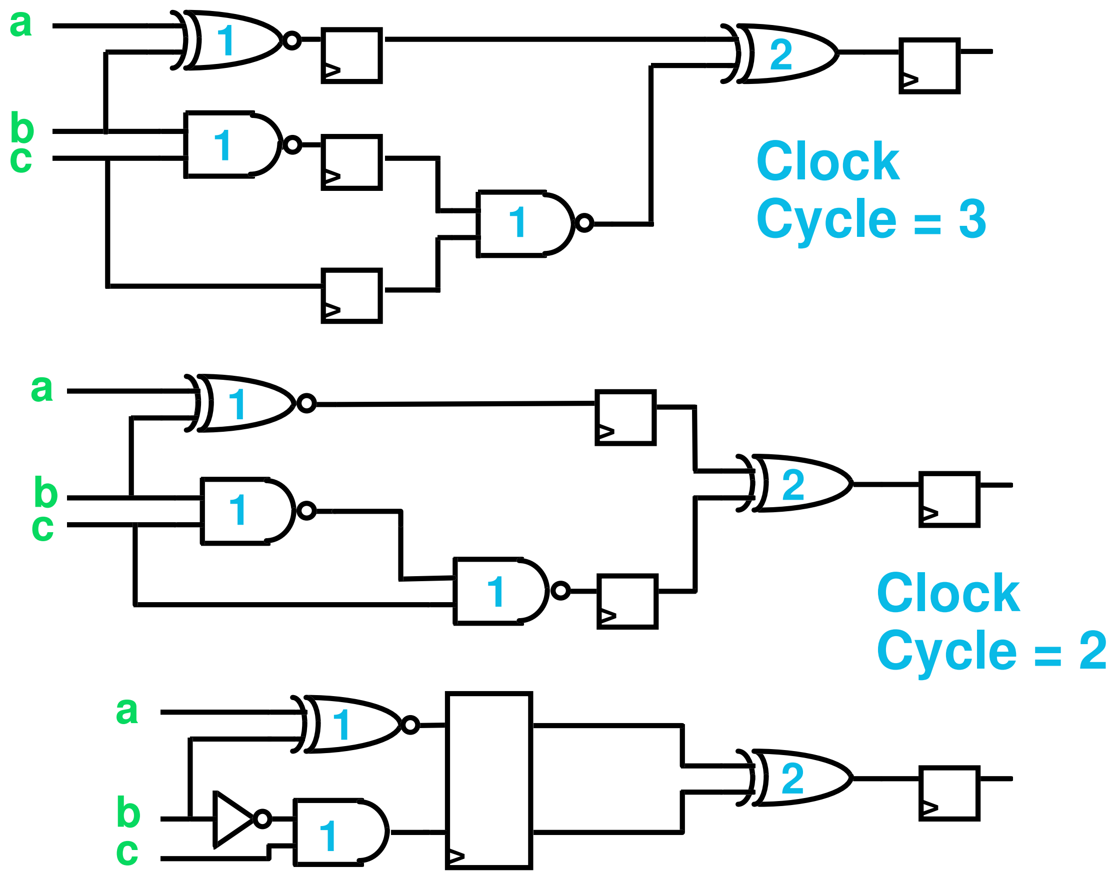
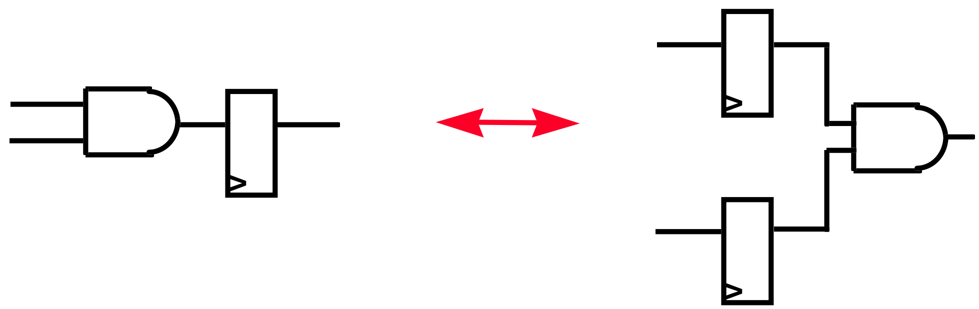
Retiming

