

Retiming for Wire Pipelining in System-On-Chip

Chuan Lin and Hai Zhou

Electrical and Computer Engineering
Northwestern University

Outline

- Motivation
- Problem formulation
- Theoretical approach under a fixed clock period
- Algorithm
- Experimental results
- Conclusions

Motivation

- Industry trend
 - Frequency: 2X/generation, Die size: 1.25X/generation
 - Problem: global signal communication

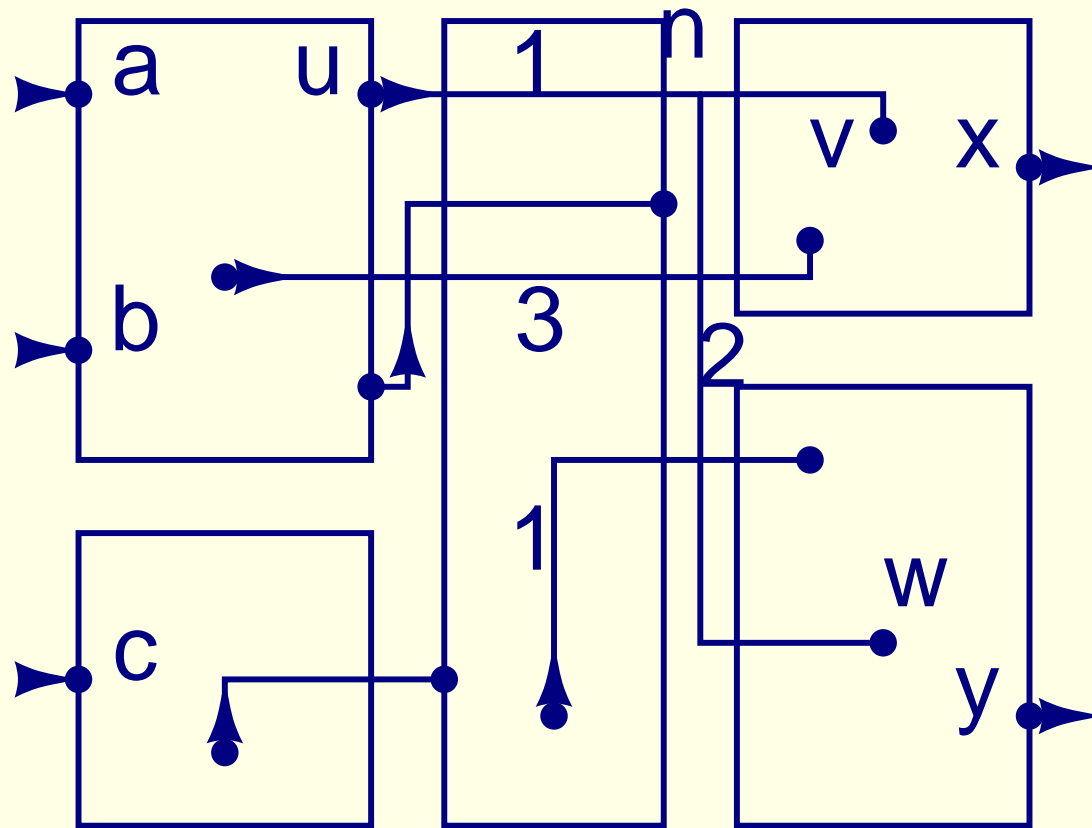
Motivation

- Industry trend
 - Frequency: 2X/generation, Die size: 1.25X/generation
 - Problem: global signal communication
- Recent research
 - Insert flip-flops on global wires (Intel, IBM, etc.)
 - How to keep functional correctness

