

ESE370: Circuit-Level Modeling, Design, and Optimization for Digital Systems

Lec 24: November 5, 2021

Synchronous Timing Discipline and
Clocking

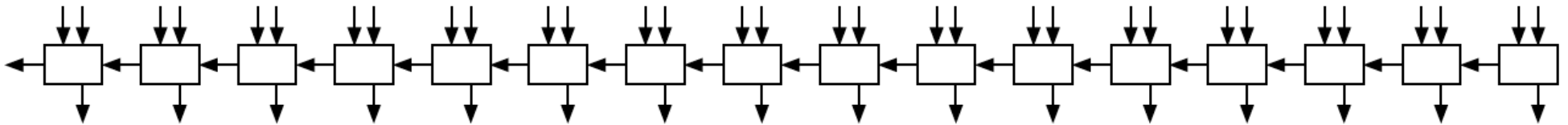


Today

- ❑ Managing Timing
- ❑ Latches
- ❑ Registers, clocking

Timing Setup (Preclass1&2)

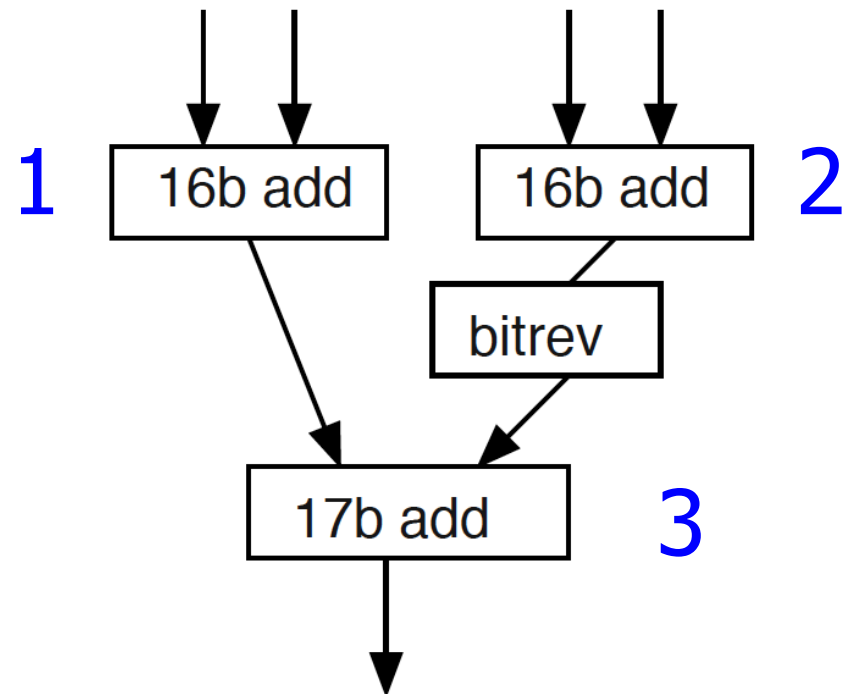
- Delay from 3 adder-inputs to 2 adder-outputbits is T_{bit}
 - i.e single bitslice delay is T_{bit}



- Worst-case delay to output?
- Shortest delay to output?
 - assuming the input switches

Timing Setup (Preclass 3)

- New set of inputs every $20T_{\text{bit}}$
 - How does it behave?
- bitrev: $\text{Out}[i] = \text{In}[17-i]$
 - Assume delay of bitrev $\ll T_{\text{bit}}$





Challenge

- ❑ Logic paths have different delays
 - E.g. different output bits in an adder
- ❑ Delay of signal is data dependent
 - E.g. length of carry propagation
- ❑ Delay is chip dependent
 - E.g. Threshold Variation
- ❑ Delay is environment dependent
 - E.g. Temperature



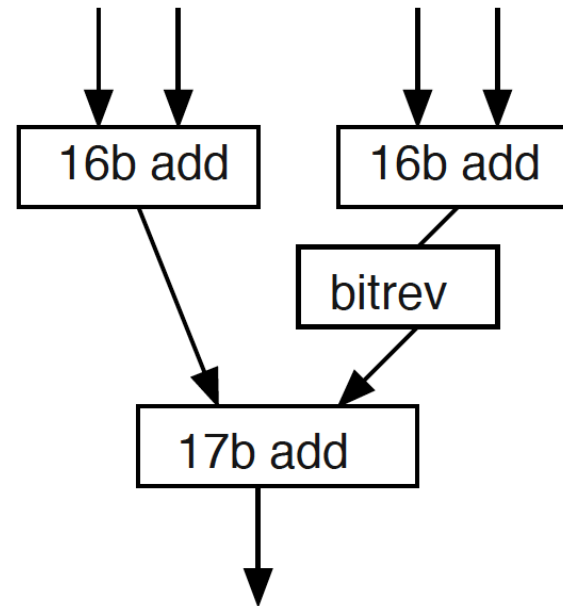
Challenge

- ❑ Logic paths have different delays
 - E.g. different output bits in an adder
- ❑ Delay of signal is data dependent
 - E.g. length of carry propagation
- ❑ Delay is chip dependent
 - E.g. Threshold Variation
- ❑ Delay is environment dependent
 - E.g. Temperature
- ❑ Proper behavior depends on inputs being coordinated



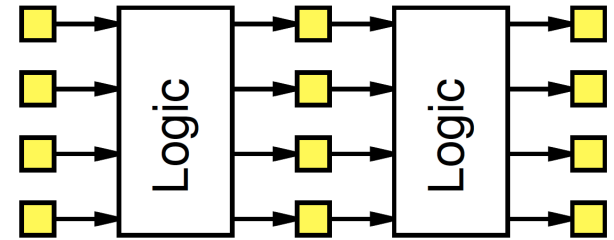
Logic Timing

- How do we fix this?
 - Make it possible to input a new value every $20T_{\text{bit}}$?



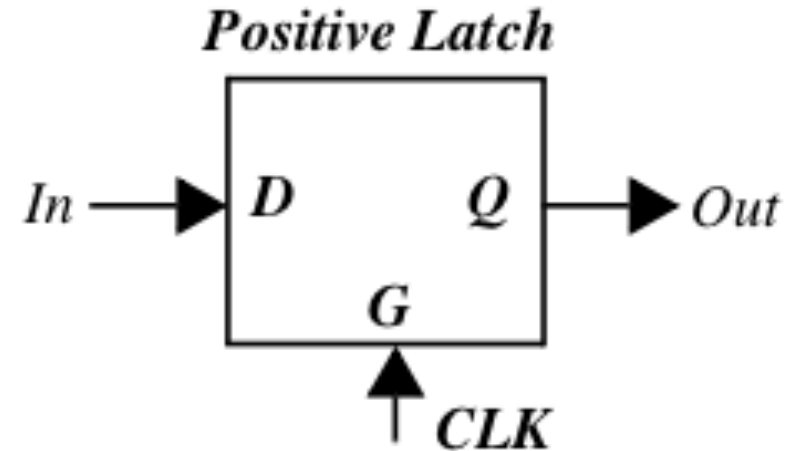
Discipline

- ❑ Add circuit elements to
 - hold values
 - and change at coordinated point
 - Control when changes seen by circuit
- ❑ Only have to make sure to **wait long enough** for all results
- ❑ Decouple outputs and inputs
 - timing of signal change (settling of output change)...
 - ...from timing of signal usage (output used as input)

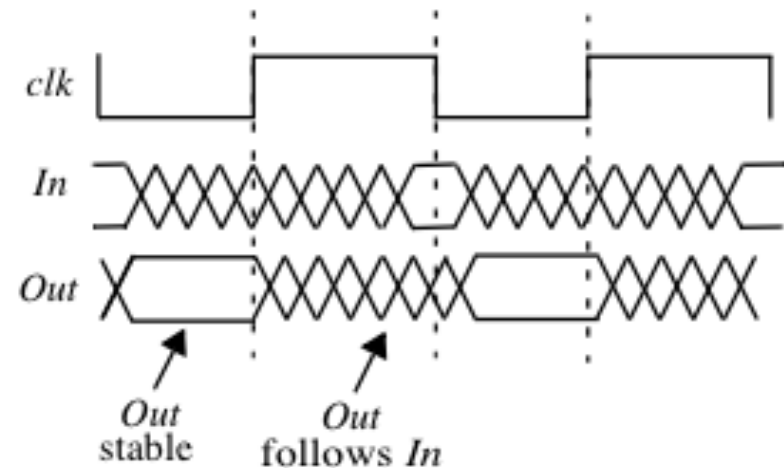


Latch

- ❑ *Level-sensitive device*
- ❑ Positive Latch
 - Output follows input if CLK high
- ❑ Negative Latch
 - Output follows input if CLK low

