Synthesis of Digital Systems

COL 719

Part 4: Retiming

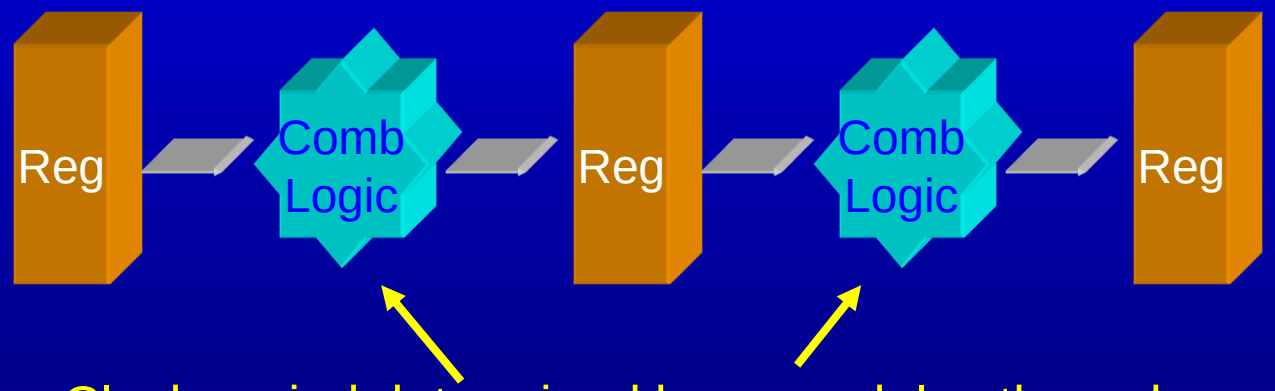
Instructor: Preeti Ranjan Panda

Department of Computer Science and Engineering

Indian Institute of Technology Delhi

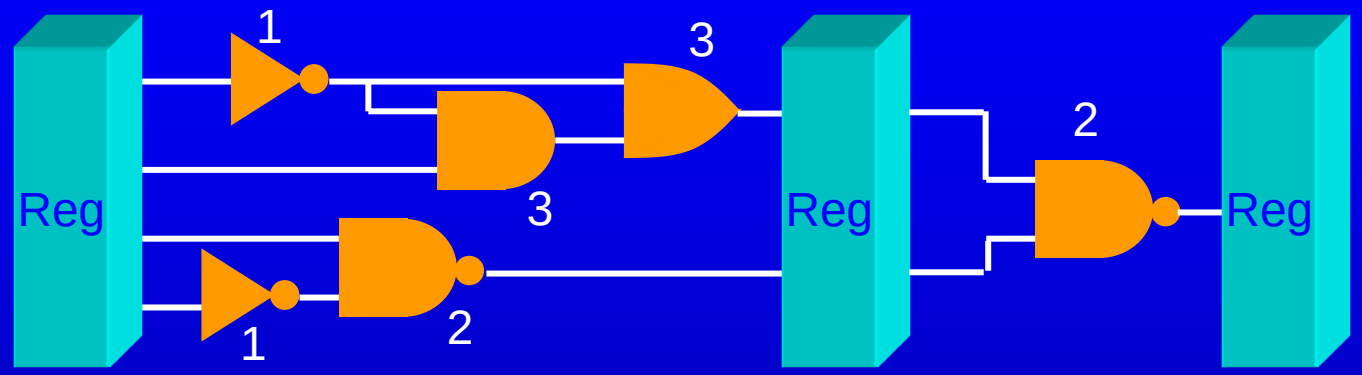
Optimising Sequential Circuits

- Minimise area and cycle time
  - Optimise combinational parts independently
  - Retiming by moving registers



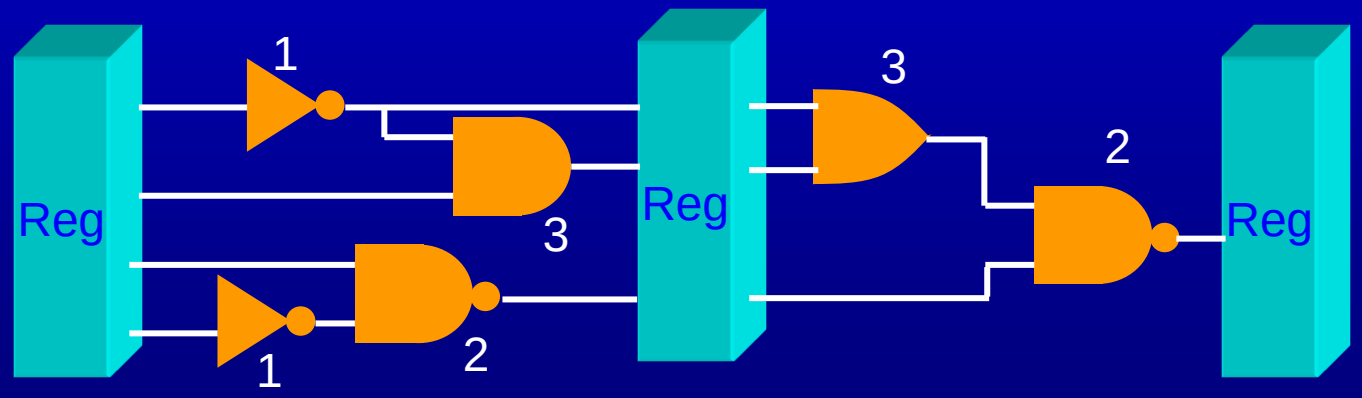
Clock period determined by max. delay through combinational logic stages