Repositioning registers in sequential circuit



Acyclic Sequential Circuits

Basic transformation

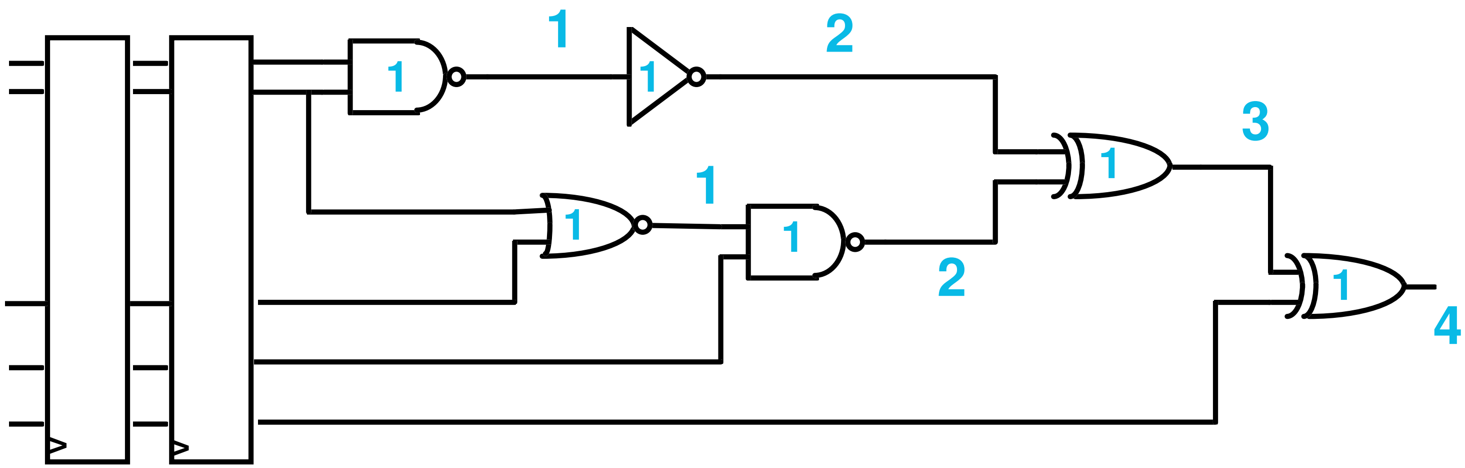


Given a combinational circuit with k registers at each input, can easily determine retiming for best clock period

- Levelize circuit 0,1, …, L
- Position registers at levels $\left\lceil \frac{L}{k} \right\rceil p \quad 0 \leq p < k$

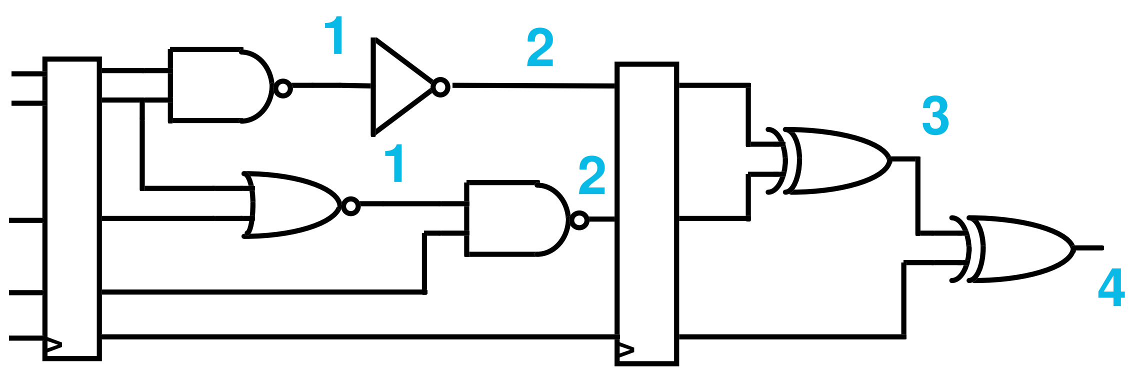
(unit-delay assumption)

Acyclic Circuit Example

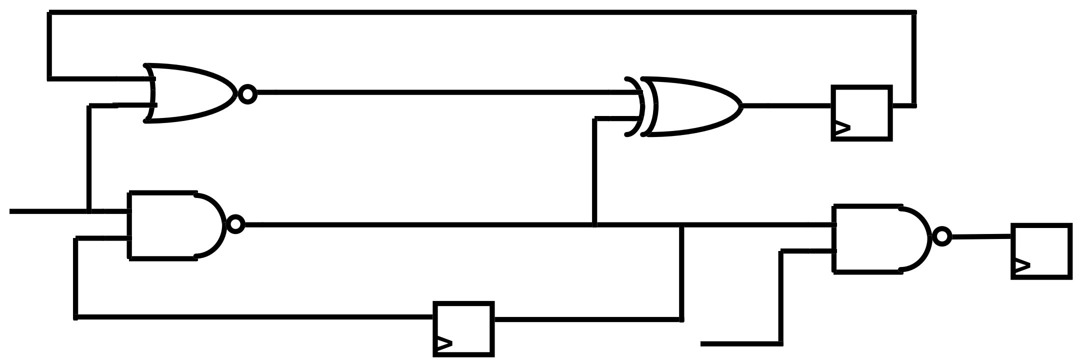


$L = 4\ ,\ \ k = 2$

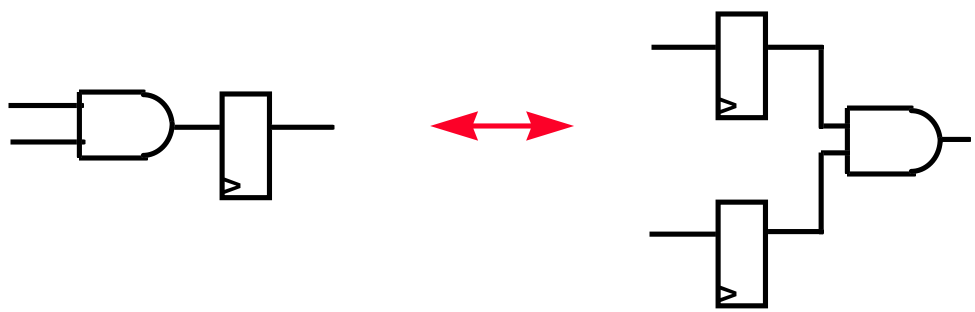
Place registers at levels $\left(\frac{4}{2}\right)0,\ \left(\frac{4}{2}\right)1$