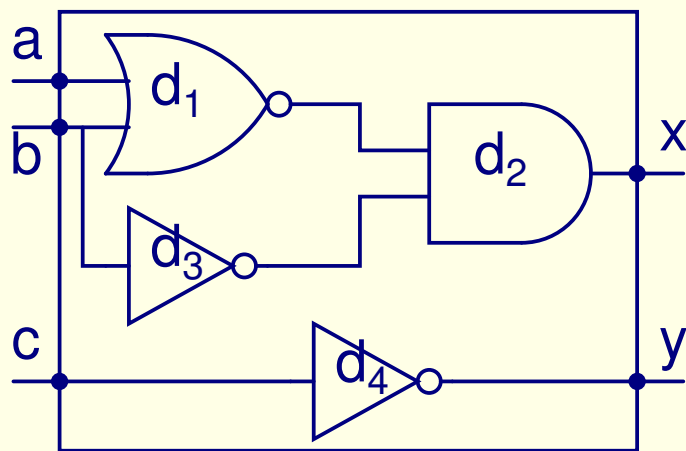
Motivation

- Retiming can move flip-flops w/o changing functionality
- Traditional retiming mainly consider gate delay
- We want flip-flops to pipeline long wire
 - Multiple flip-flops on a wire
 - Positions of flip-flops on a wire

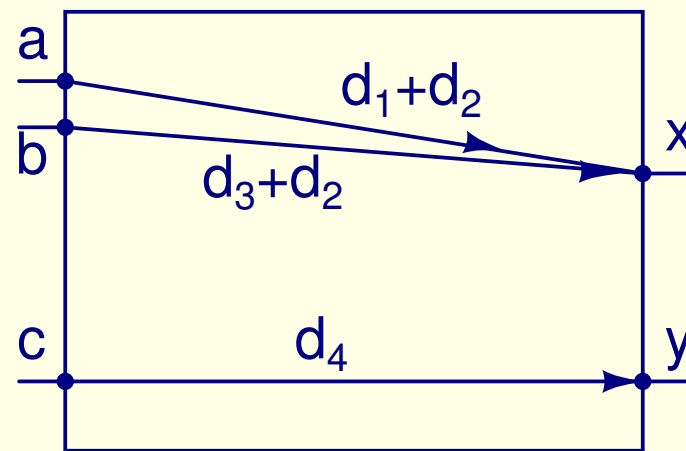
An SOC design example



Timing model for a combinational block

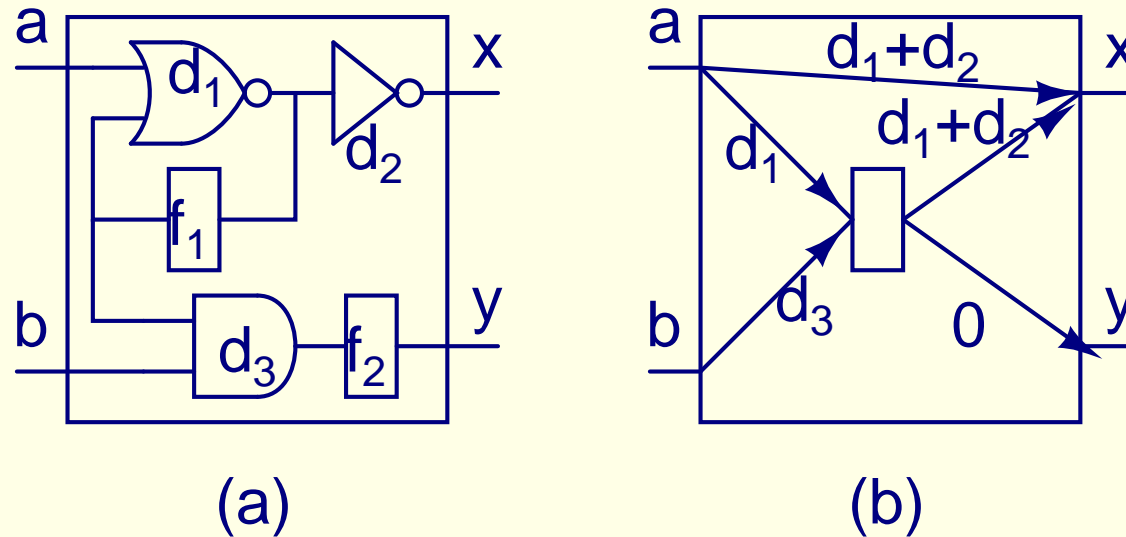


(a)