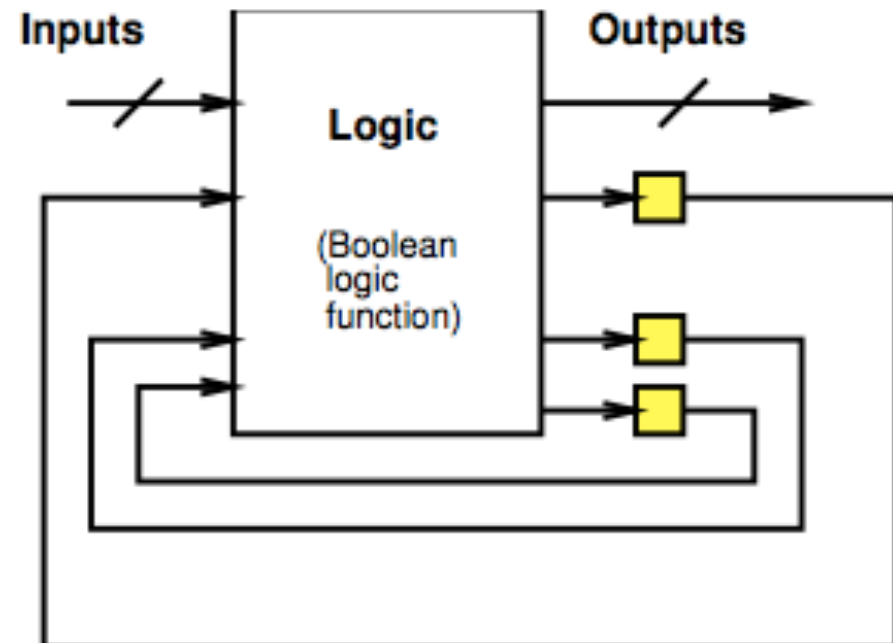
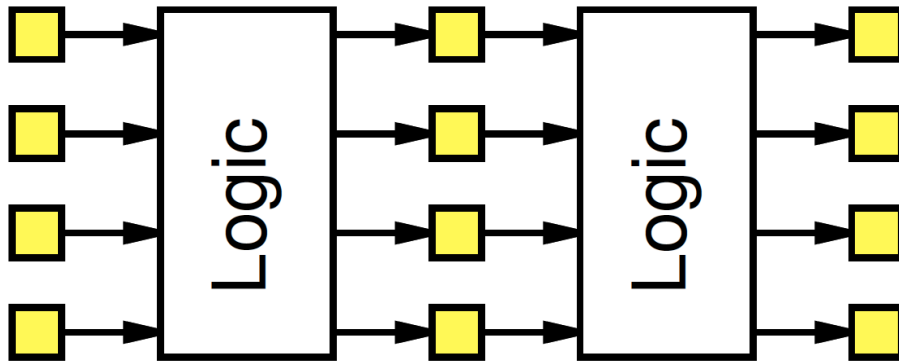


$$Q = \overline{CLK} \cdot Q + CLK \cdot In$$



Synchronous Discipline

- ❑ Add state elements (registers, latches)
- ❑ Compute
 - From state elements
 - Through combinational logic
 - To new values for state elements





Latch

- ❑ Build with combinational logic
- ❑ Build with pass logic



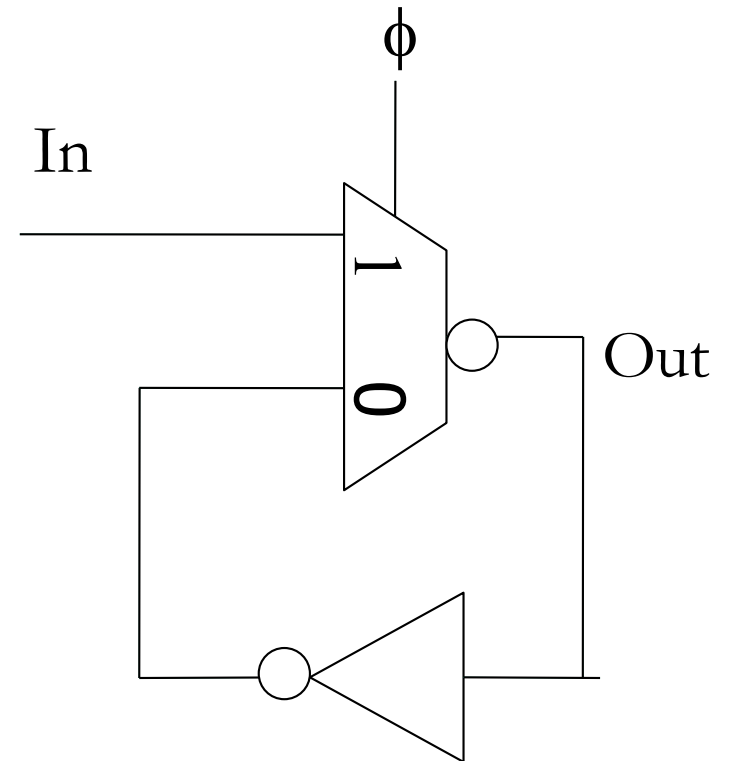
Latch (Preclass 4a)

- $\phi=1 \rightarrow \text{Out}=\text{In}$
- $\phi=0 \rightarrow \text{Out}=\text{Out}$

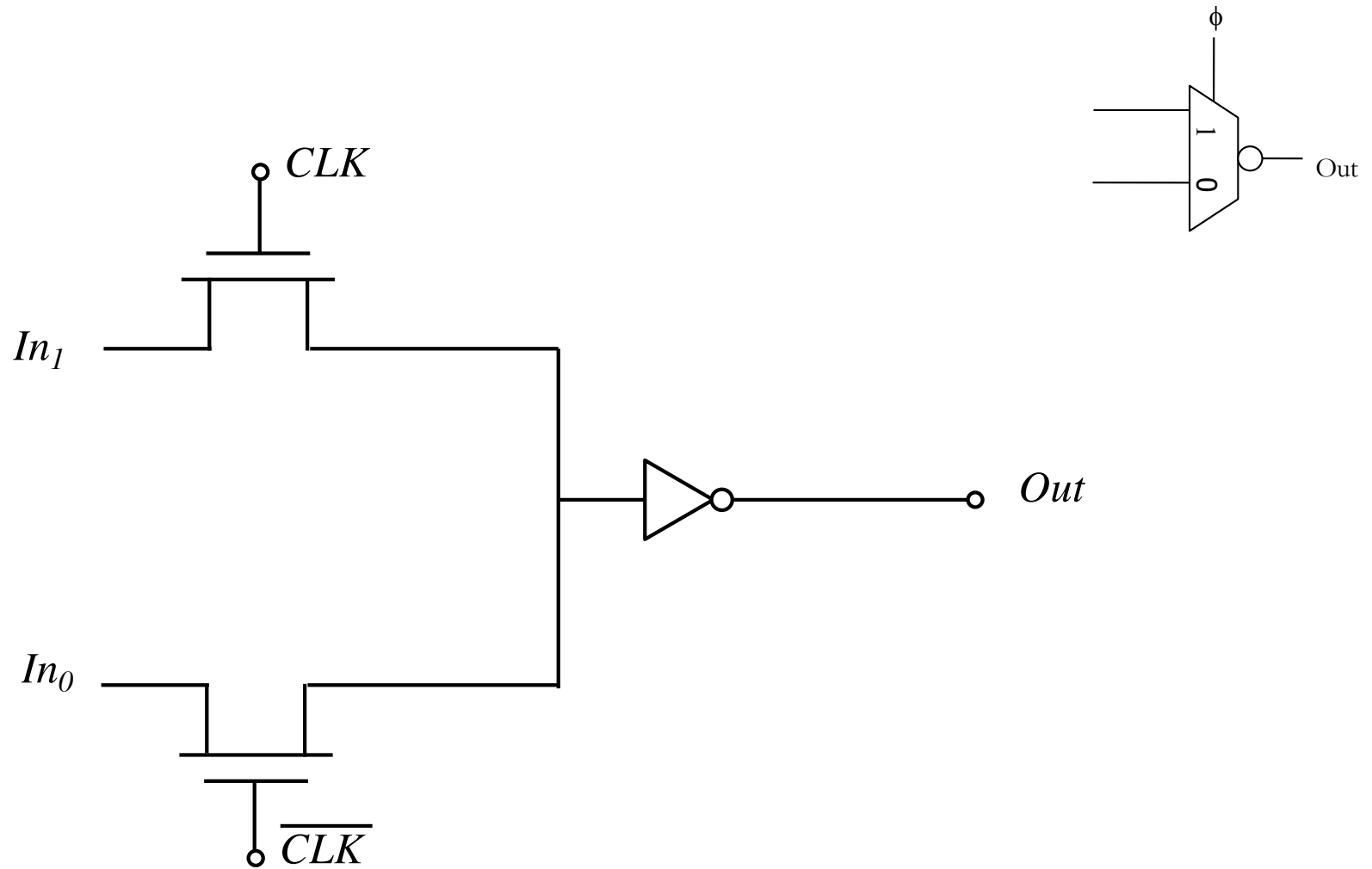
- How do we build a latch from combinational logic?
 - At gate level

Multiplexer-based Latch from CMOS Logic

- $\phi=1 \rightarrow \text{Out}=\text{/In}$
- $\phi=0 \rightarrow \text{Out}=\text{Out}$