Cyclic Circuits

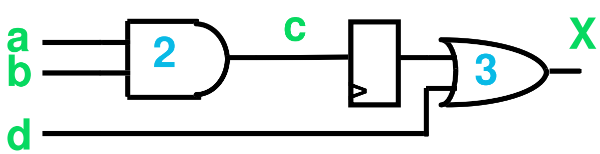


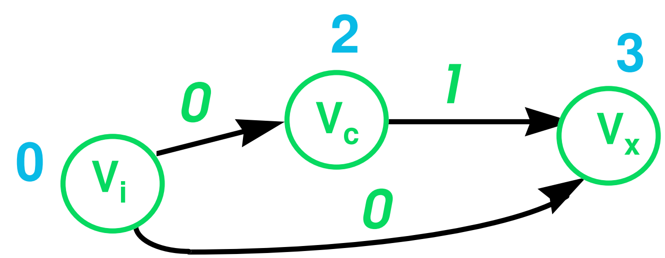
cannot levelize cyclic circuit but can still apply basic transformation

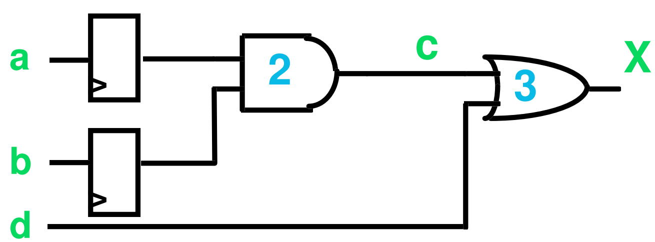


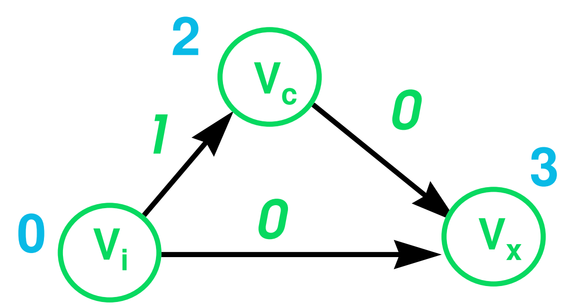
repeatedly to obtain better performance and/or minimize number of registers