Reducing Cycle Time by Retiming



Cycle time = 7

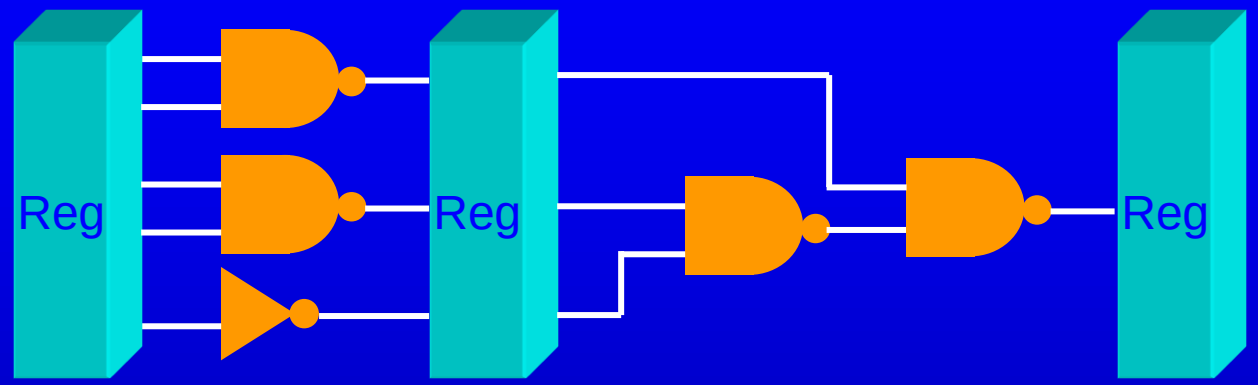
Retime: move location of register



Cycle time = 5

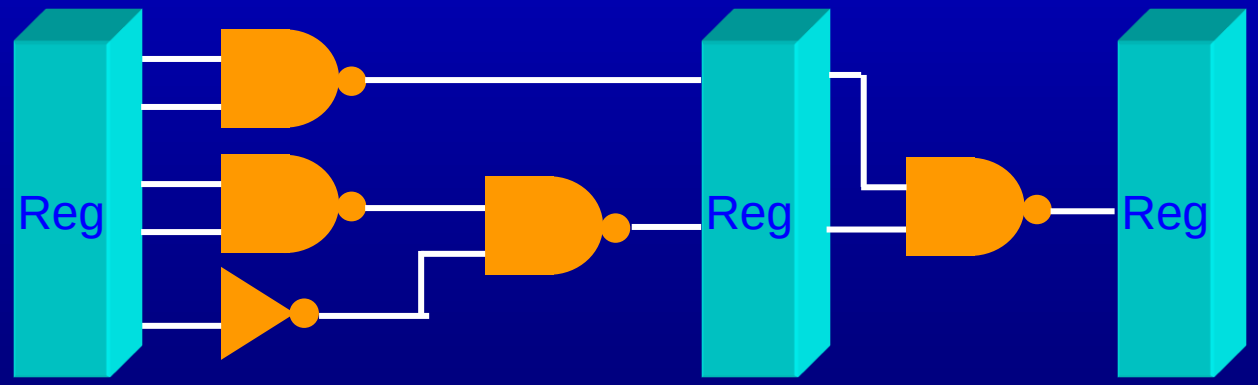
Moving register position leads to balancing of delays