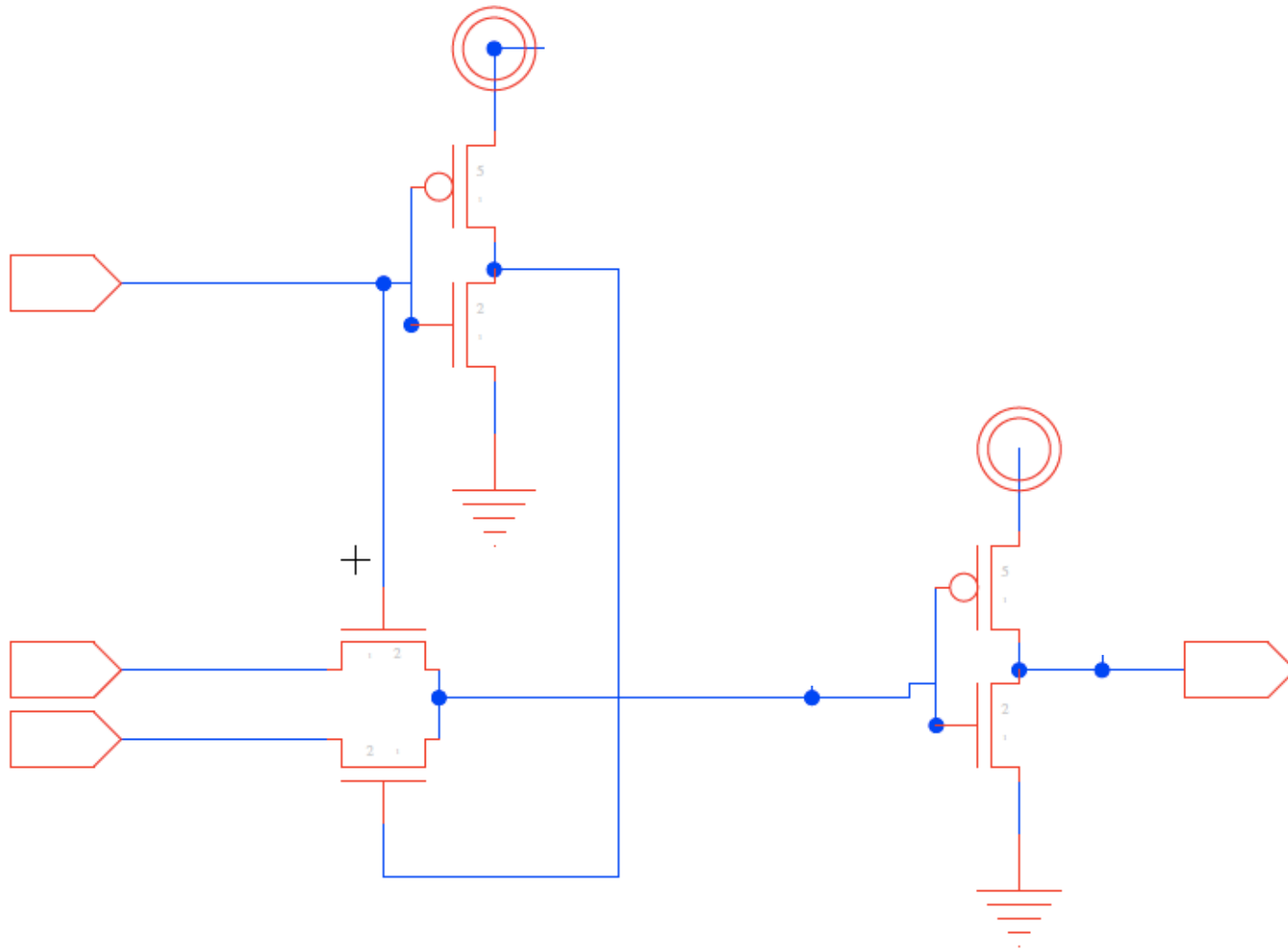


Mux at Transistor Level

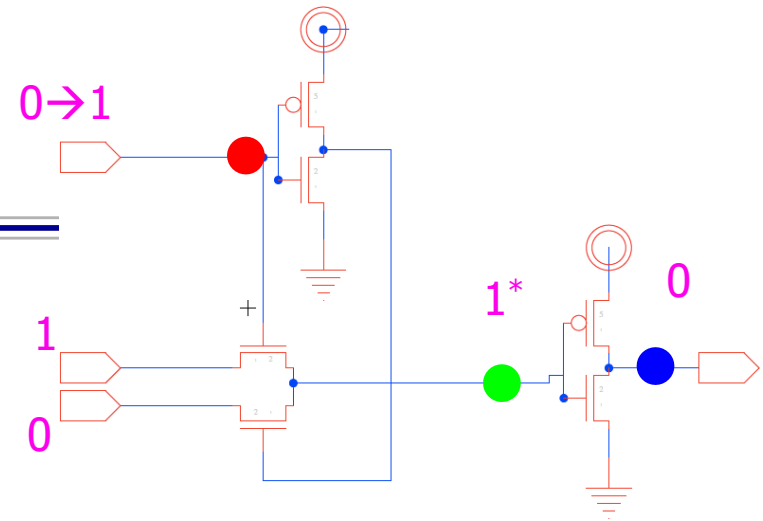




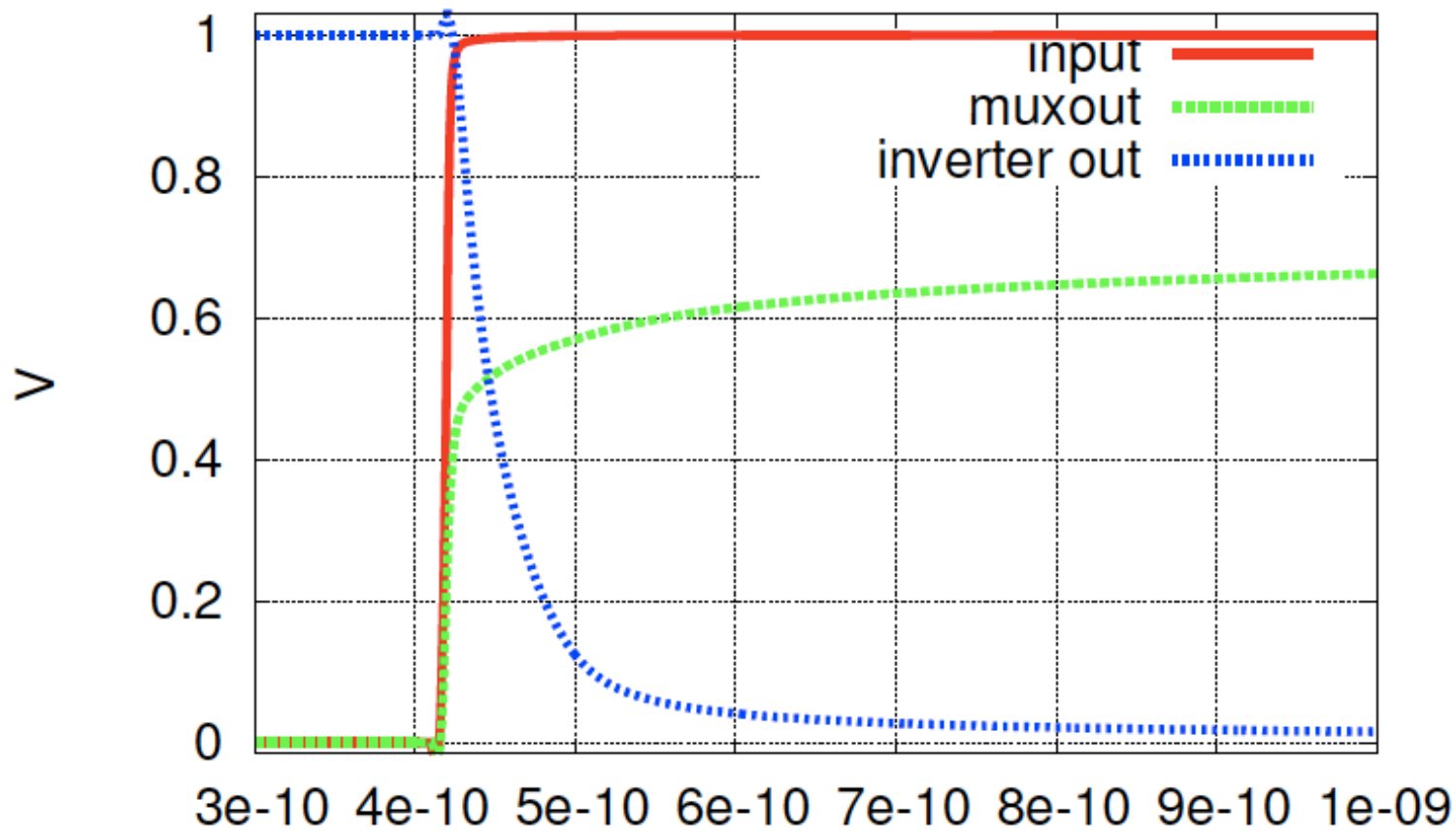
MuxL



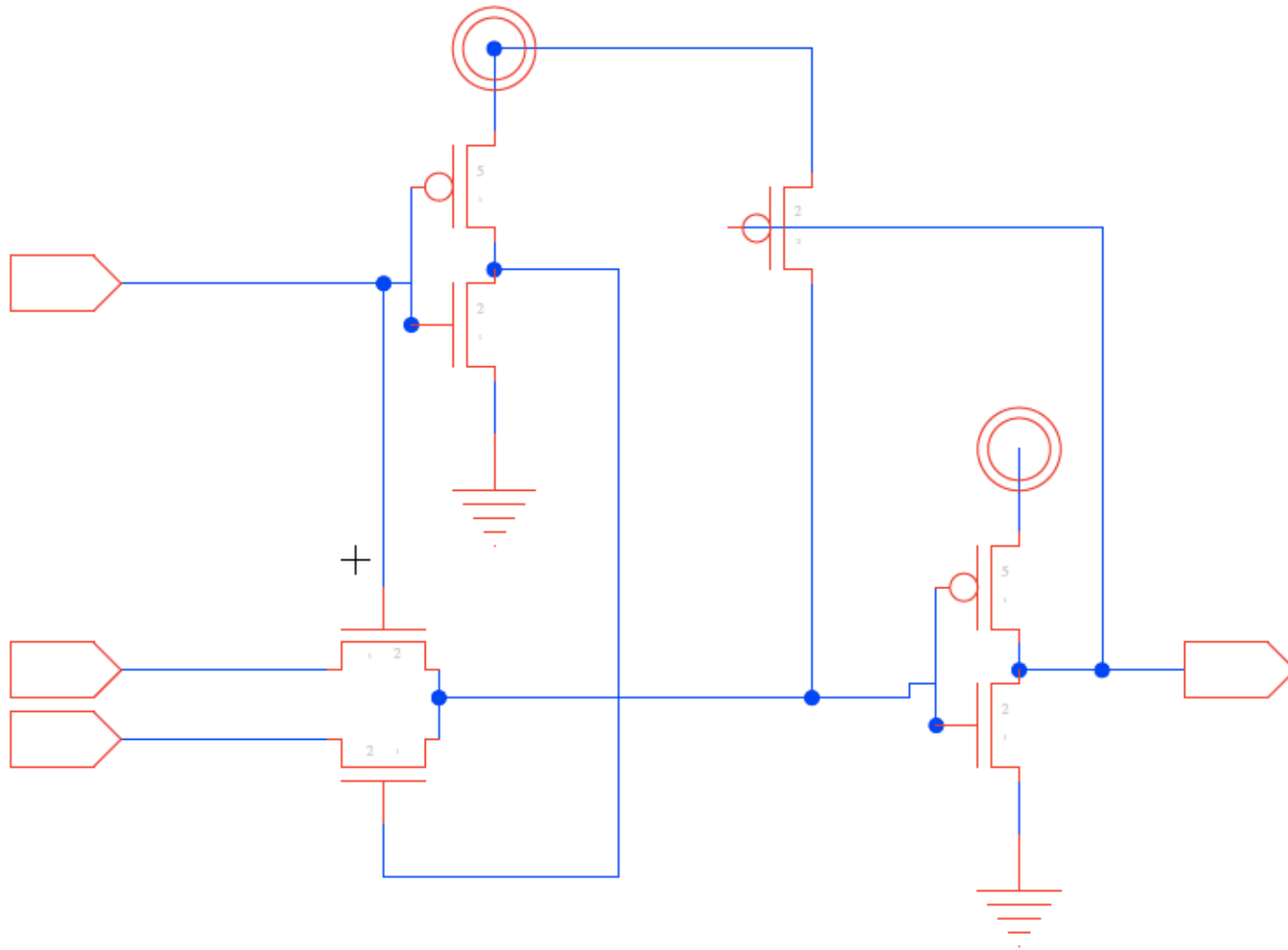
Without level restorer



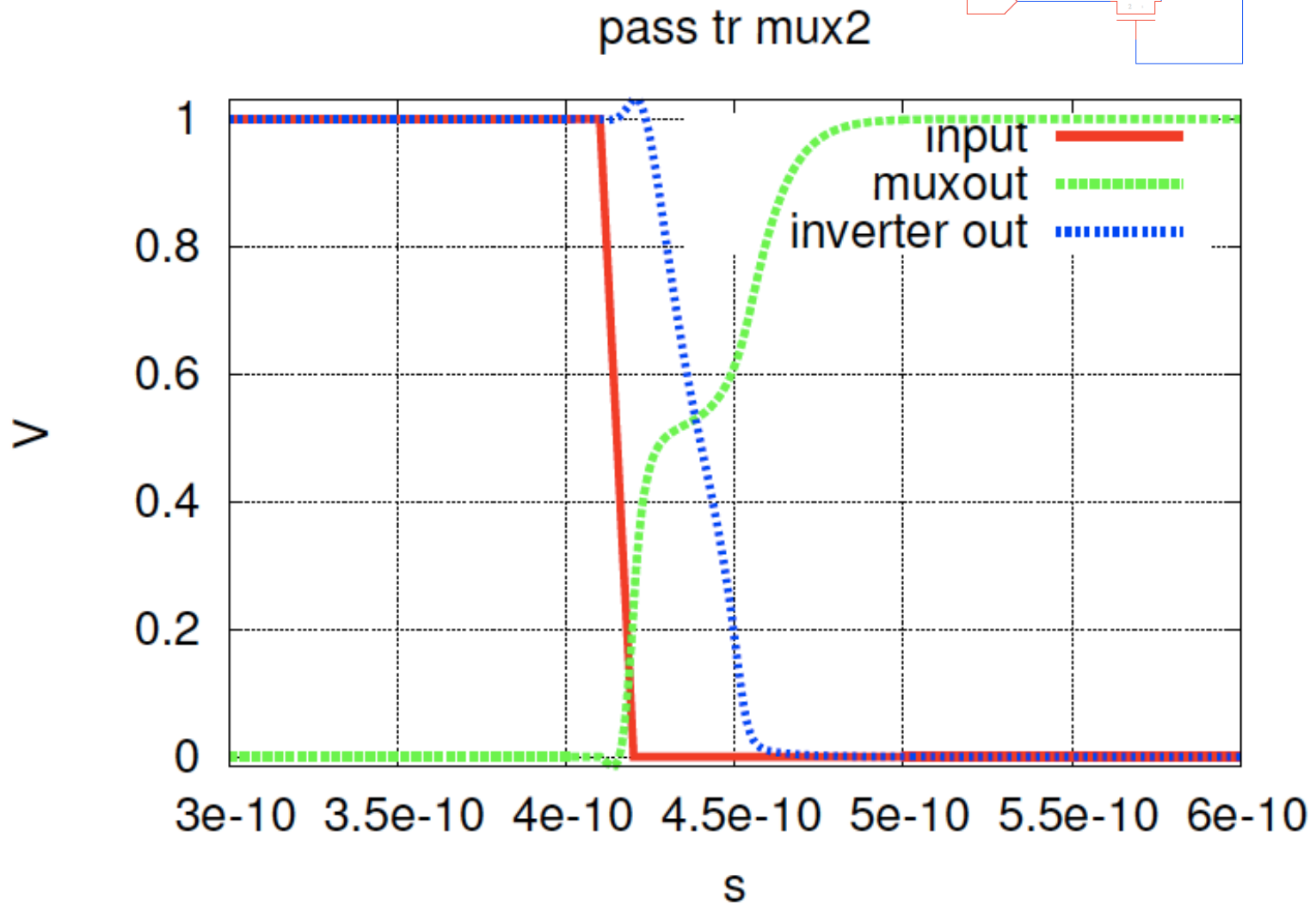
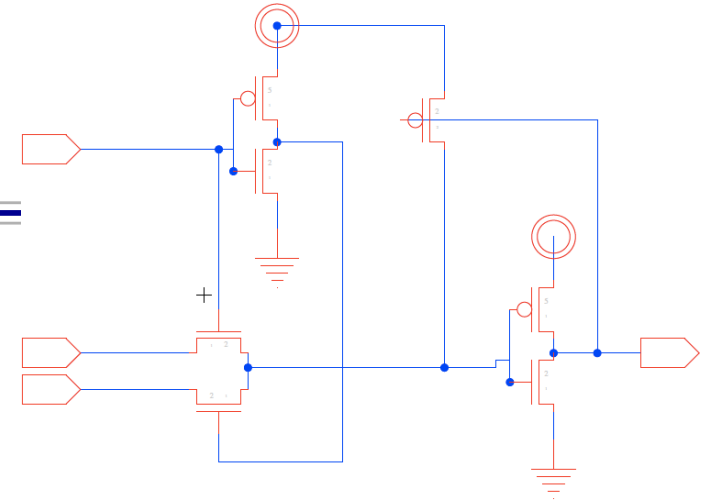
pass tr mux2