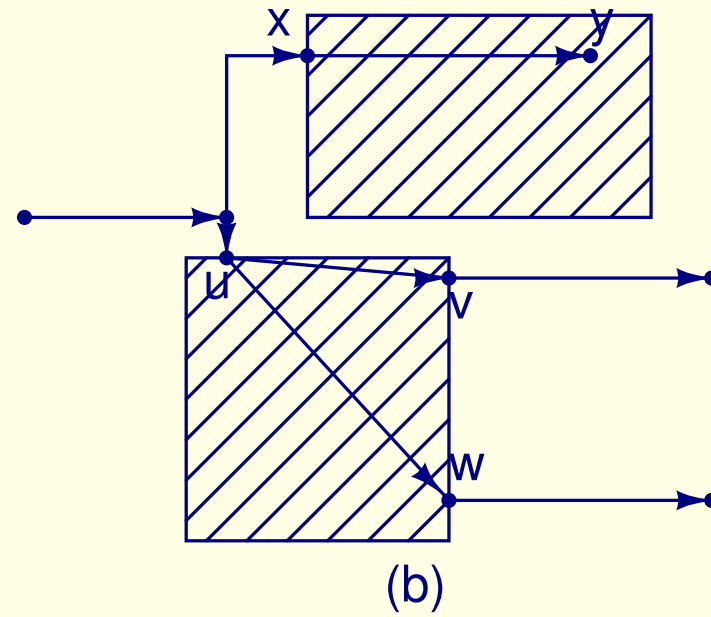
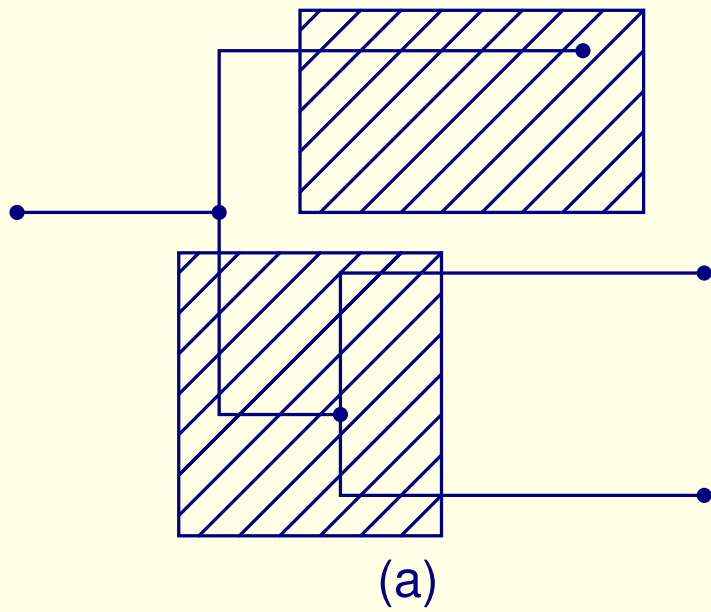
(b)

Timing model for a sequential block



- Virtual flip-flop
 - Arrival time of a : no larger than $T - d_1$
 - Arrival time of x : no smaller than $d_1 + d_2$
 - Maintain the connections

Timing model for a net



Problem formulation

[Minimum Period Wire Retiming]

- $G = (V, E)$, $E = E_1 \cup E_2$, $E_1 \cap E_2 = \emptyset$
delay: $d(e)$, #flip-flop: $w(e)$, $\forall e \in E$

Problem formulation

[Minimum Period Wire Retiming]

- $G = (V, E)$, $E = E_1 \cup E_2$, $E_1 \cap E_2 = \emptyset$
delay: $d(e)$, #flip-flop: $w(e)$, $\forall e \in E$
- $\forall e \in E_2$, $d(e)$ is proportional to its length
 - Buffers can be inserted to make it linear

Problem formulation

[Minimum Period Wire Retiming]

- $G = (V, E)$, $E = E_1 \cup E_2$, $E_1 \cap E_2 = \emptyset$
delay: $d(e)$, #flip-flop: $w(e)$, $\forall e \in E$
- $\forall e \in E_2$, $d(e)$ is proportional to its length
 - Buffers can be inserted to make it linear
- Find a relocation of flip-flops
 - No flip-flop change on any $e \in E_1$
 - Minimize the maximum delay between any two consecutive flip-flops(clock period)