Reducing Area by Retiming



Number of FFs:
5+3+1 = 9

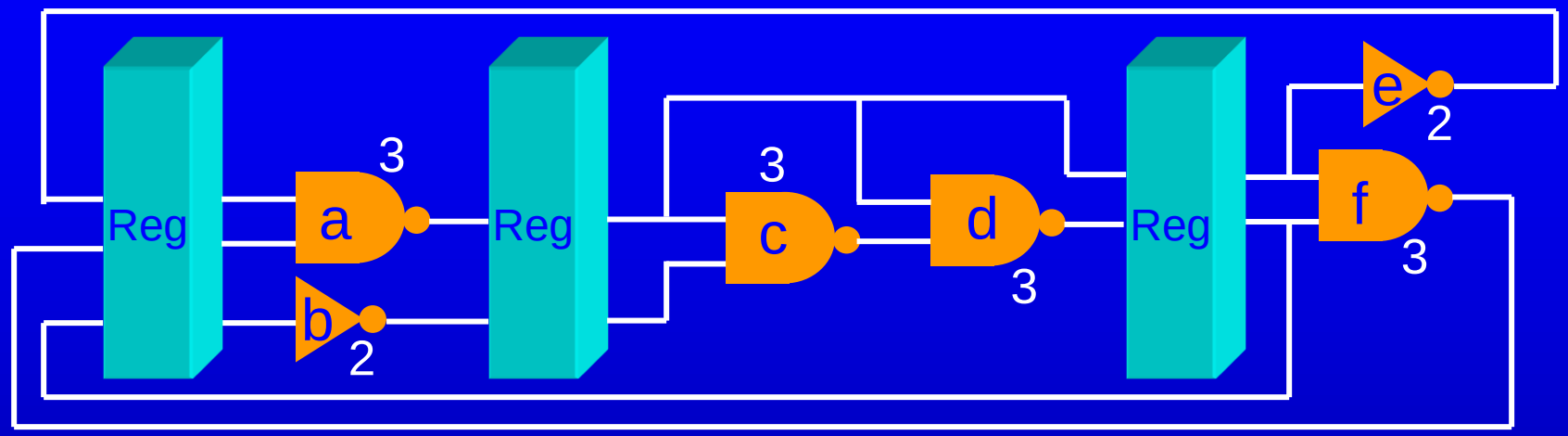
Retime: move location of register
External behaviour unchanged



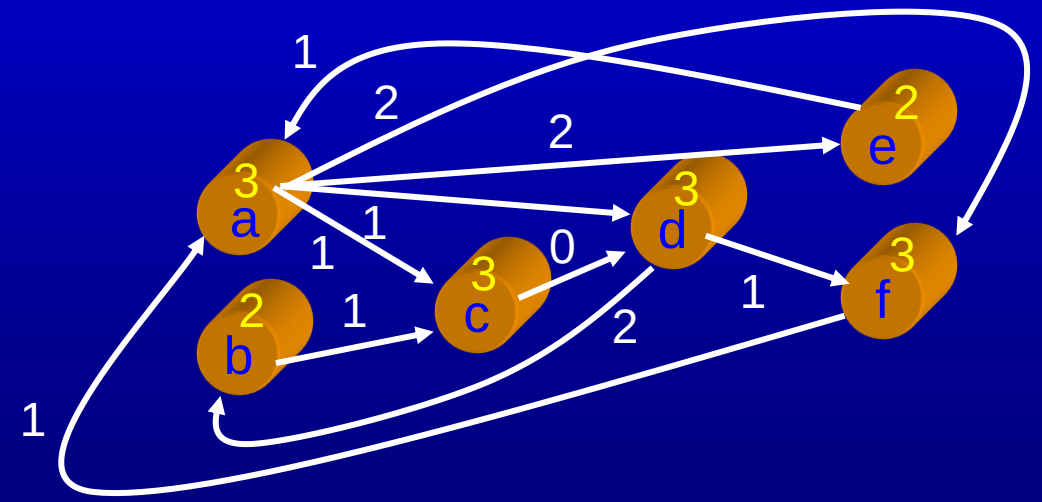
Number of FFs:
5+2+1 = 8

Moving register position leads to reduced FF count (cycle time unchanged)

Graph Representation of Sequential Circuits for Retiming



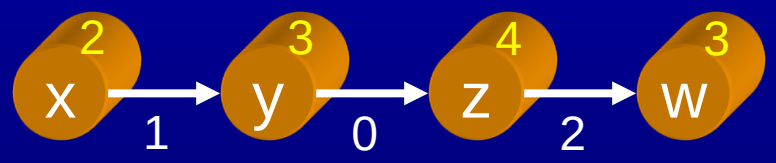
- Nodes = Gates
  - weight = gate delay
- Edges = Registers
  - weight = #Registers between gates



