VirtualSync: Timing Optimization by Synchronizing Logic Waves with Sequential and Combinational Components as Delay Units

Grace Li Zhang[1], Bing Li[1], Masanori Hashimoto[2] and Ulf Schlichtmann[1]
1 Chair of Electronic Design Automation, Technical University of Munich (TUM)
2 Department of Information Systems Engineering, Osaka University

Overview

Motivation

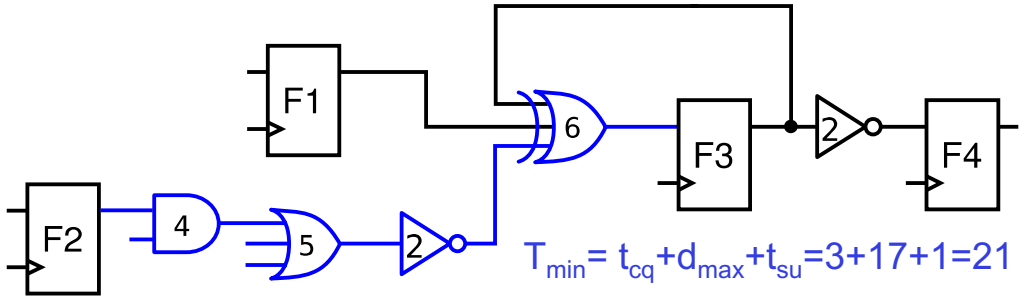
VirtualSync Timing Model

Timing Optimization Framework of VirtualSync

Experimental Results

Summary

The Traditional Timing Paradigm



Clock-to-q delay t_{cq}: 3
Setup time t_{su}: 1
Hold time t_h: 1

- **Sequential components** such as flip-flops synchronize signal propagations.
- **Combinational gates** perform logic computations.

Reduce design effort

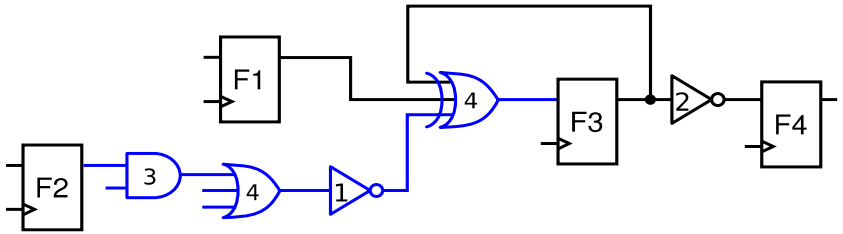
Disadvantages

Flip-flops have clock-to-q delays and impose setup time.

Delay imbalances between flip-flop stages degrade performance.