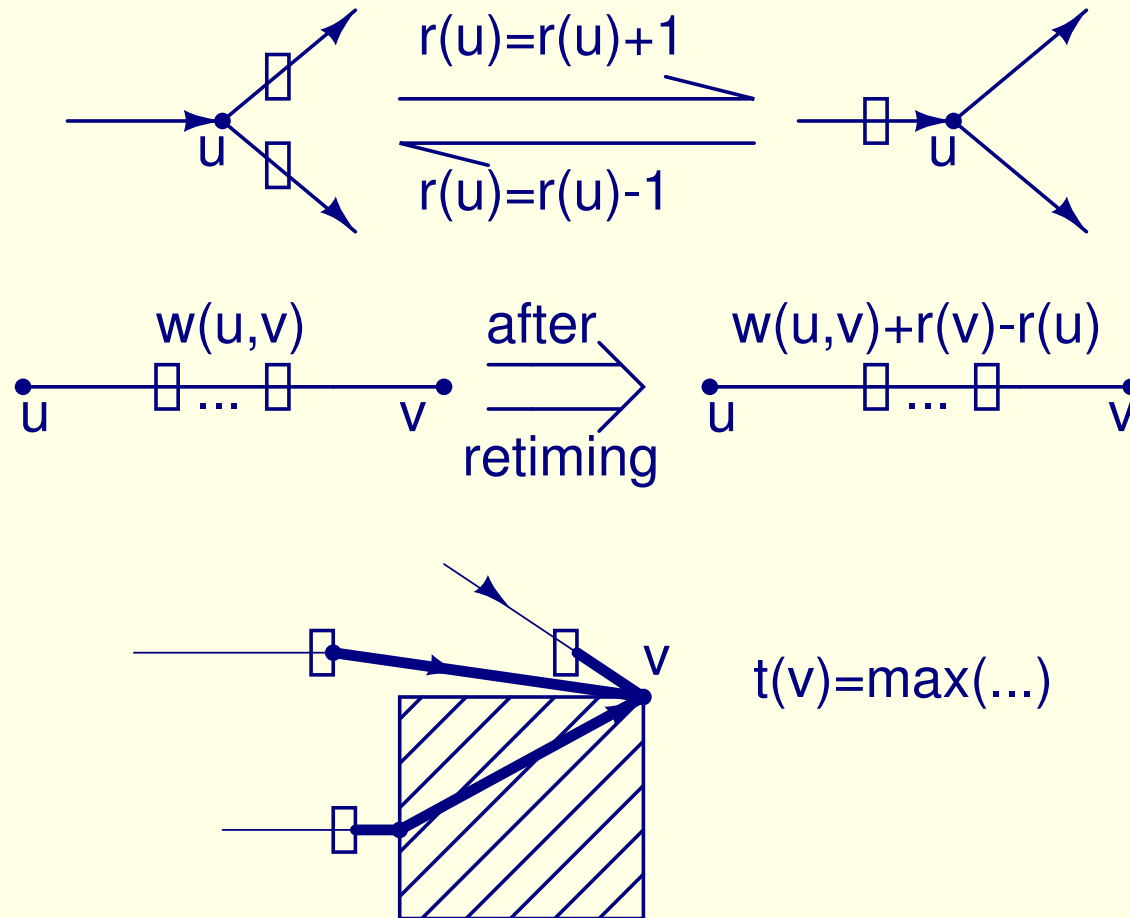
Strategy of solving the problem

- Fixed Period Wire Retiming
 - Determine if exists a retiming under a fixed clock period(T)

Strategy of solving the problem

- Fixed Period Wire Retiming
 - Determine if exists a retiming under a fixed clock period(T)
- Binary Search
 - Over the range from lower to upper bound