Path Delay between Nodes

- Sum of propagation delays of nodes along path (including end-points)
- Let d_k be path delay of node v_k
- For path $(v_i,\ldots,v_j)$, path delay:

$$d(v_i,\ldots,v_j) = \Sigma_{vk\in(vi,\ldots,vj)} d_k$$



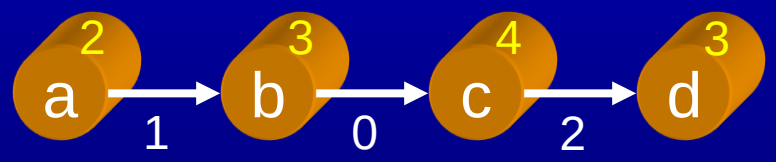
d (x,...,w) = 2+3+4+3 = 12

Path delay is independent of registers

Path Weight between Nodes

- Register count along path (including end-points)
- Let w_{kl} be weight of edge $v_k \rightarrow v_l$
- For path $(v_i,\ldots,v_j)$, path weight:

$$w(v_i,\ldots,v_j) = \Sigma_{vk \rightarrow vl \in (vi,\ldots,vj)} w_{kl}$$

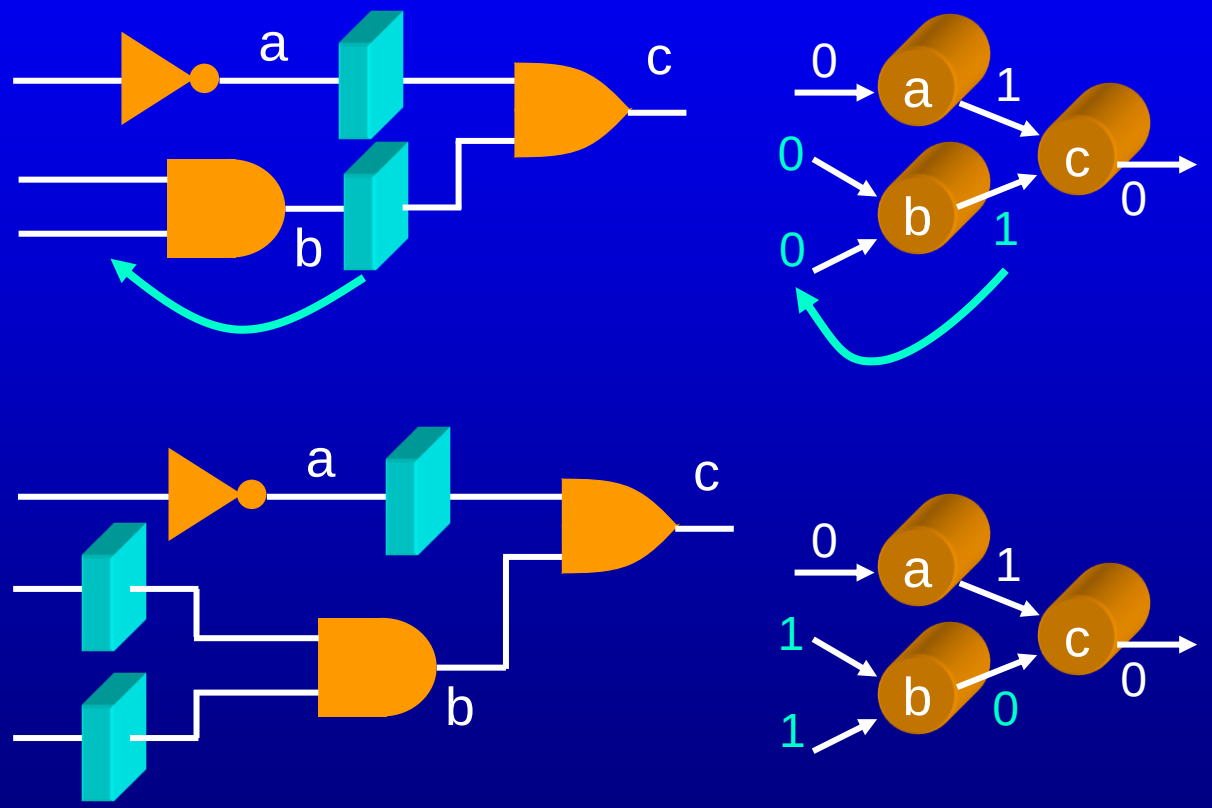


w (a,...,d) = 1+0+2 = 3

Path weight is independent of node delays

Retiming a Node

- Moving registers from outputs to inputs
  - or vice versa
  - amount of synchronous delay moved past node
  - “register” = “flip flop”
- Positive or negative value