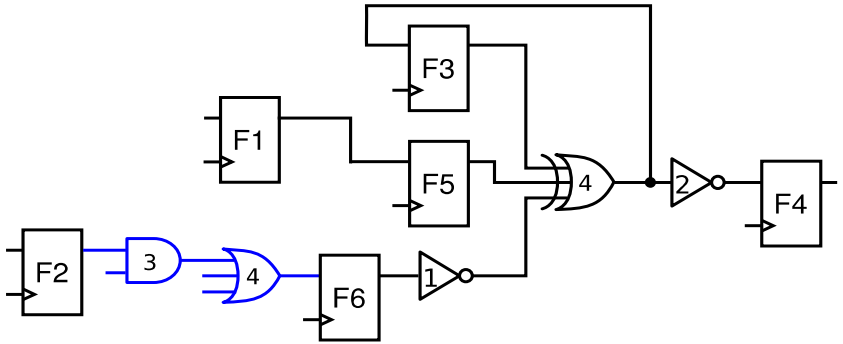
Timing Optimization Methods

Gate Sizing



T_{min}= 3+12+1=16

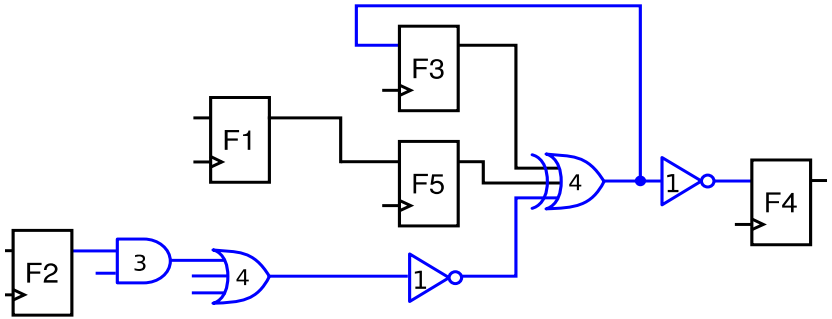
Retiming



T_{min}= 3+7+1=11

The limit in the traditional timing paradigm

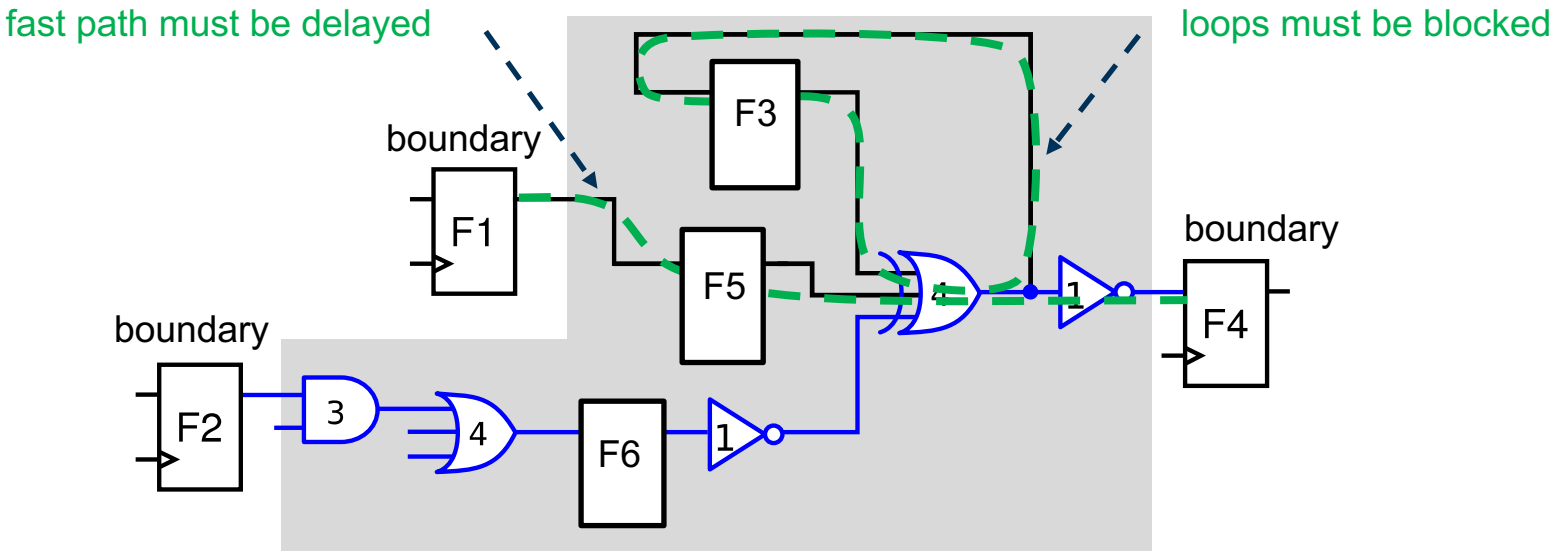
VirtualSync



T_{min}=(3+13+1)/2=8.5

22.7% reduction compared with retiming&sizing

VirtualSync Concept

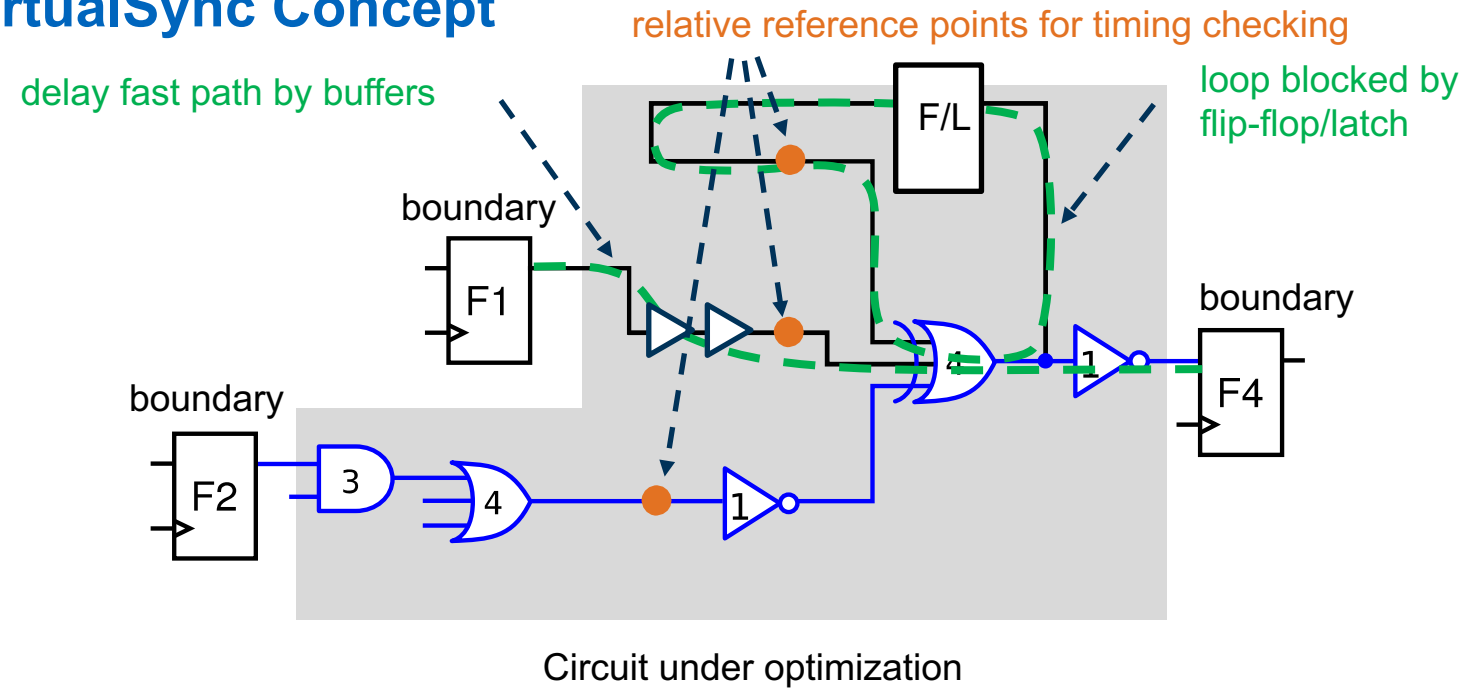


Circuit under optimization

VirtualSync:
Step 1: Remove all flip-flops except those at the boundary of the module