MuxL Level Restorer (“Staticizer”)



With level restorer



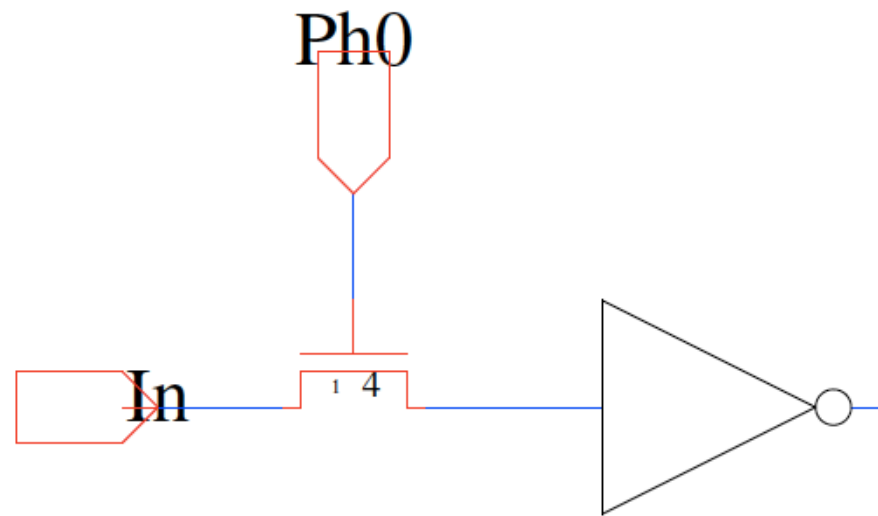


Latch (Preclass 4b)

- $\phi=1 \rightarrow \text{Out}=\text{In}$
 - $\phi=0 \rightarrow \text{Out}=\text{Out}$
 - ϕ transitions $1 \rightarrow 0$ Out holds value
-
- How do we build a latch from pass transistors?