Circuit Model



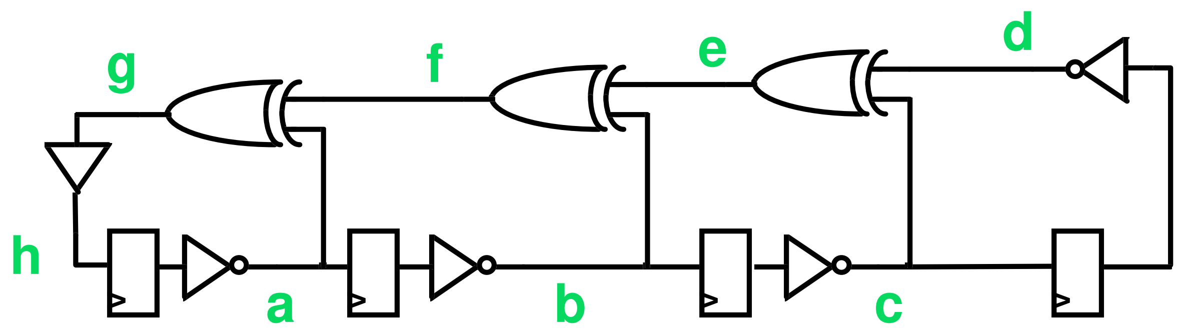


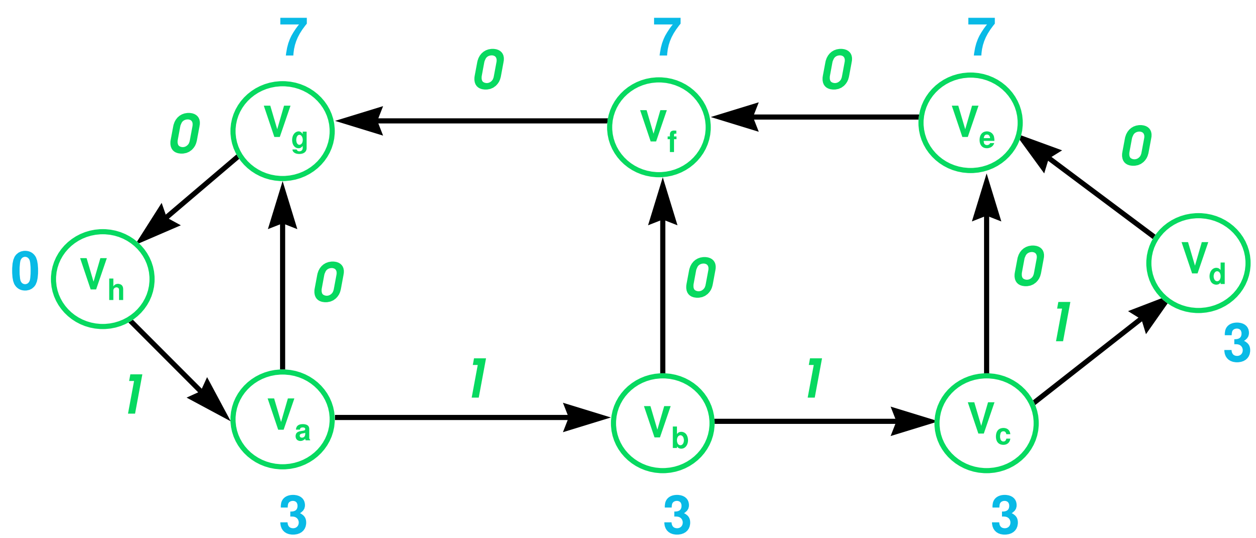




circuit + environment

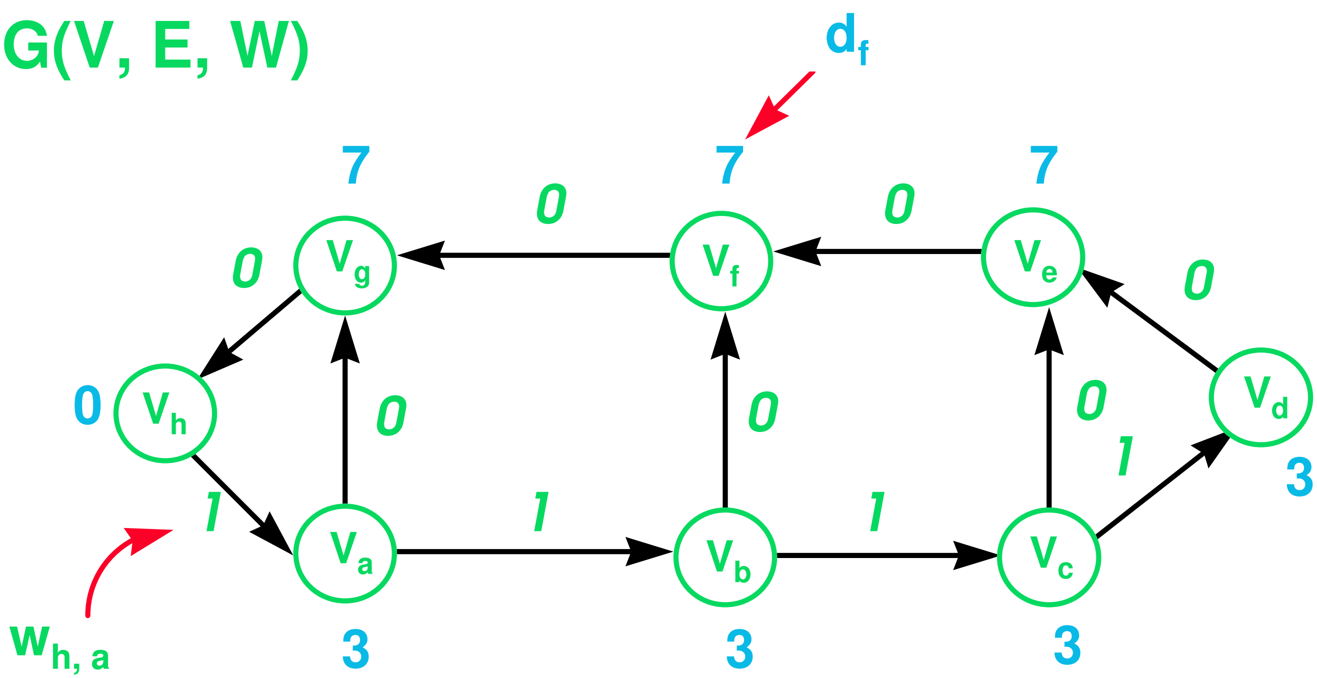
Circuit Model - 2





Synchronous Networks

$G(V, E, W)$



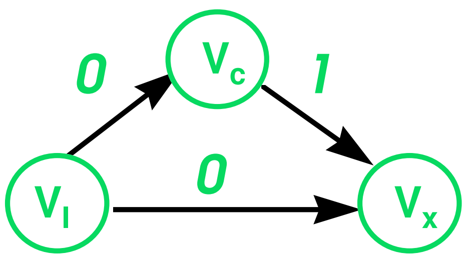
path delay $d(v_i, \ldots, v_j) = \sum_{k:\, v_k \in (v_i, \ldots, v_j)} d_k$

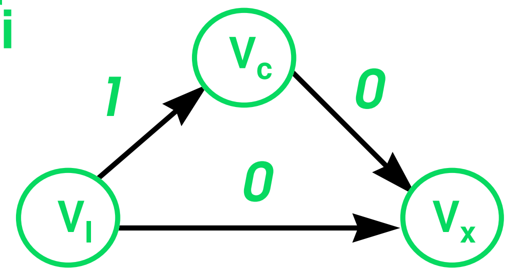
path weight $w(v_i, \ldots, v_j) = \sum_{k,\, l:\, (v_k, v_l) \in (v_i, \ldots, v_j)} w_{kl}$