Step 2: Block fast signals for timing synchronization, including

- signals arriving at boundary flip-flops too early through fast paths
- signals traveling across combinational loops

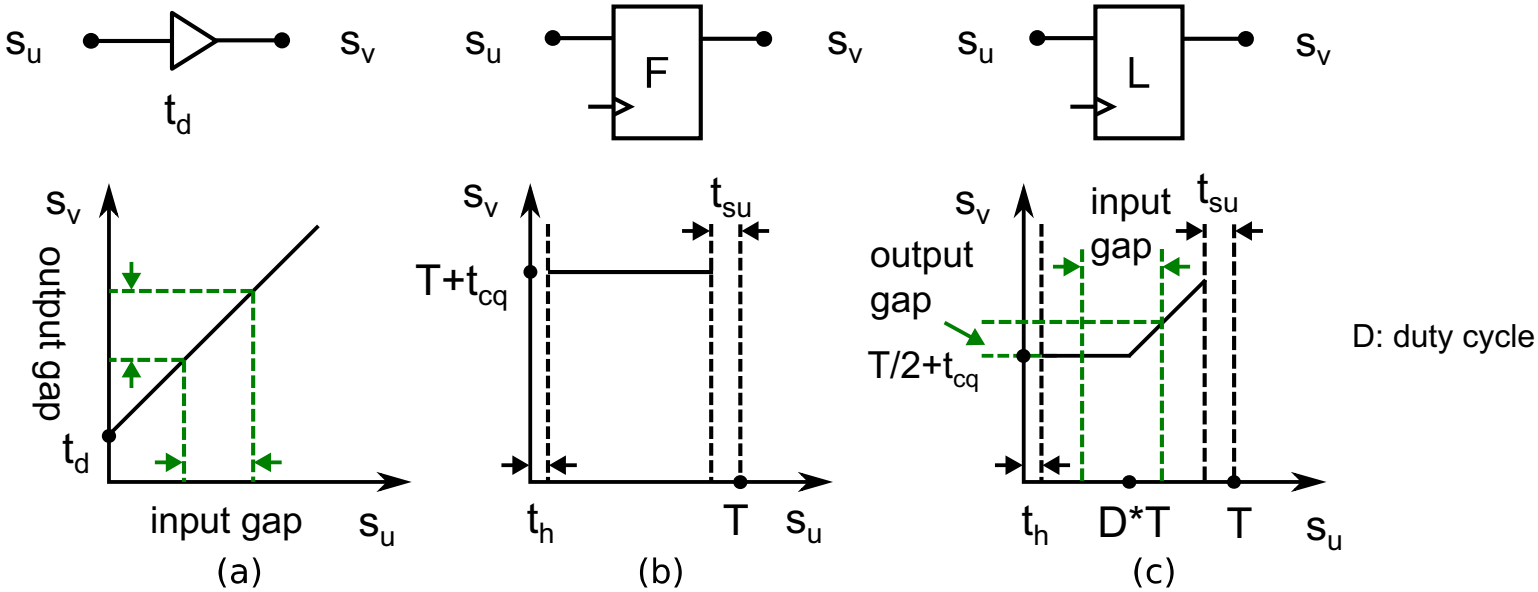
VirtualSync Concept



Circuit under optimization

- **Delay units (logic gates, flip-flops and latches)** are used to slow down signals on fast paths and loops.
- **Relative reference points** provide relative timing information.