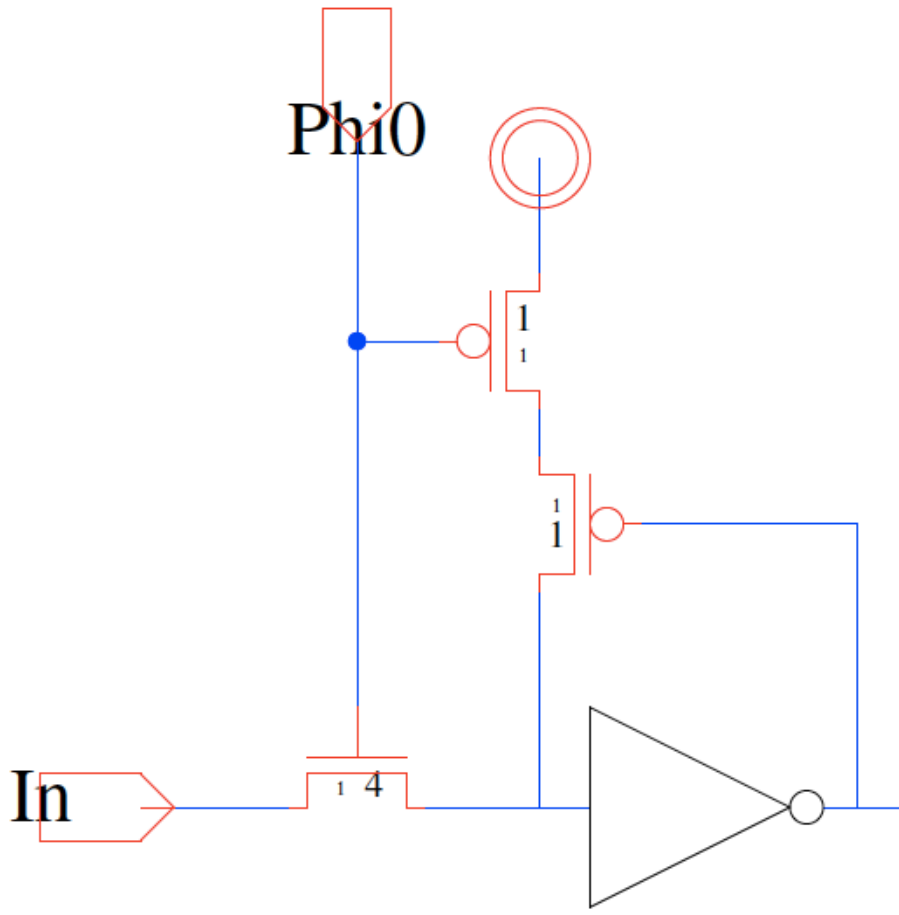


Simple Latch

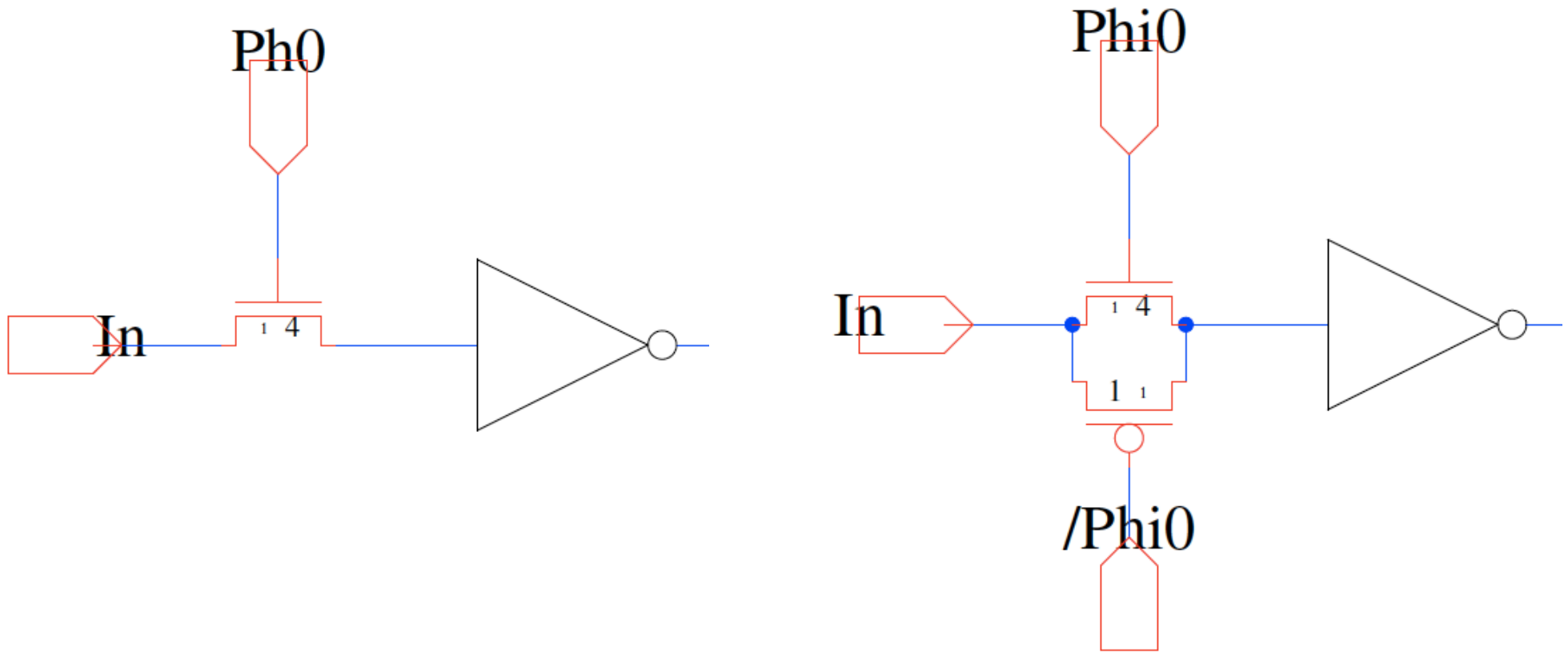




Latch with Level Restore



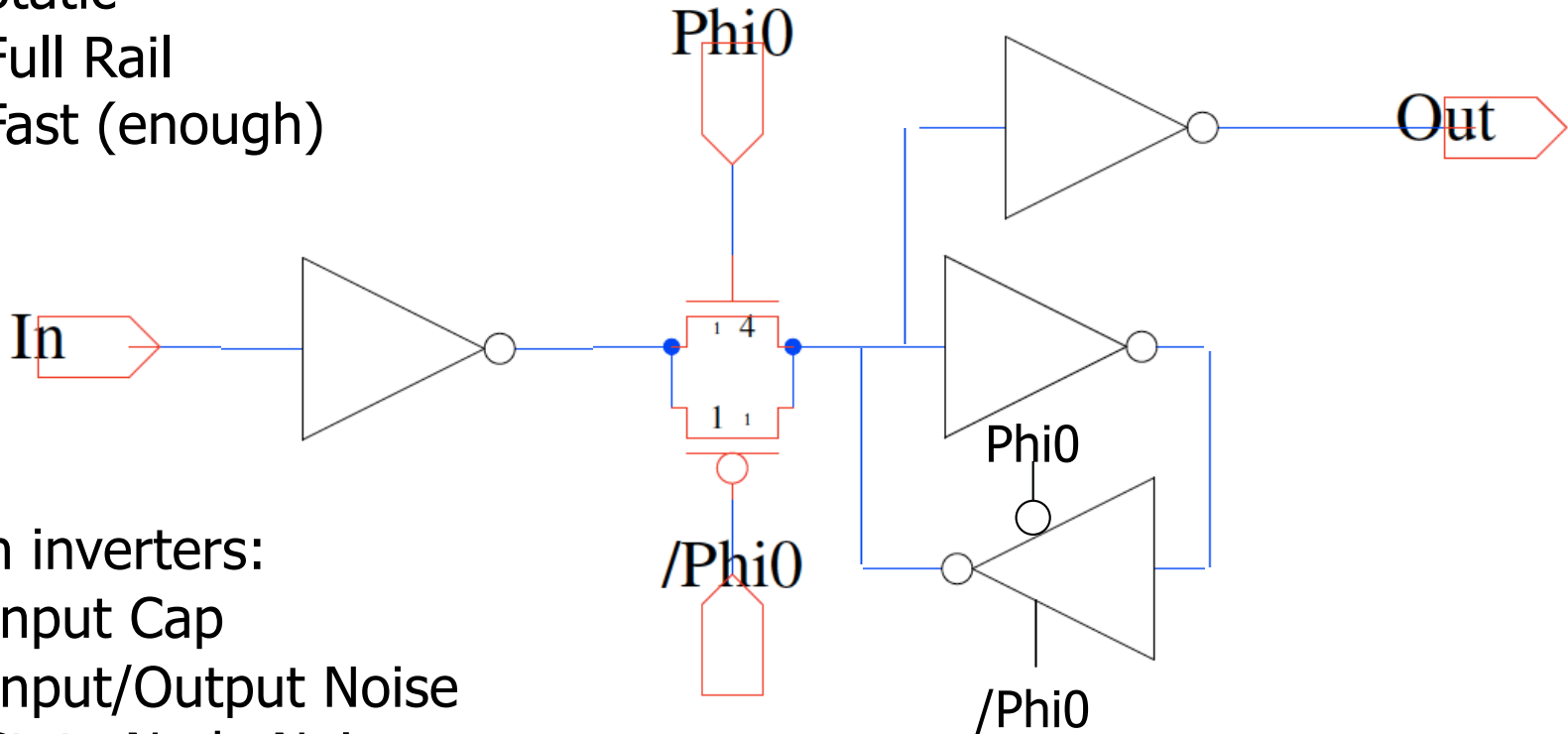
What is the difference?