Two essential variables r and t



Requirements for r and t

$$r(x) = r(y), \quad \forall (x, y) \in E_1 \quad (1)$$

$$w(u, v) + r(v) - r(u) \geq 0, \quad \forall (u, v) \in E_2 \quad (2)$$

$$t(y) \geq t(x) + d(x, y), \quad \forall (x, y) \in E_1 \quad (3)$$

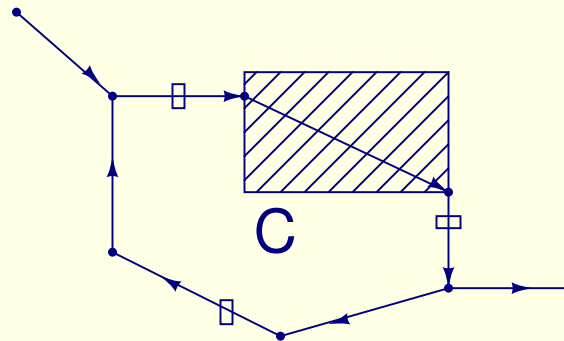
$$t(v) \geq t(u) + d(u, v) - [w(u, v) + r(v) - r(u)]T, \forall (u, v) \in E_2 \quad (4)$$

$$0 \leq t(u) \leq \mathbf{T}, \quad \forall u \in V \quad (5)$$

Lower bounds for T

$$T \geq T_1 \stackrel{\text{def}}{=} \max_{(x,y) \in E_1} d(x,y)$$

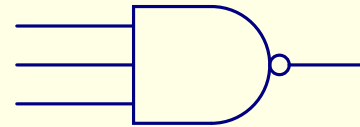
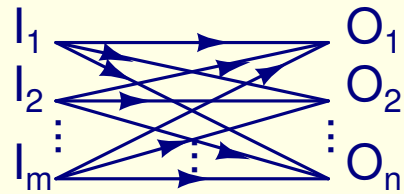
$$T \geq T_2 \stackrel{\text{def}}{=} \max_{c \in \text{cycle}} \frac{d(c)}{w(c)} \quad (6)$$



- Maximum Cycle Ratio (Howard's algorithm)
 - A. Dasdan, S. S. Irani, and R. K. Gupta [DAC 99]

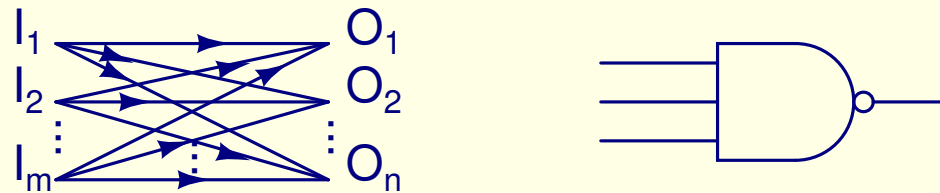
Upper bound for T

Lemma 1 *If each connected component in the subgraph $G_1 = (V, E_1)$ is a **complete bipartite graph**, the optimal clock period can be upper bounded by $T_1 + T_2$.*

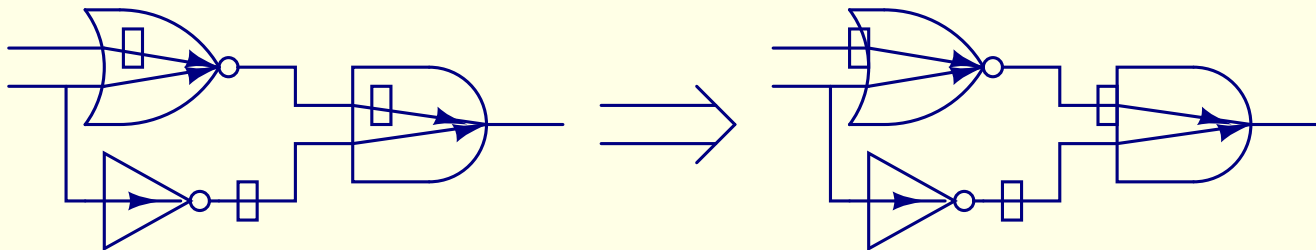


Upper bound for T

Lemma 1 *If each connected component in the subgraph $G_1 = (V, E_1)$ is a **complete bipartite graph**, the optimal clock period can be upper bounded by $T_1 + T_2$.*