Delay Units in VirtualSync



Linear delaying effect of a **combinational delay unit**

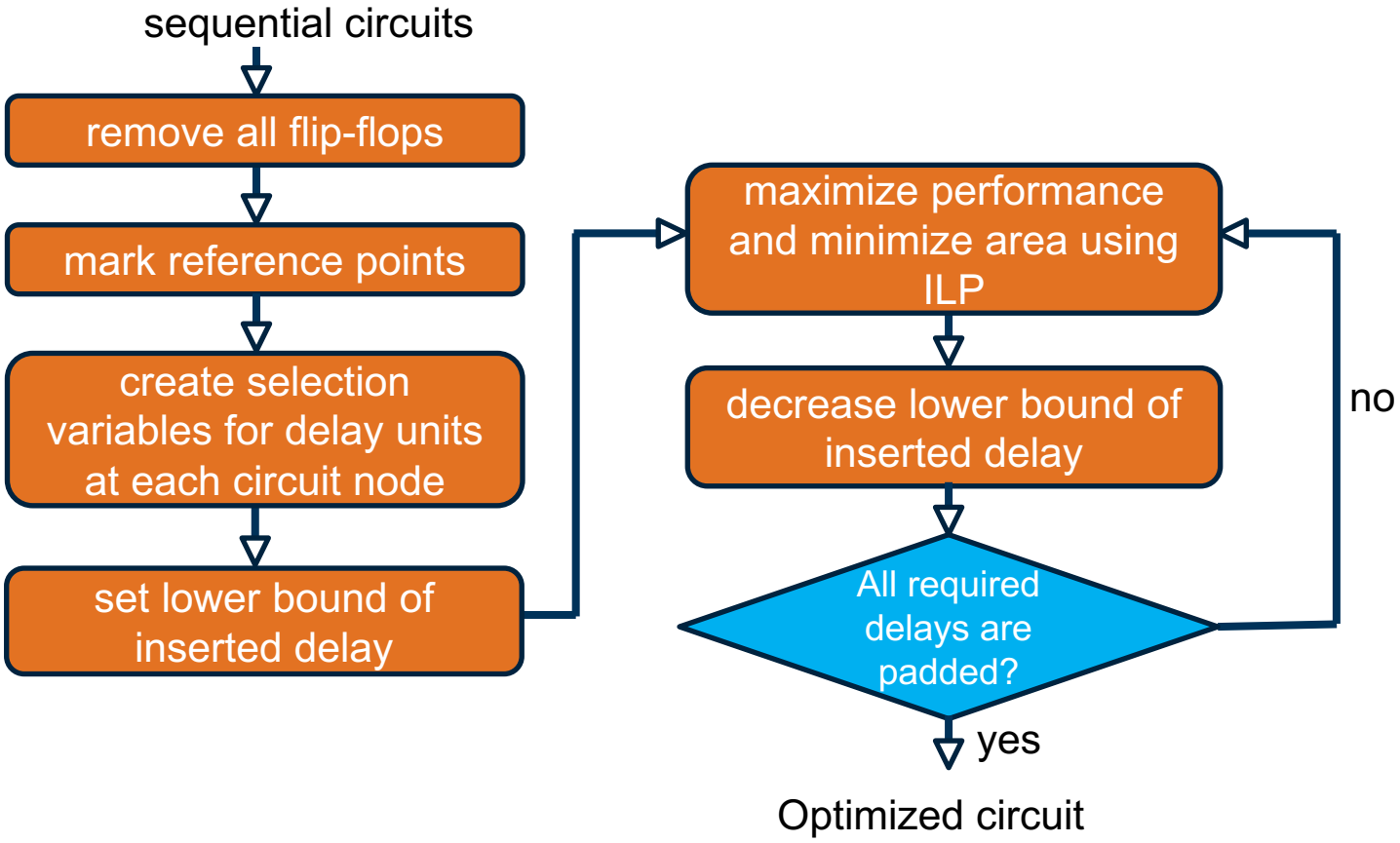
Constant delaying effect of a **flip-flop**

