





$$R_V = \max_{V \xrightarrow{e} ?} \{w_r(e)\}$$

Ex: R_U=7

$$COST = \sum R_V$$

Retiming for Register Minimization (2/2)

- Minimize $COST = \sum R_V$ subject to

1. (fanout constraint) $R_V \geq w_r(e)$ for all V and all edges $V \xrightarrow{e} ?$.
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 $D(U,V) > c$.