Typical Static Latch

Advantages: VERY ROBUST

- Static
- Full Rail
- Fast (enough)



Isolation inverters:

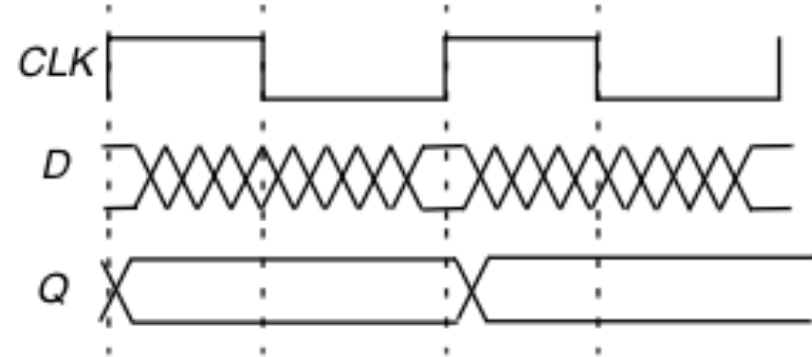
- Input Cap
- Input/Output Noise
- State Node Noise

Disadvantages:

- Large
- High clock loading

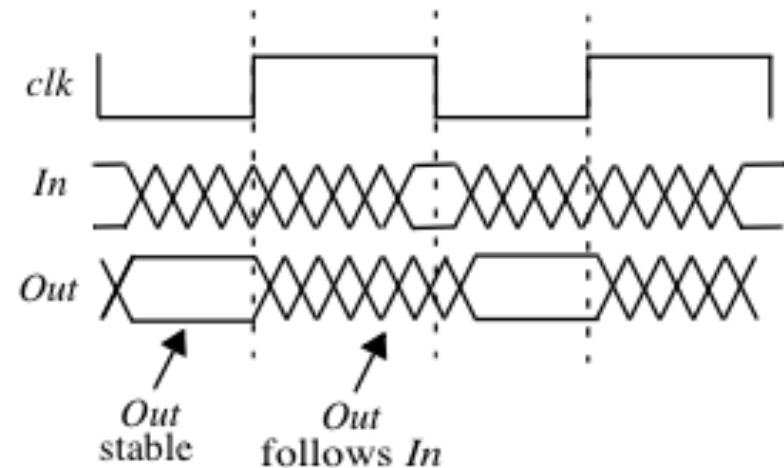
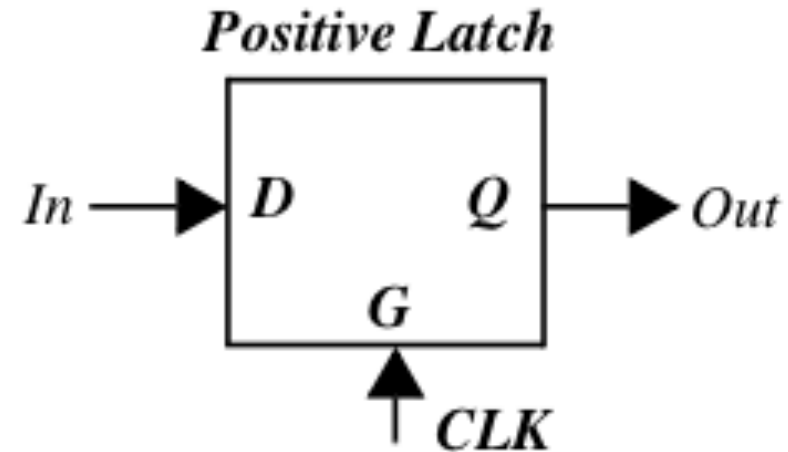
Register

- ❑ *Edge-triggered storage element*
- ❑ Positive edge-triggered
 - Input sampled on rising CLK edge
- ❑ Negative edge-triggered
 - Input sampled on falling CLK edge



Latch

- ❑ *Level-sensitive device*
- ❑ Positive Latch
 - Output follows input if CLK high
- ❑ Negative Latch
 - Output follows input if CLK low



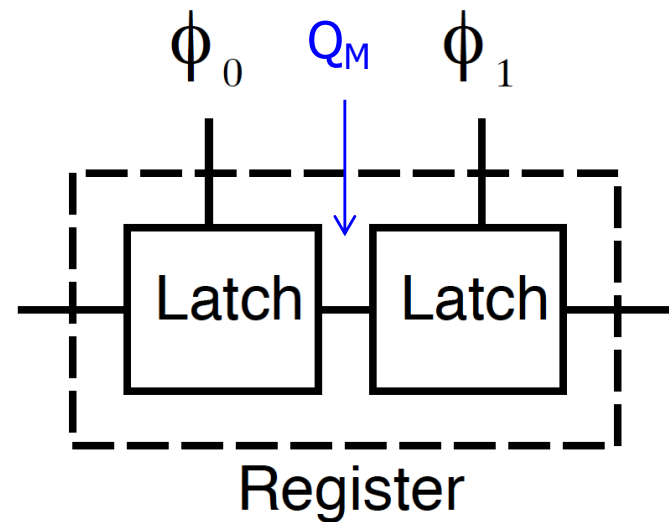
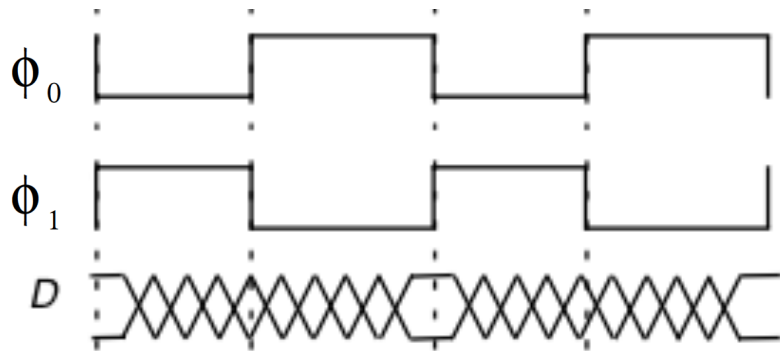


Shift Register

- How do you make a shift register out of latches?

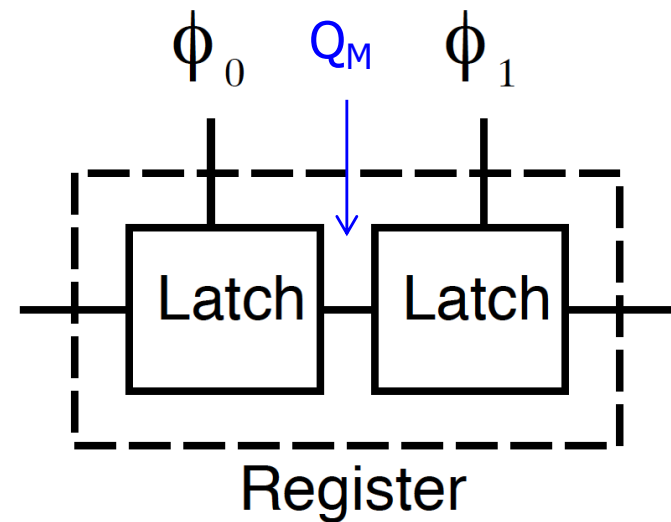
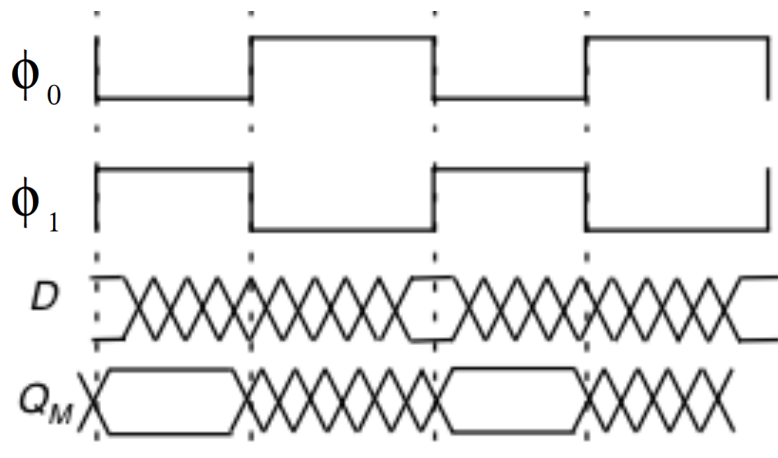
Positive-Edge Triggered Register

- ❑ Build register from pair of latches
- ❑ What happens when ϕ_0 is high?
- ❑ What happens when ϕ_1 is high?