Piecewise delaying effect of a **latch**

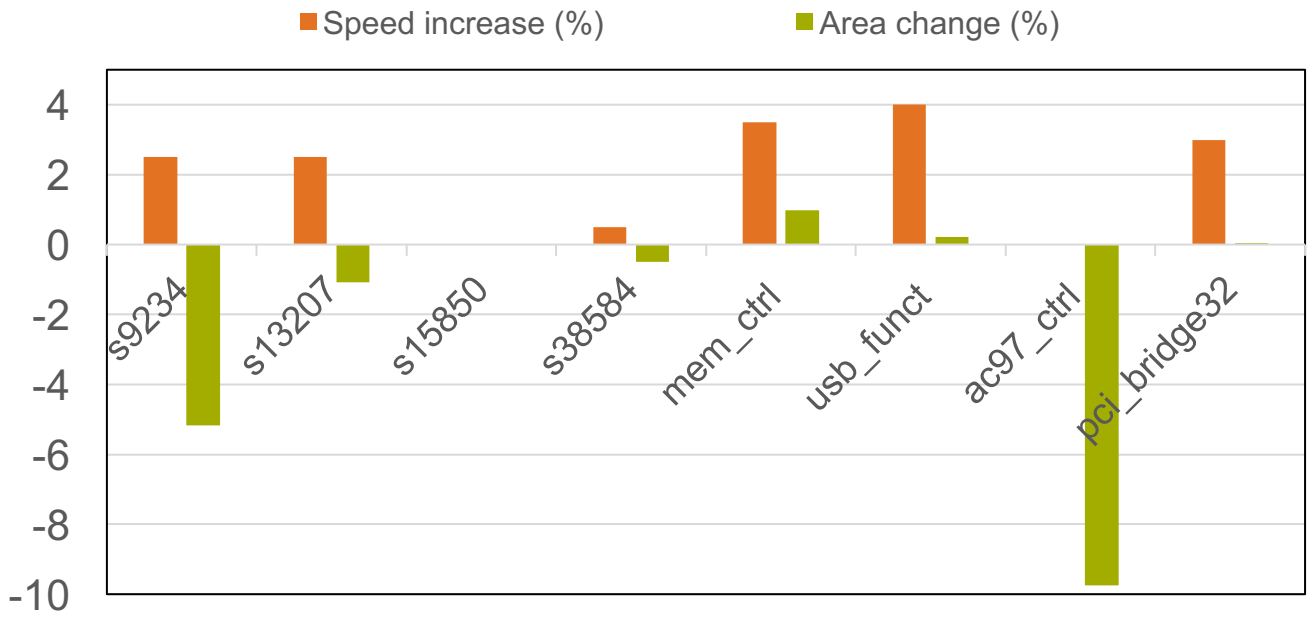
D: duty cycle

Input gap: the difference of arrival times of two signals at a delay unit
Output gap: the difference between their arrival times after they pass through the unit

Overall Flow of VirtualSync



Results of VirtualSync



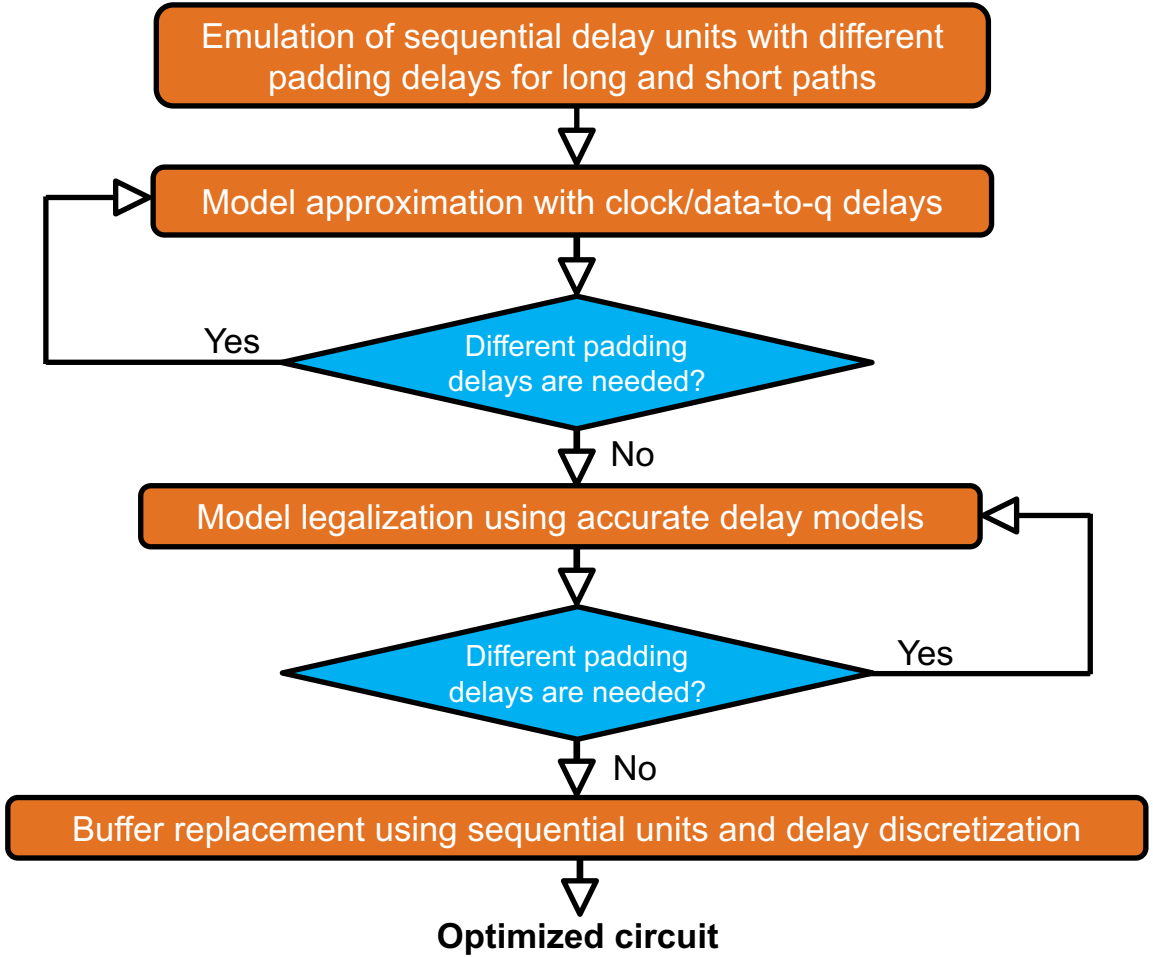
Speed increase and area results compared with ideally balanced design

Summary

- A new timing model, VirtualSync, with sequential components and combinational logic gates as delay units is proposed.
- By viewing flip-flops and latches as delay units, circuit performance can be pushed even beyond the limit of the traditional timing paradigm.
- VirtualSync demonstrates a good potential for high-performance designs.

Thank you for your attention!