Retiming of a Network

Integer valued vertex labeling $r : V \rightarrow Z$ that transforms $G(V,E,W)$ to $\tilde{G}(V,E,\tilde{W})$ where for each edge $(v_i,v_j) \in E$

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





$w_{IC} = 0$

$w_{CX} = 1$

$w_{IX} = 0$

	I	C	X
r	0	1	0

$\tilde{w}_{IC} = 1$

$\tilde{w}_{CX} = 0$

$\tilde{w}_{IX} = 0$

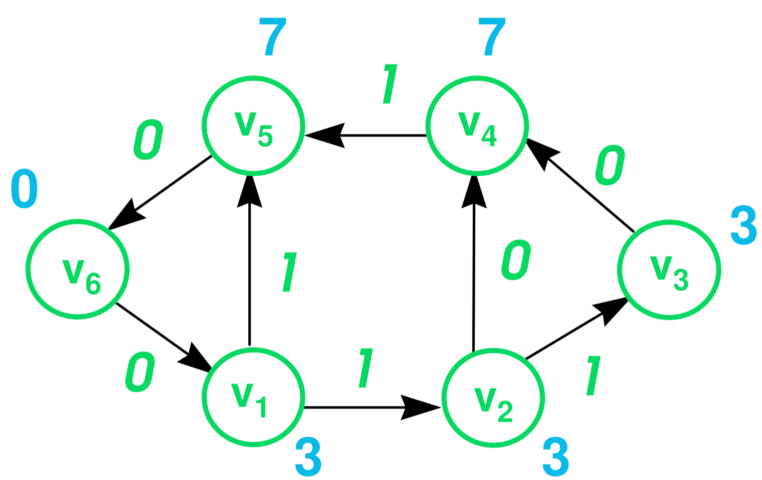
Weights on cycles are invariant in retiming

Legal Retiming

A retiming is legal if the retimed network has no negative weights

Definition: $\underline{W}(v_i, v_j) = \min w(v_i, \ldots, v_j)$
over all paths from v_i to v_j

$\underline{D}(v_i, v_j) = \max d(v_i, \ldots, v_j)$
over all paths from v_i to v_j
with weight $\underline{W}(v_i, v_j)$



v_1, v_4

$\underline{W}(v_1, v_4) = 1$

$\underline{D}(v_1, v_4) = 13$

Timing Feasible Retiming

For a given cycle time ϕ the network is timing feasible if it can operate correctly under ϕ

$\Rightarrow$ will be feasible if and only if $\underline{W}(v_i,v_j) \geq 1$ for all $v_i, v_j \in V$ such that $\underline{D}(v_i, v_j) > \phi$

Theorem: Given $G(V,E,W)$, clock period ϕ, r is a feasible retiming if and only if:

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

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

Minimum Delay Retiming

RETIME-DELAY (G(V,E,W)) {

Compute matrices W, D ;

Sort values of elements in D ;

Construct inequalities (a) ;

For each ϕ determined by binary search on elements in D {

Construct inequalities (b) ;

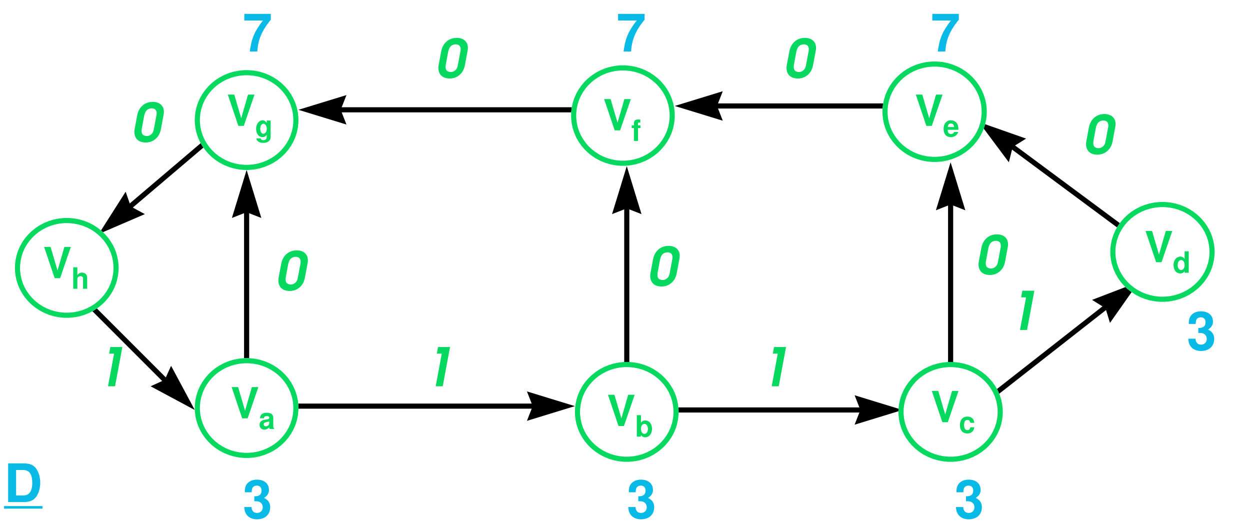
Solve inequalities ;