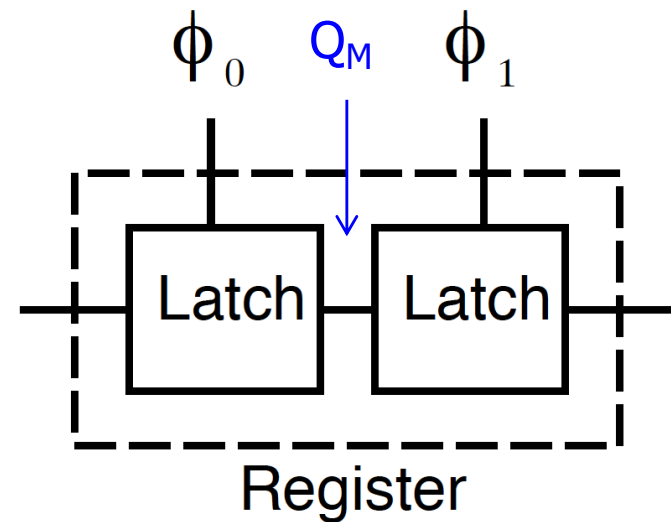
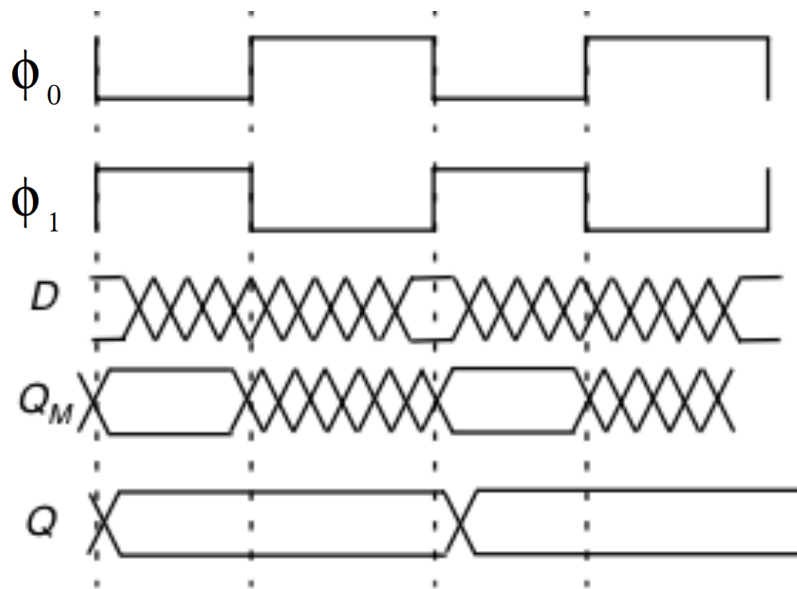
Positive-Edge Triggered Register

- Build register from pair of latches



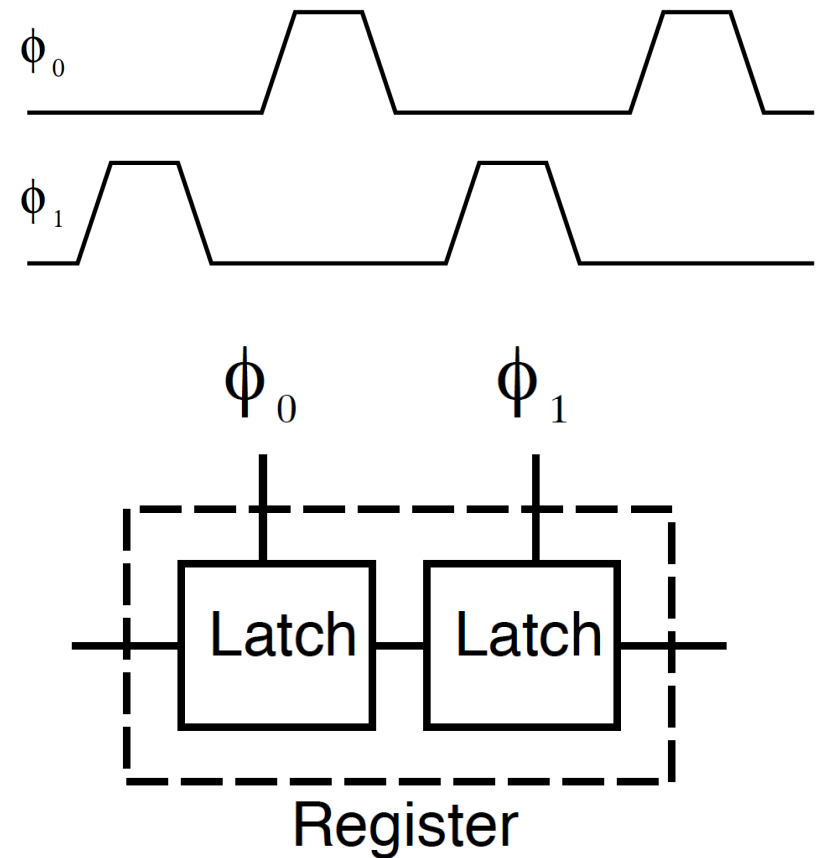
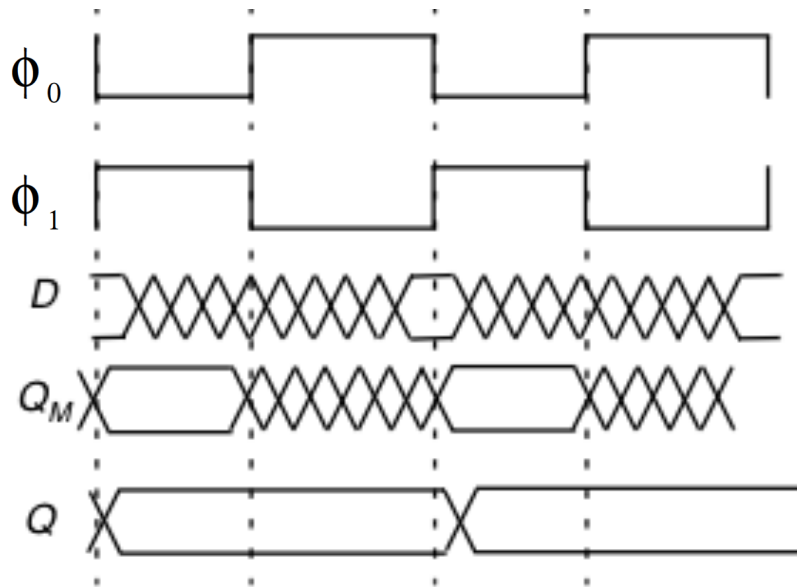
Positive-Edge Triggered Register

- ❑ Build register from pair of latches
- ❑ What could go wrong if clocks overlap?