Heuristic method in VirtualSync

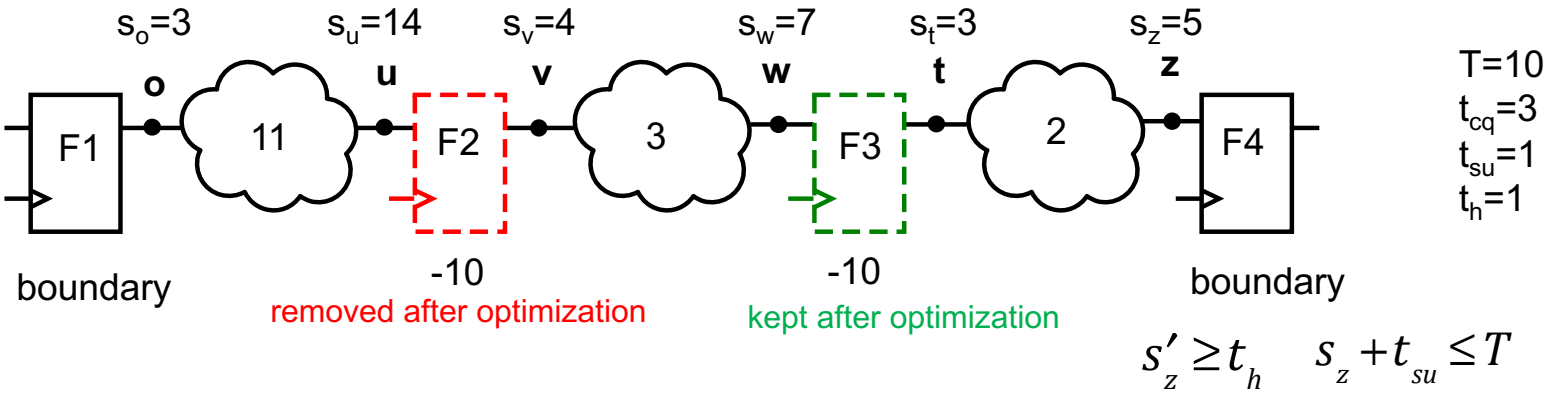


Results of VirtualSync

Circuit	Critical part		Optimized circuit			Comparison	
	#flip-flop	#gates	#flip-flop	#latch	#buffer	clock period reduction	area increase
s5378	35	1877	11	14	94	**11.5%**	2.84%
s9234	91	3981	58	45	91	2.5%	-5.17%
s13207	191	3483	95	73	52	2.5%	-1.09%
s15850	71	3847	72	18	26	0%	6.01%
s38584	126	9498	62	75	46	0.5%	-0.5%
systemcdes	92	3232	90	81	227	3.5%	2.43%
mem_ctrl	136	7500	101	39	140	3.5%	0.97%
usb_funct	138	5378	123	37	60	4%	0.21%
ac97_ctrl	237	4873	42	172	218	0%	-9.76%
pci_bridge	239	9510	188	68	338	3%	0.05%

The comparison was made with extreme retiming and sizing, with which the timing performance has reached the limit in the traditional timing paradigm.

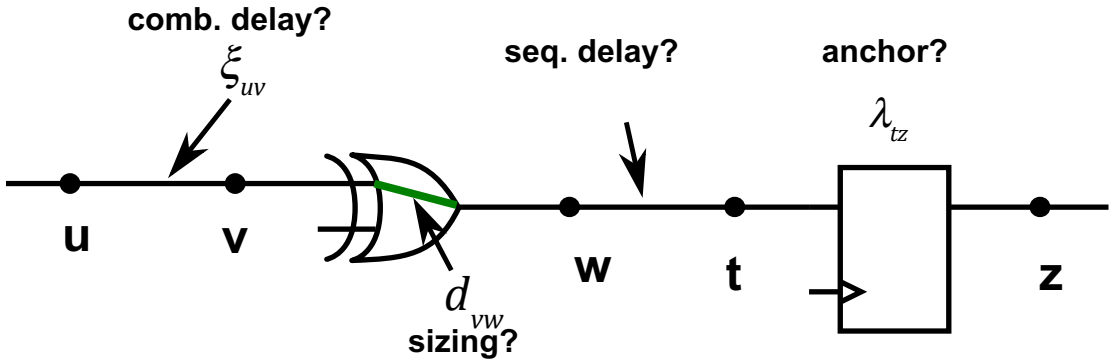
Relative Timing References in VirtualSync



- The location of the removed flip-flops such as F2 and F3 are called ***anchor points***.
- The ***anchor points*** allow to relate timing information to boundary flip-flops. Every time when a signal passes an ***anchor point***, its arrival time is converted by subtracting T.
- If F3 is removed, the arrival time s_z becomes -3+2=-1, violating the hold time constraint.

The timing constraints at the boundary flip-flops force the usage of the internal sequential delay units!