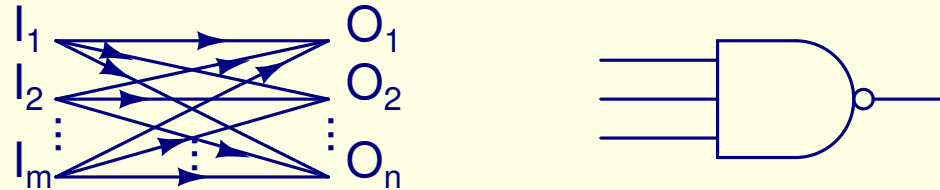


- Continuous retiming – P. Pan [ICCD'97]

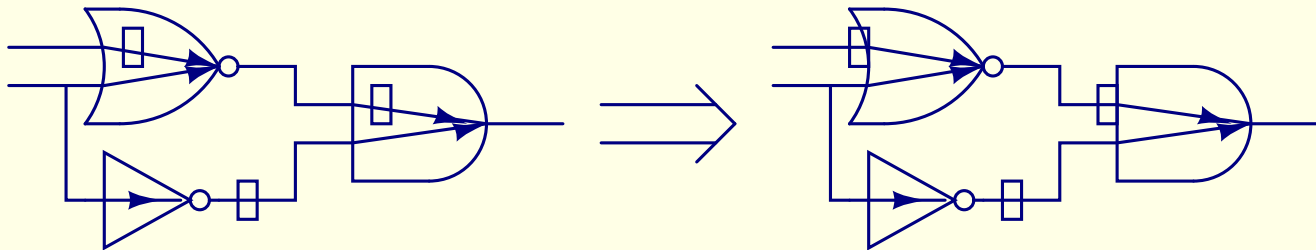


Upper bound for T

Lemma 1 *If each connected component in the subgraph $G_1 = (V, E_1)$ is a **complete bipartite** graph, the optimal clock period can be upper bounded by $T_1 + T_2$.*



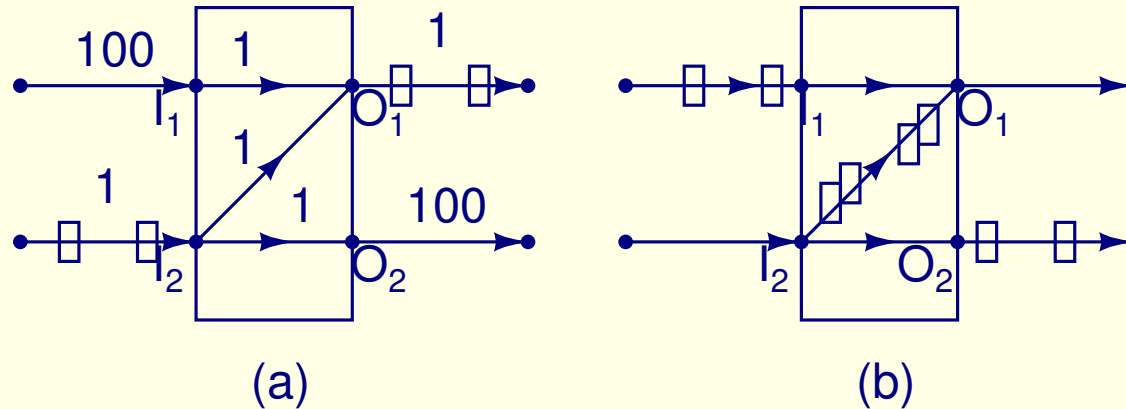
- Continuous retiming – P. Pan [ICCD'97]



- Would it still be an upper bound in a SOC design?

Upper bound for T

An example of non-complete bipartite circuit



- Continuous retiming: $T_{ub} = 35$
- Cong and Yuan's approach [DAC 03]: $T_{ub} = 202$
- Our approach: $T_{ub} = 102$ (**Optimal: 101**)

Solving the Fixed Period Wire Retiming problem

Theorem 1 *The fixed period wire retiming problem is feasible if and only if (1), (6) and (7) have a solution.*

$$r(x) = r(y), \quad \forall (x, y) \in E_1 \quad (1)$$

$$T \geq T_2 \stackrel{\text{def}}{=} \max_{c \in \text{Cycle}} \frac{d(c)}{w(c)} \quad (6)$$

$$r(v) - r(u) \geq \lceil (d(p) - w(p)T)/T \rceil - 1, \quad \forall u \neq v \in V, u \stackrel{p}{\rightsquigarrow} v \quad (7)$$