}

Determine r, ϕ_0 by last successful solution of inequalities;

}

Example



W / D

	a	b	c	d	e	f	g	h
a	0/0	1/6	2/9	3/12	2/16	1/13	0/10	0/7
b	1/20	0/0	1/6	2/9	1/13	0/10	0/17	0/17
c	1/27	2/30	0/0	1/6	0/10	0/17	0/24	0/24
d	1/27	2/30	3/33	0/0	0/10	0/17	0/24	0/24
e	1/24	2/27	3/30	4/33	0/0	0/14	0/21	0/21
f	1/17	2/20	3/23	4/26	3/27	0/0	0/14	0/14
g	1/10	2/13	3/16	4/19	3/23	2/20	0/0	0/7
h	1/3	2/6	3/9	4/12	3/16	2/13	1/10	0/0

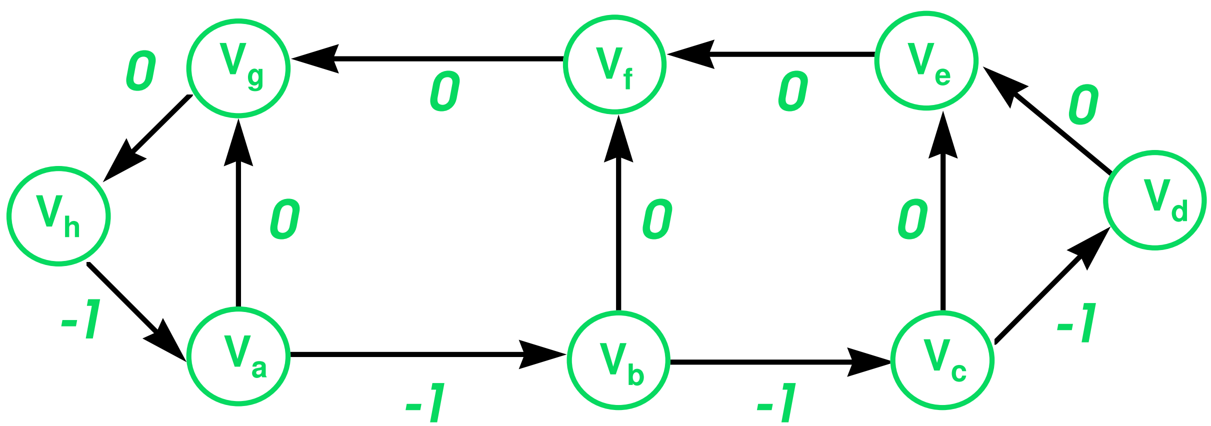
Binary Search

Sorted values of D

{ 33, 30, 27, 26, 24, 23, 21, 20, 19, 17, 16, 14, 13, 12, 10, 9, 7, 6, 3, 0 }

Perform binary search after constructing inequalities
(a) $r_i - r_j \leq w_{ij} \Rightarrow r_j \geq r_i - w_{ij}$

Represent inequalities by graph with same topology as network



Solving Inequalities

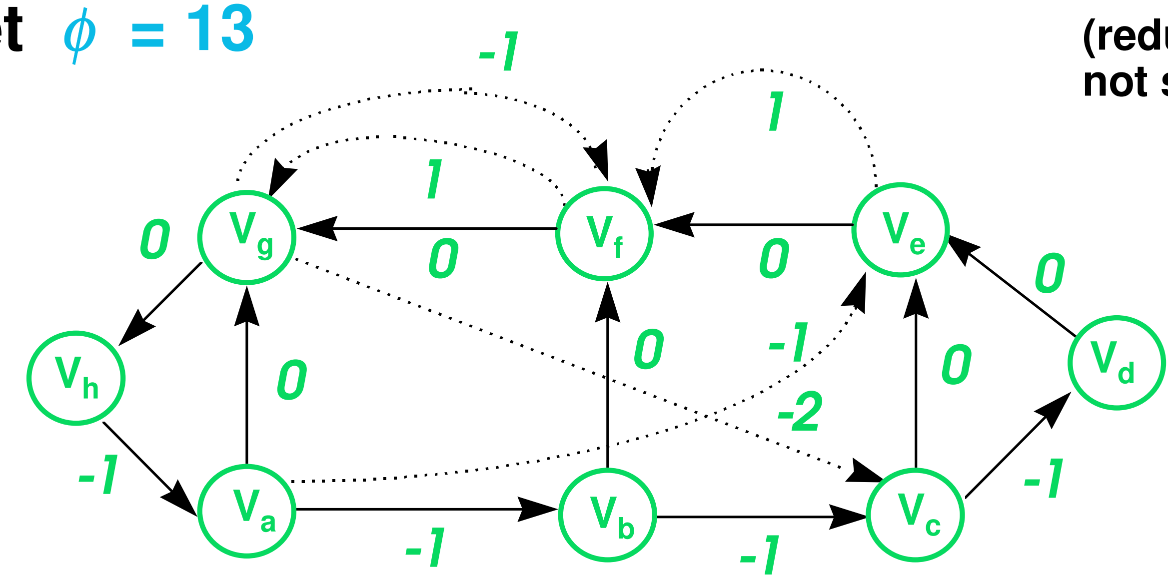
(a) $r_i - r_j \le w_{ij} \Rightarrow r_j \ge r_i - w_{ij}$