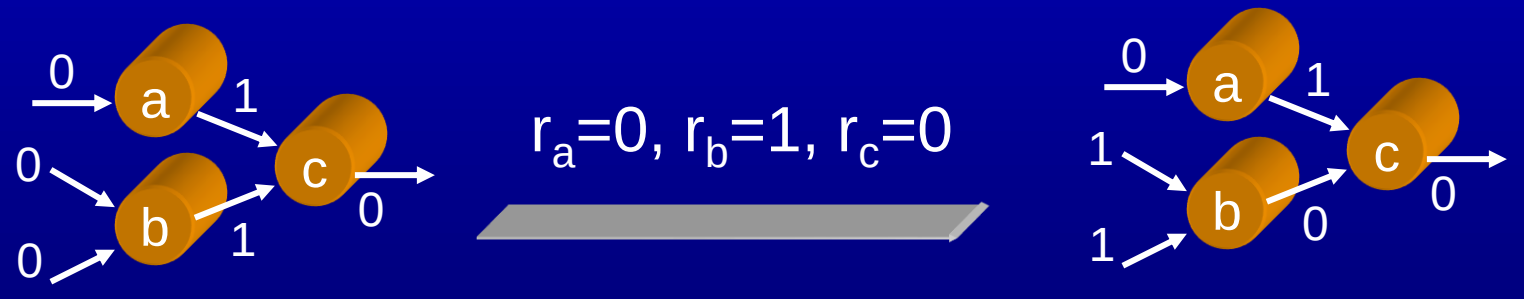


Retiming the Graph

- Graph transformation: G(V,E,W) to G'(V,E,W')
- Integer vector $r:V \rightarrow Z$
- For all edges $v_i \rightarrow v_j$

$$w'_{ij} = w_{ij} + r_j - r_i$$

r_j registers added to edge
r_i registers removed from edge

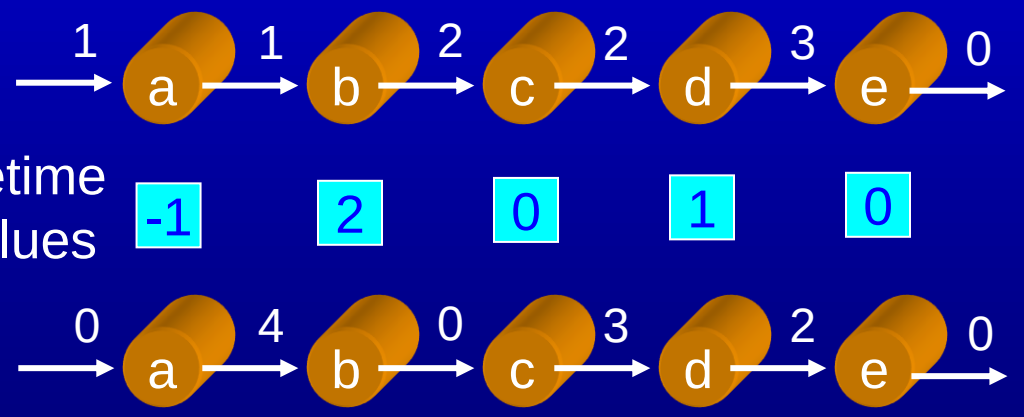


Path Weights after Retiming

- Path weight depends on retiming of extreme nodes only

$$w'(v_i,\ldots,v_j) = w(v_i,\ldots,v_j) + r_j - r_i$$

- Weight of cycle is invariant on retiming

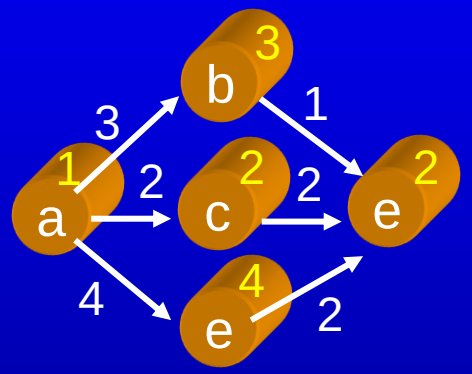


$w(a,\ldots,e) = 1+2+2+3 = 8$

$w'(a,\ldots,e) = 4+0+3+2 = 9$
$= w(a,\ldots,e) + 0 - (-1)$

Definitions

W_{ij} = min $w(v_i,\ldots,v_j)$ for all paths $(v_i,\ldots,v_j)$



W_{ae} = w(a,b,e) = w (a,c,e) = 4

D_{ij} = max $d(v_i,\ldots,v_j)$ for all paths $(v_i,\ldots,v_j)$ with weight W_{ij}

d(a,b,e) = 1+3+2 = 6
d(a,c,e) = 1+2+2 = 5

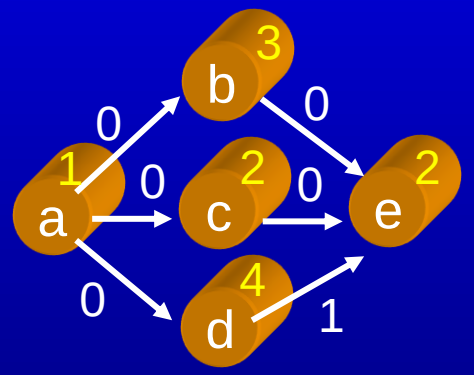
D_{ae} = d(a,b,e) = 6

Legality and Timing Feasibility

- Legality
  - No negative edge weights

  (number of registers ≥ 0)
- Timing Feasibility
  - given clock period ϕ
  - Path delay larger than ϕ should be broken by at least one register