Solving the Fixed Period Wire Retiming problem

Theorem 1 *The fixed period wire retiming problem is feasible if and only if (1), (6) and (7) have a solution.*

$$r(x) = r(y), \quad \forall (x, y) \in E_1 \quad (1)$$

$$T \geq T_2 \stackrel{\text{def}}{=} \max_{c \in \text{cycle}} \frac{d(c)}{w(c)} \quad (6)$$

$$r(v) - r(u) \geq \lceil (d(p) - w(p)T)/T \rceil - 1, \quad \forall u \neq v \in V, u \stackrel{p}{\rightsquigarrow} v \quad (7)$$

- Why solving (1),(6),(7) is easier than (1)-(5)?
 - (7) is a difference relation involving only two INT variables
 - (1) can be replaced by a pair of inequalities
 - Take T_2 as the lower bound of the binary search

Algorithm

Algorithm Retiming for Wire Pipelining

Find the upper bound T_u and lower bound T_l ;

do

$$T = \frac{T_u + T_l}{2};$$

Compute the inequality set (7)

$$\{\forall u \neq v \in V, u \xrightarrow{p} v : r(v) - r(u) \geq \left\lceil \frac{d(p) - w(p)T}{T} \right\rceil - 1\}$$

Check the solvability of (1) and (7)

If exists a retiming then

$$T_u = T;$$

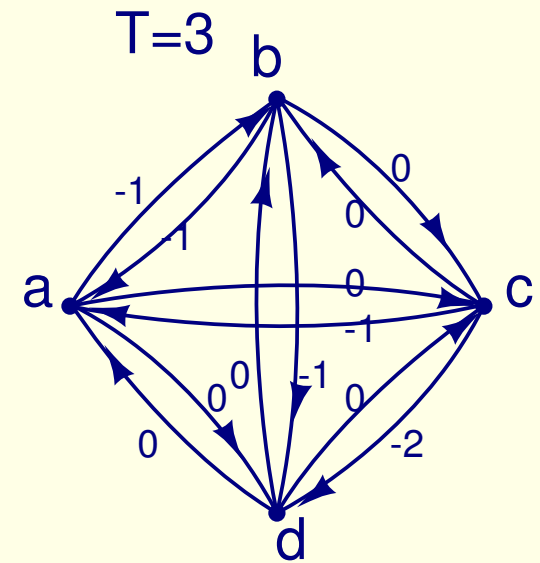
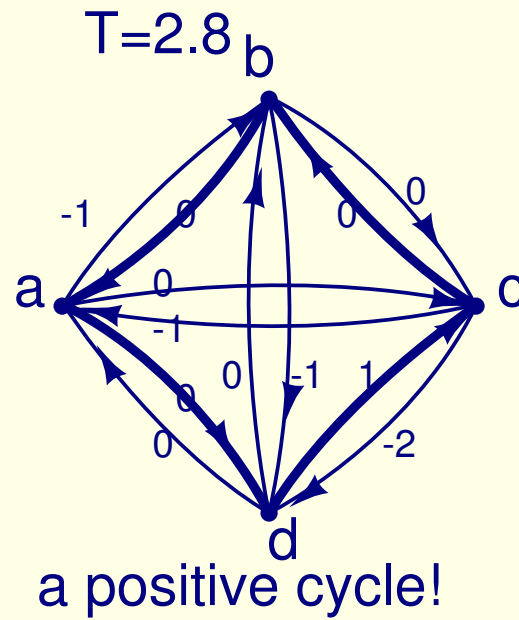
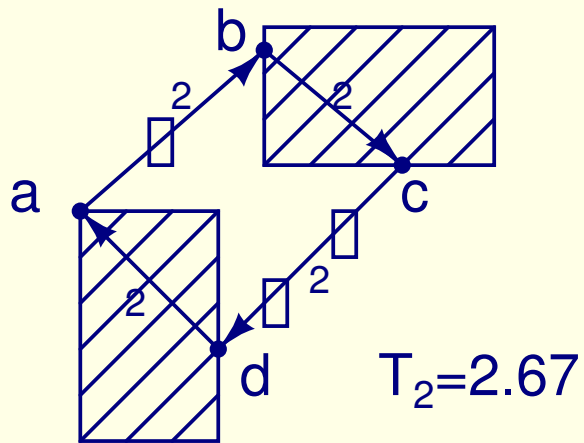
Else

$$T_l = T;$$

while $T_u - T_l > \epsilon$;