Synchronizing Logic Waves by Delay Units



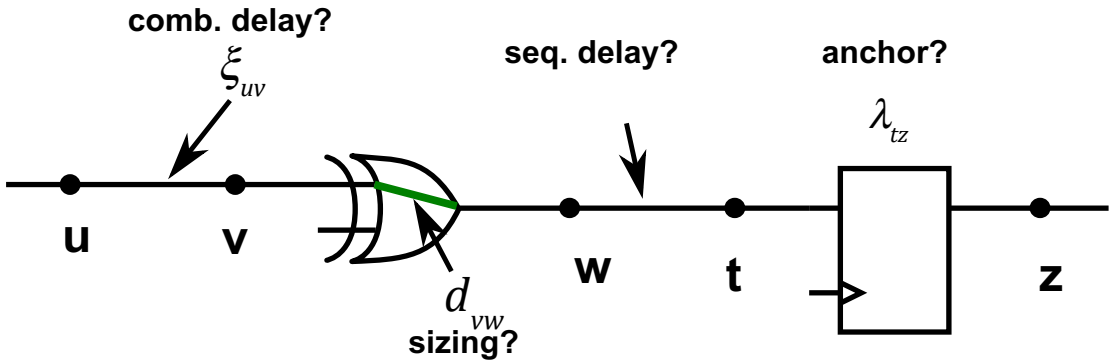
1. Combinational delay unit and gate sizing

$$s_w \geq s_u + \xi_{uv} * r^u + d_{vw} * r^u \quad (1)$$

$$s'_w \leq s'_u + \xi_{uv} * r^l + d_{vw} * r^l \quad (2)$$

s_u, s'_u, s_w, s'_w are the latest and earliest arrival time of node u and w.
ξ_{uv} is the delay of an inserted buffer.
r^u and r^l are two constants to reserve a guard band for process variations.

Synchronizing Logic Waves by Delay Units



2. Insertion of sequential delay units

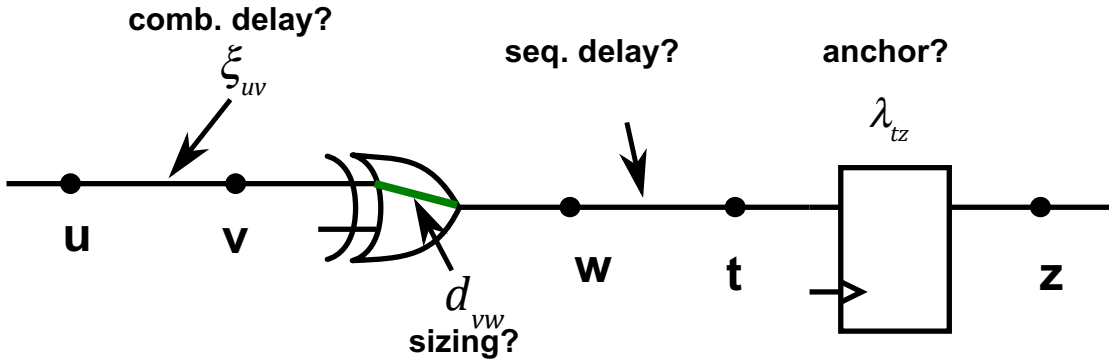
Case 1: ***No sequential delay unit*** is inserted between w and t

$$s_t \geq s_w \quad (3)$$

$$s'_t \leq s'_w \quad (4)$$

s_t, s'_t, s_w, s'_w are the latest and earliest arrival time of node t and w.

Synchronizing Logic Waves by Delay Units



2. Insertion of sequential delay units

Case 1: ***A flip-flop*** is inserted between w and t

$$s_w, s'_w \geq N_{wt} * T + \phi_{wt} + t_h * r^u \quad (5)$$

$$s_w, s'_w \leq (N_{wt} + 1) * T + \phi_{wt} - t_{su} * r^u \quad (6)$$

$$s_t \geq (N_{wt} + 1) * T + \phi_{wt} + t_{cq} * r^u \quad (7)$$

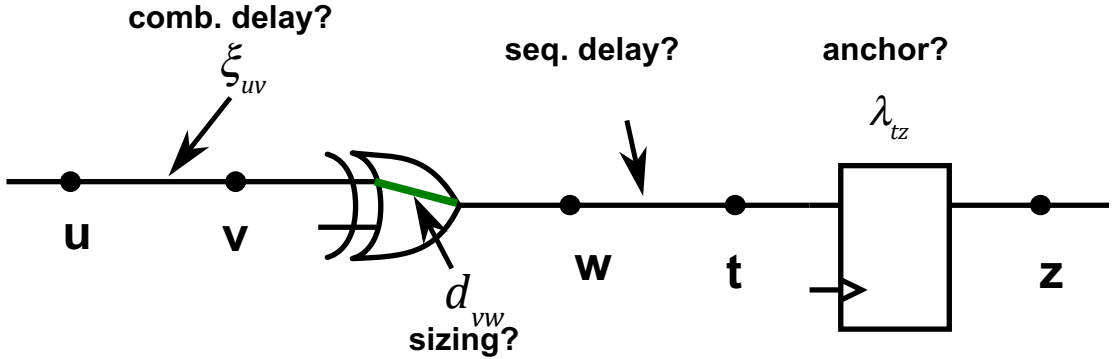
$$s'_t \leq (N_{wt} + 1) * T + \phi_{wt} + t_{cq} * r^l \quad (8)$$

ϕ_{wt} is the phase shift of the clock signal

A flip-flop only works in a region t_h after the rising clock edge and t_{su} before the next rising clock edge.

The signal always starts to propagate from the next active clock edge.