$w'_{ij} \geq 0$



$W'_{ae} = 0$

$D_{ae} = 6$

Infeasible for $\phi = 5$

Feasible for $\phi = 6$

$W'_{ij} \geq 1$ if $D_{ij} > \phi$

Feasibility of Retiming

- Retiming is feasible if
  - legal
  - retimed graph is timing feasible
- Retiming r is feasible if

$\forall\ (v_i, v_j) \in E$

Legality Check

$w'_{ij} \geq 0 \Rightarrow w_{ij} + r_j - r_i \geq 0 \Rightarrow r_i - r_j \leq w_{ij}$

$\forall\ v_i, v_j : D(v_i, v_j) > \phi$

Timing Check

$W'_{ij} \geq 1 \Rightarrow W_{ij} + r_j - r_i \geq 1 \Rightarrow r_i - r_j \leq W_{ij} - 1$

System of Linear Inequalities

- Compute W and D matrices
- Build set of linear inequalities
  - Upto 2 inequalities for every pair of nodes
- Solve for vector r

$\forall\ (v_i, v_j) \in E$ $\quad r_i - r_j \leq w_{ij}$

$\forall\ v_i, v_j : D(v_i, v_j) > \phi$ $\quad r_i - r_j \leq W_{ij} - 1$

Remove Redundant Inequations and Solve

$r_1 - r_2 \leq 2$
$r_1 - r_2 \leq 1$

$r_1 - r_3 \leq -1$
$r_1 - r_3 \leq 0$

$r_2 - r_4 \leq -1$
$r_2 - r_4 \leq -2$

$r_2 - r_3 \leq 1$

$r_3 - r_4 \leq -1$

$r_1 - r_2 \leq 1$
$r_1 - r_3 \leq -1$
$r_2 - r_4 \leq -2$
$r_2 - r_3 \leq 1$
$r_3 - r_4 \leq -1$

Retain one equation per (r_i, r_j) pair
Solve for r_1, r_2, r_3, r_4

How do we Solve System of Inequalities?

- General problem: Linear Programming (LP)
  - Maximise $\Sigma c_i x_i$ given $Ax \leq b$
  - Simplex method (Exponential time)
  - Ellipsoid/Karmakar (Polynomial time)
- Restricted version: Integer Linear Programming (ILP)
  - Variables restricted to be integers
  - NP-Complete!

$$r_1 - r_2 \leq 1$$
$$r_1 - r_3 \leq -1$$
$$r_2 - r_4 \leq -2$$
$$r_2 - r_3 \leq 1$$
$$r_3 - r_4 \leq -1$$

$$A_{11}x_1 + A_{12}\, x_2 \leq b_1$$
$$A_{21}x_1 + A_{22}\, x_2 \leq b_2$$
$$A_{31}x_1 + A_{32}\, x_2 \leq b_3$$
$$A_{41}x_1 + A_{42}\, x_2 \leq b_4$$
$$A_{51}x_1 + A_{52}\, x_2 \leq b_5$$

Inequalities in Retiming Problem

- Restricted system of inequalities
  - Exactly 2 variables in each inequality
  - Coefficients are 1, -1
- Can be solved in Polynomial time!

$$r_1 - r_2 \leq 1$$
$$r_1 - r_3 \leq -1$$
$$r_2 - r_4 \leq -2$$
$$r_2 - r_3 \leq 1$$
$$r_3 - r_4 \leq -1$$

Solution to System of Inequalities

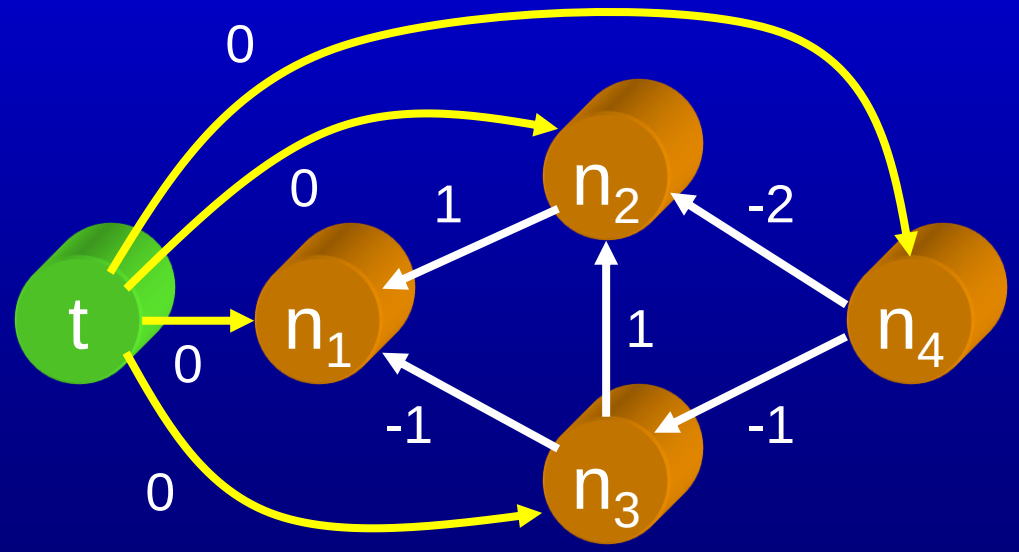
- No solution
- System has infinite solutions
  - if r is a solution, r+c is also a solution
- Let $r_1 = 0$
  - Can set arbitrary r_i to 0
  - For example:
    - if (2,3,-1,0) is one solution, (0,1,-3,-2) is also a solution