Algorithm

An example of how the algorithm works



Computational complexity

- One binary search step:

$$O(|V|^3)$$

- Entire algorithm:

$$O(|V|^3 \lg \frac{T_u - T_l}{\epsilon})$$

Experimental results

- Benchmark: ISCAS-89
 - 1st test set: treat gates as blocks
 - 2nd test set: circuits w/ non-complete bipartite blocks
 - * Use hMETIS to partition a circuit into groups
 - * Treat each group as a block
- Binary search precision: $\epsilon = 0.1$

Experimental results

Circuit	$ V $	$ E $	T_2	w/o non-CB	w/ non-CB
				T_{opt}	T_{opt}
myex	21	24	13.0	18.7	24.0
s386	519	700	51.0	51.1	55.0
s400	511	665	32.2	32.2	50.6
s444	557	725	35.0	35.2	63.2
s838	1299	1206	76.0	76.0	84.0
s953	1183	1515	60.6	60.6	69.5
s1238	1581	2100	100.2	100.3	100.3
s1488	2054	2780	70.1	70.6	73.3
s1494	2054	2792	76.8	76.9	80.0
s5378	7205	8603	111.0	111.2	115.3

Experimental results

Circuit	$ V $	$ E $	w/o non-CB		w/ non-CB	
			No.Iter	time(s)	No.Iter	time(s)
myex	21	24	7	0.00	10	0.01
s386	519	700	5	1.97	10	3.67
s400	511	665	6	1.64	10	3.38
s444	557	725	5	2.23	10	4.31
s838	1299	1206	5	8.79	11	33.42
s953	1183	1515	5	9.76	10	17.56
s1238	1581	2100	5	7.88	11	28.45
s1488	2054	2780	5	35.17	11	98.88
s1494	2054	2792	5	34.13	11	62.86
s5378	7205	8603	5	684.60	13	1344.74

Summary

- Wire retiming is becoming more and more important on a SOC design

Summary

- Wire retiming is becoming more and more important on a SOC design
- Timing models for both combinational and sequential blocks are established

Summary

- Wire retiming is becoming more and more important on a SOC design
- Timing models for both combinational and sequential blocks are established
- FF location restrictions within blocks and on wires through blocks are handled uniformly

Summary

- Wire retiming is becoming more and more important on a SOC design
- Timing models for both combinational and sequential blocks are established
- FF location restrictions within blocks and on wires through blocks are handled uniformly
- A fully polynomial-time approximation scheme for clock period minimization was proposed, that is, for any given relative error $\epsilon > 0$, the running time is proportional to $1/\epsilon$.

Summary

- Wire retiming is becoming more and more important on a SOC design
- Timing models for both combinational and sequential blocks are established
- FF location restrictions within blocks and on wires through blocks are handled uniformly
- A fully polynomial-time approximation scheme for clock period minimization was proposed, that is, for any given relative error $\epsilon > 0$, the running time is proportional to $1/\epsilon$.
- Can we improve the computational complexity?

Further work

- Basic idea: fixpoint computation [DATE04]
- Results

Circuit	t_{old} (sec)	t_{new} (sec)
s386	3.67	0.01
s400	3.38	0.01
s444	4.31	0.01
s838	33.42	0.02
s953	17.56	0.07
s1488	98.88	0.05
s1494	62.86	0.09
s5378	1344.74	0.29
s13207	-	206.52
s35932	-	6.19
s38584	-	21992.67

Thank you !

Wire retiming by fixpoint computation

- Solution vector $X = (x_1, x_2, \dots, x_n)$, where $n = |V|$ and $x_v = (r(v), t(v))$
- $X = F(X)$
- Theorem: if T is feasible, we will reach the least fixpoint of F
- Iterative relaxation on the given graph G