(b) $r_i - r_j \le \underline{W}(v_i, v_j) - 1 \Rightarrow r_j \ge r_i - \underline{W}(v_i, v_j) + 1$

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

Set $\phi = 13$

(redundant edges not shown)

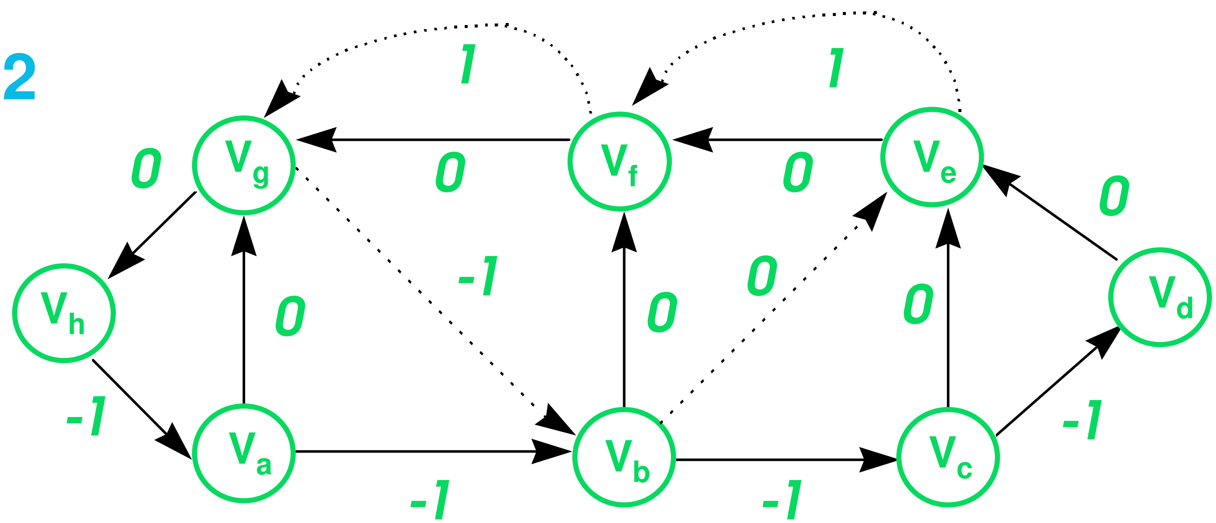


Positive cycle in graph $\Leftrightarrow$ constraints cannot be solved

no positive cycle in above graph

Solving Inequalities - 2

$\phi = 12$



dotted edge entries:

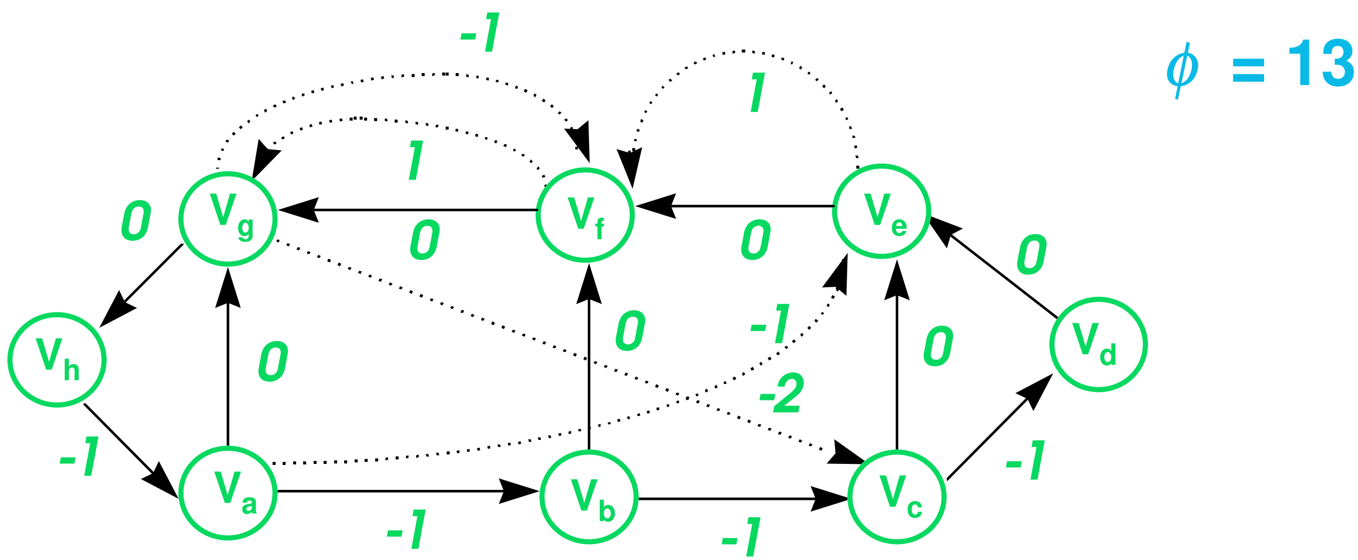
	W/D
e,f	*0*/14
f,g	*0*/14
g,b	*2*/13
b,e	*1*/13

positive cycle V_e, V_f, V_g, V_b, V_e

$1 + 1 - 1 + 0 > 0$

Obtaining Retiming Vector

Retiming vector entries are the weights of the longest path from a reference vertex <u>to</u> the v_i