$$r_1 - r_2 \leq 1$$
$$r_1 - r_3 \leq -1$$
$$r_2 - r_4 \leq -2$$
$$r_2 - r_3 \leq 1$$
$$r_3 - r_4 \leq -1$$

Build Constraint Graph

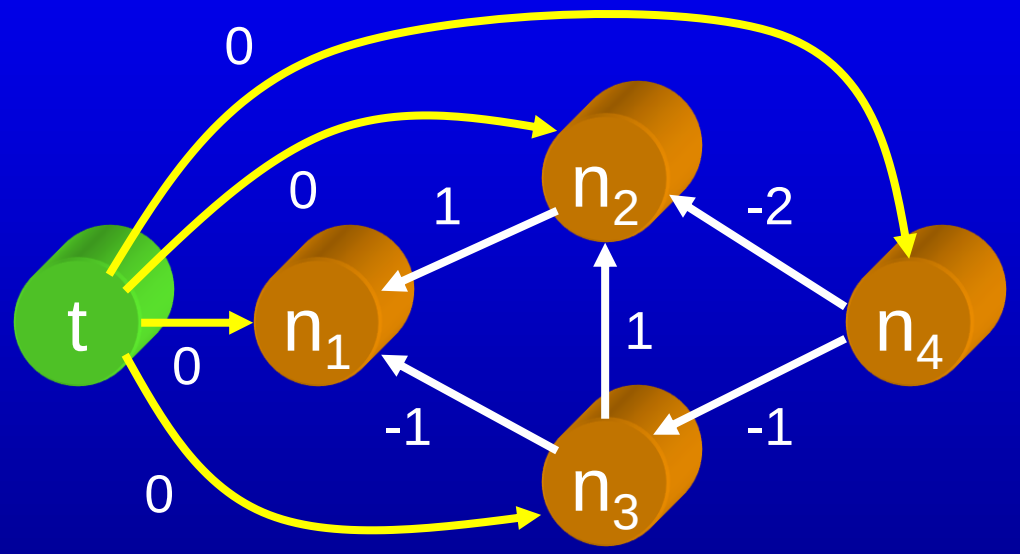
- One node n_i for each r_i
- Constraint $r_a - r_b \leq k$ implies Edge Weight $wt(n_b \rightarrow n_a) = k$
- Add auxiliary node t
  - zero-weight edges to all other nodes from t

$r_1 - r_2 \leq 1$
$r_1 - r_3 \leq -1$
$r_2 - r_4 \leq -2$
$r_2 - r_3 \leq 1$
$r_3 - r_4 \leq -1$



Solution using Shortest Path Algorithm

- Find shortest path in graph from t to all other nodes
  - Polynomial time
  - Bellman-Ford algorithm
- This gives r_i values for other nodes!



Solution: $r_1 = -2$, $r_2 = -2$, $r_3 = -1$, $r_4 = 0$

Why does Shortest Path work?

Consider edge $n_b \rightarrow n_a$

$sp\ (t \rightarrow n_a) \leq sp\ (t \rightarrow n_b) + wt\ (n_b \rightarrow n_a)$ -- by defn of shortest path

$sp\ (t \rightarrow n_a) - sp\ (t \rightarrow n_b) \leq wt\ (n_b \rightarrow n_a)$