Synchronizing Logic Waves by Delay Units



2. Insertion of sequential delay units

Case 1: ***A level-sensitive latch*** is inserted between w and t

$$s_t \geq N_{wt} * T + \phi_{wt} + D * T + t_{cq} * r^u \quad (9)$$

$$s_t \geq s_w + t_{dq} * r^u \quad (10)$$

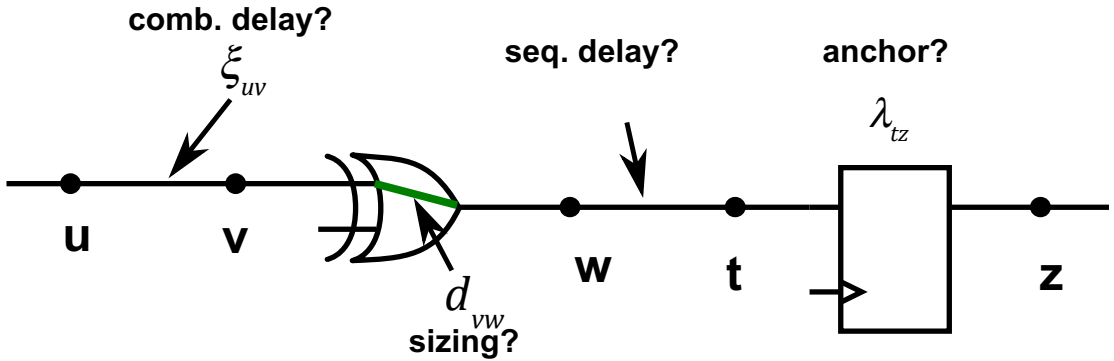
$$s'_t \leq \max(N_{wt} * T + \phi_{wt} + D * T + t_{cq} * r^l, \; s'_w + t_{dq} * r^l) \quad (11)$$

D is the duty cycle of the clock signal

The upper is the case that the latch is non-transparent; the lower is the case that the latch is transparent.

The signal starts to propagate from the maximum of the earliest time.

Synchronizing Logic Waves by Delay Units



3. Reference shift with respect to anchor points

$$s_z = s_t - \lambda_{tz} * T \qquad (12)$$

4.Wave non-interference condition

$$s_u + t_{stable} \le s'_u + T \qquad (13)$$

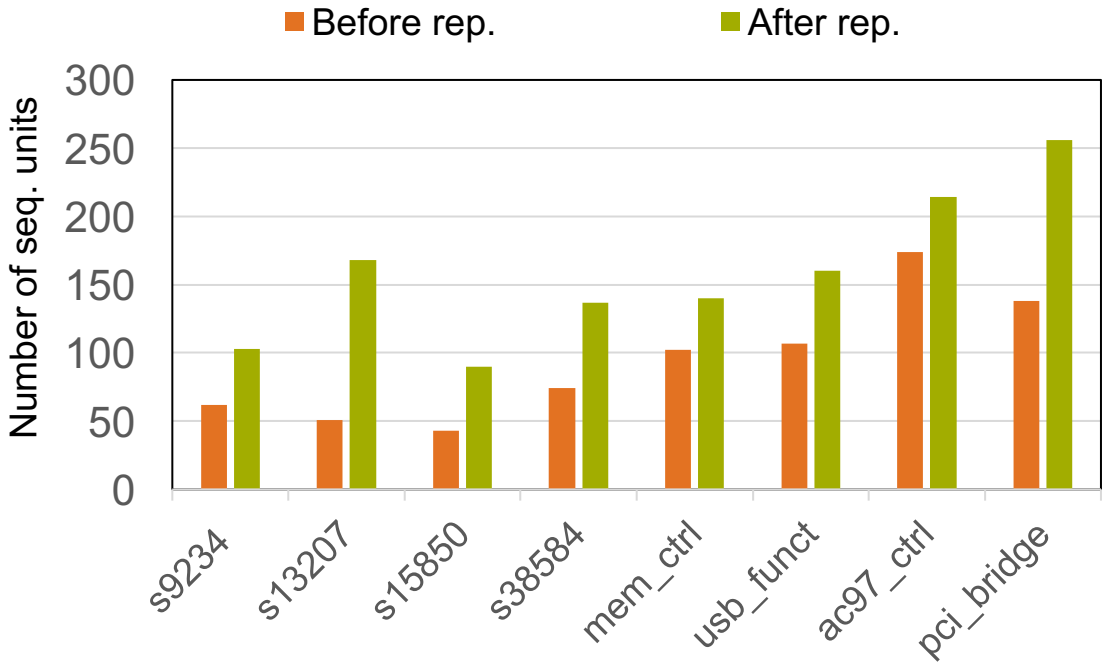
Overall formulation

Objective: find a solution to make the circuit work at a given clock period

Subject to: (1)-(13) and setup and hold time constraints at the boundary flip-flops

NP-hard!

Results of seq. delay units after buffer replacement



Runtime

Circuit	T_r(s)
s5378	121.6
s9234	7251.1
s13207	3121.6
s15850	289.97
s38584	1142.3
systemcdes	7310.5
mem_ctrl	3750.1
usb_funct	1211.7
ac97_ctrl	2936.8
pci_bridge	7418.5