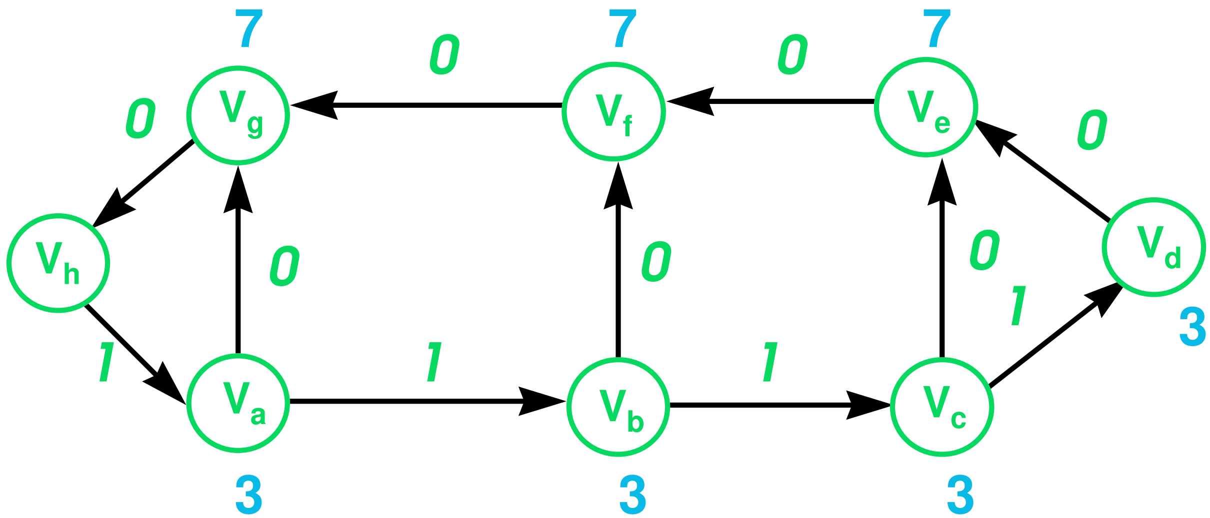


Choose V_h with $r_h = 0$

<u>Retiming vector</u>

$r_a = -1$ $r_b = -2$ $r_c = -2$ $r_d = -3$ $r_e = -2$ $r_f = -1$ $r_g = 0$ $r_h = 0$

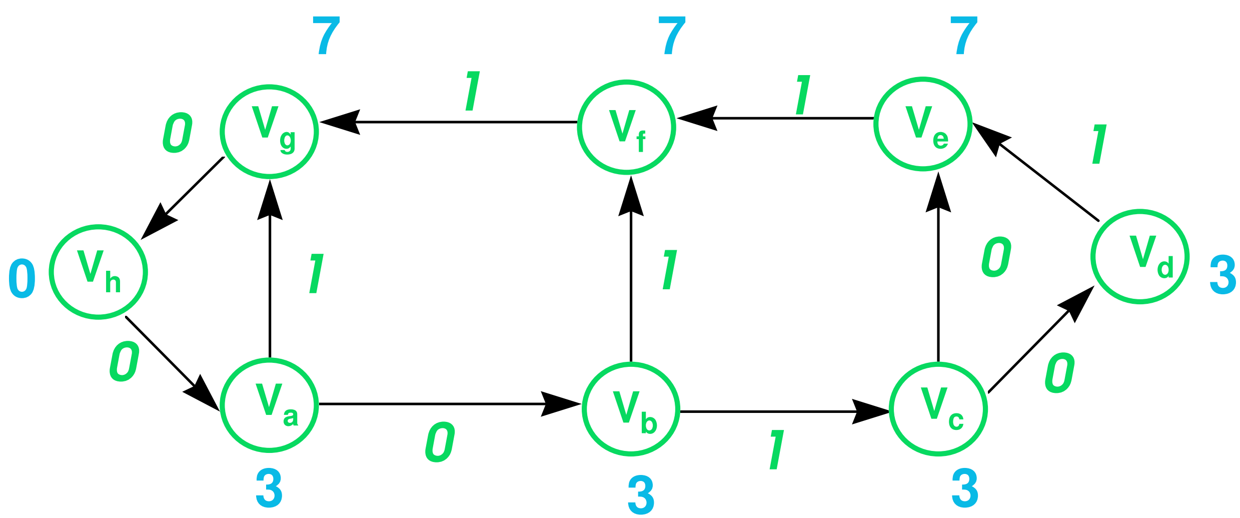
Original Network



Longest path with no register $V_c, V_e, V_f, V_g, V_h = 24$

Retiming vector - [*1 2 2 3 2 1 0 0*]

Retimed Network - 2



Longest path with no register V_g, V_h, V_a, V_b = 13

Summary

Logic-level transformation, as opposed to State Transition Graph level, that can be applied to sequential circuits with large number of memory elements

Extensions of retiming algorithms

- Load dependent delays for nodes
- Integrate retiming with combinational logic timing optimization