Letting

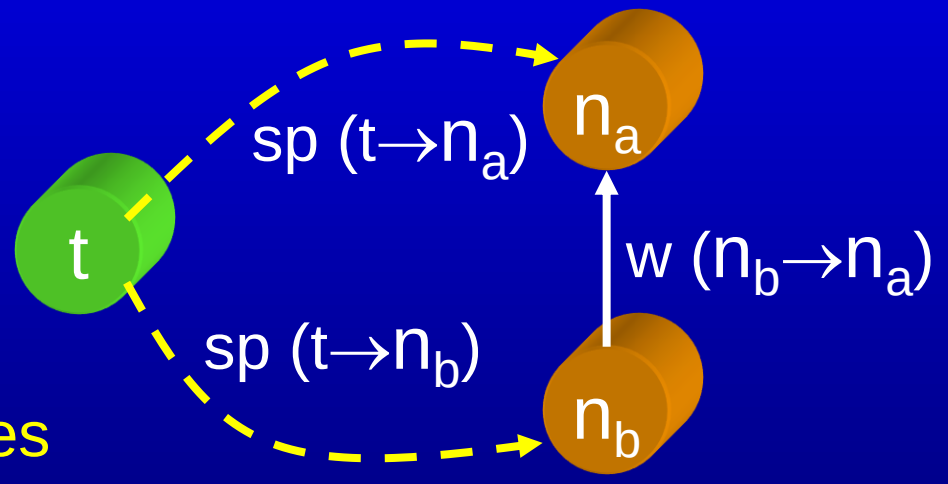
$r_a = sp\ (t \rightarrow n_a)$

$r_b = sp\ (t \rightarrow n_b)$

$r_a - r_b \leq wt\ (n_b \rightarrow n_a)$

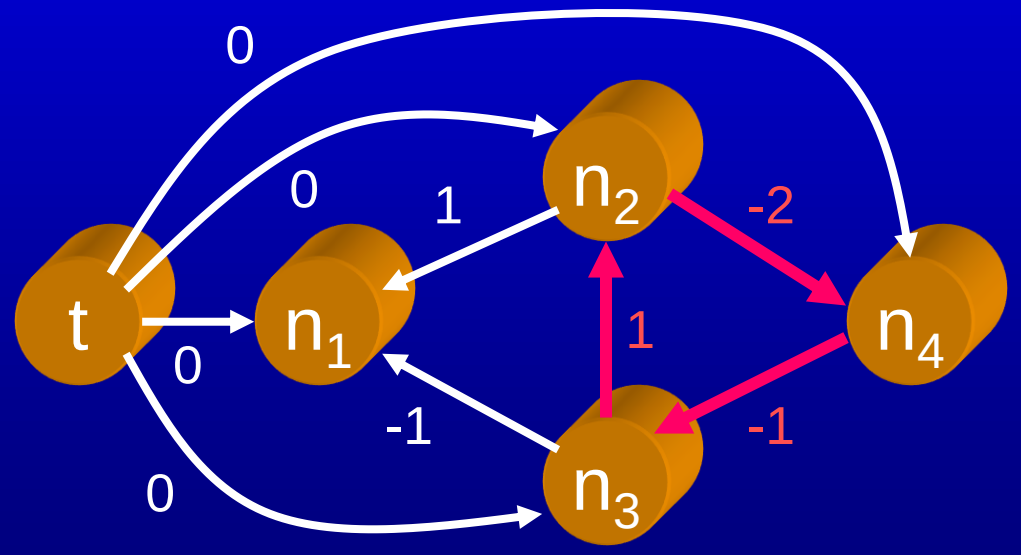
$r_a - r_b \leq k$

Thus, choice of sp for r_i satisfies constraint represented by edge wt



What if Graph has Negative Cycle?

- Sum of edge wts in cycle is negative
- Bellman-Ford algorithm returns error
  - shortest path not defined



Negative Cycle Leads to Infeasible Constraints

$r_2 - r_1 \leq wt\ (n_1 \rightarrow n_2)$
$r_3 - r_2 \leq wt\ (n_2 - n_3)$
$r_4 - r_3 \leq wt\ (n_3 - n_4)$
...
$r_k - r_{k-1} \leq wt\ (n_{k-1} - n_k)$
$r_1 - r_k \leq wt\ (n_k - n_1)$

Ref: Cormen, Leiserson, and Rivest, "Introduction to Algorithms"

Adding both sides:

$0 \leq wt\ (n_1 \rightarrow n_2) + wt\ (n_2 - n_3) + \ldots + wt\ (n_{k-1} - n_k) + wt\ (n_k - n_1)$

Impossible because RHS is negative

i.e., Bellman-Ford algorithm can be used directly for our problem

Now, we have Retiming values for all nodes (gates)

Retiming Strategy

- Compute D and W of original circuit
- For given ϕ set up inequalities and solve for r

$\forall\ (v_i, v_j) \in E$ $\quad r_i - r_j \leq w_{ij}$

$\forall\ v_i, v_j : D(v_i, v_j) > \phi$ $\quad r_i - r_j \leq W_{ij} - 1$

- If solution exists reduce ϕ and solve again
- Until no more solution

How to Choose ϕ Values?

- Optimal clock period has to be one of the D_{ij} values
  - otherwise, we could reduce ϕ to the nearest D_{ij} value without violating clock period
- First enumerate and sort all D_{ij} values in decreasing order
  - E.g.: 20, 16.5, 15, 13.2, 10, 8, 7, 5
- Use binary search