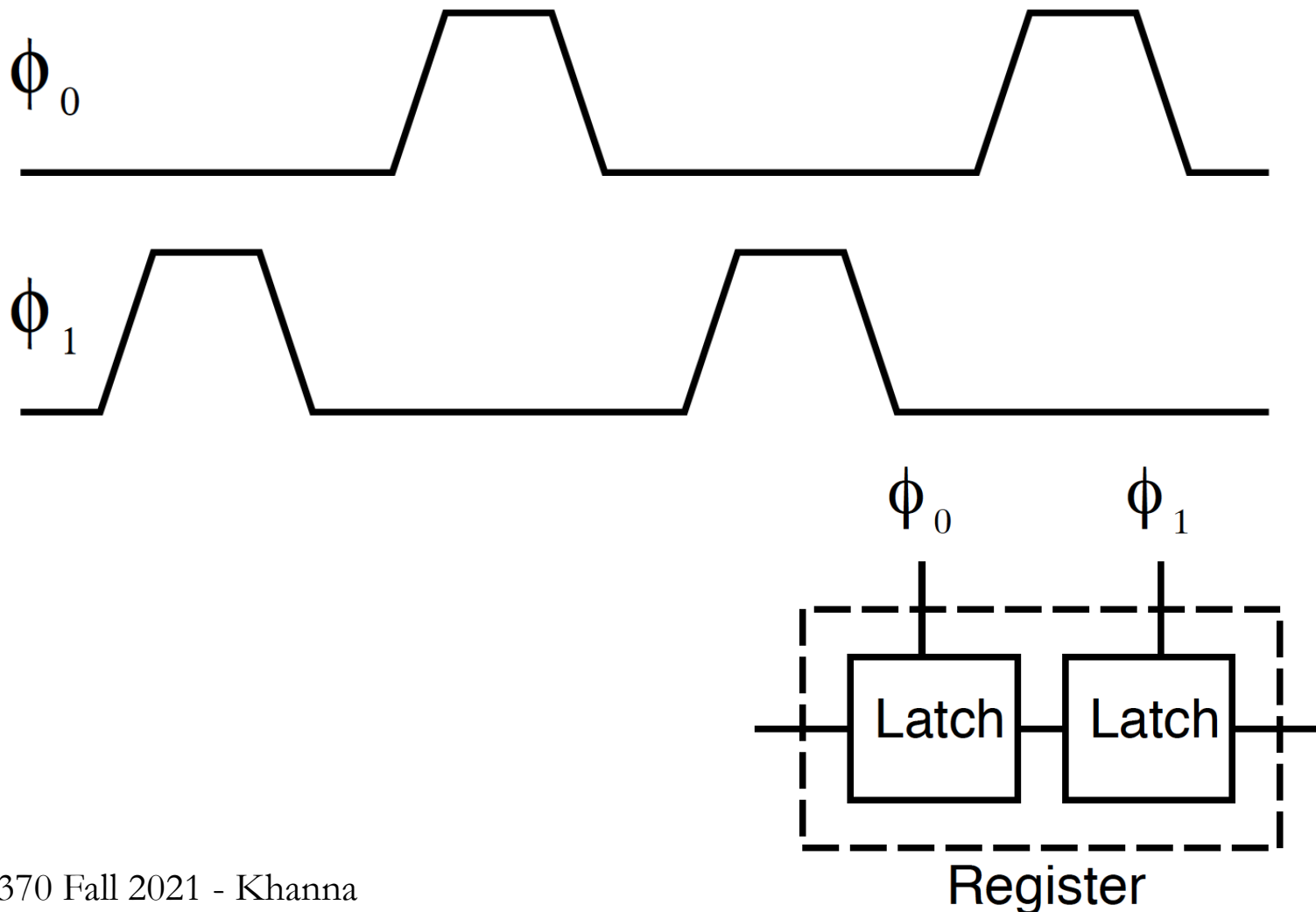
Positive-Edge Triggered Register

- ❑ Build register from pair of latches
- ❑ Control with non-overlapping clocks



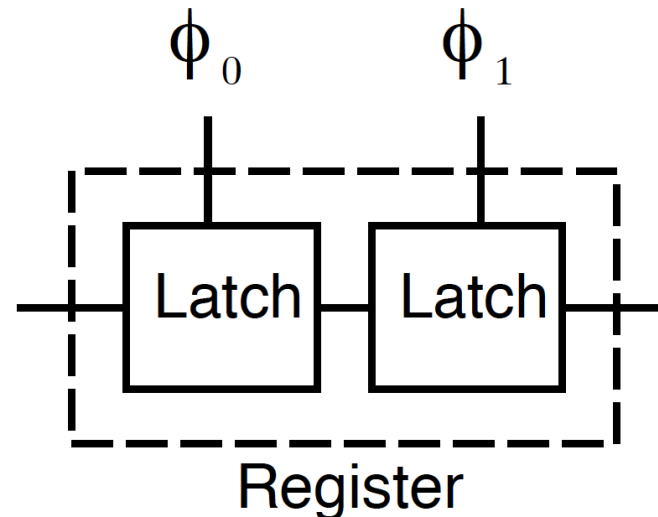
Two Phase Non-Overlapping Clocks

- What timing constraints do we have?



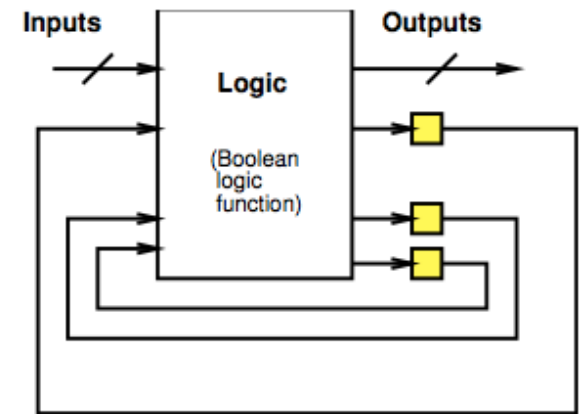
Latch Timing Issues

- What timing constraints do latches impose?
 - When can ϕ change?
 - How long must ϕ be high?
 - Delay when ϕ is high?

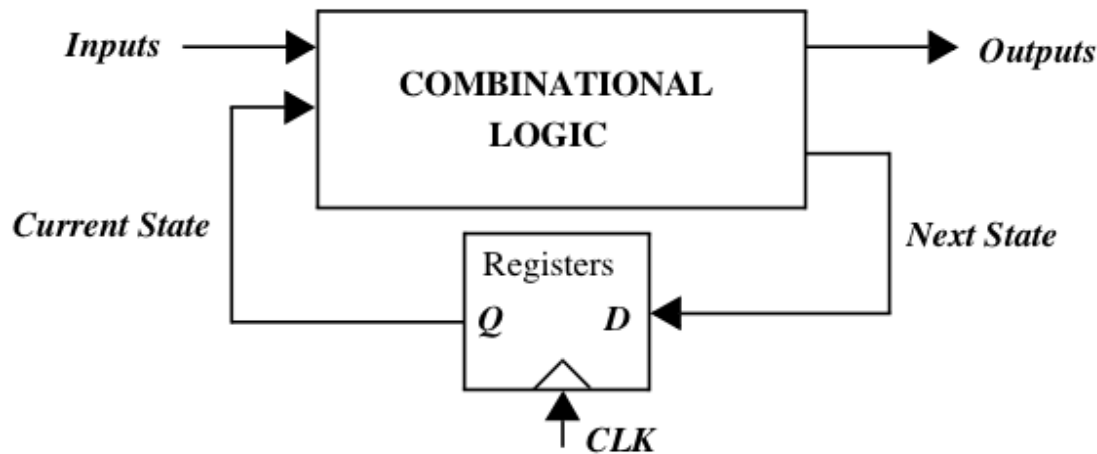


Clocking Discipline

- ❑ Follow discipline of combinational logic broken by registers
- ❑ Compute
 - From state elements
 - Through combinational logic
 - To new values for state elements
- ❑ As long as clock cycle long enough,
 - Will get correct behavior



Latch Timing Issues

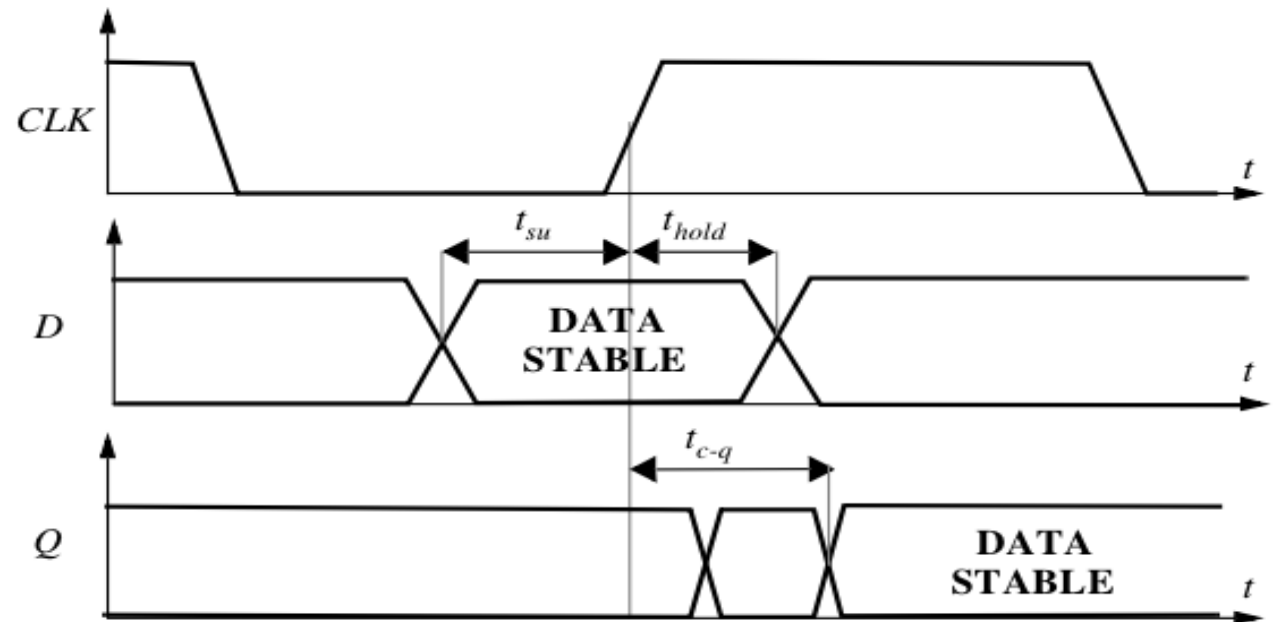


Worst case delays

$$T \geq t_{c-q} + t_{plogic} + t_{su}$$

$$t_{cdregister} + t_{cdlogic} \geq t_{hold}$$

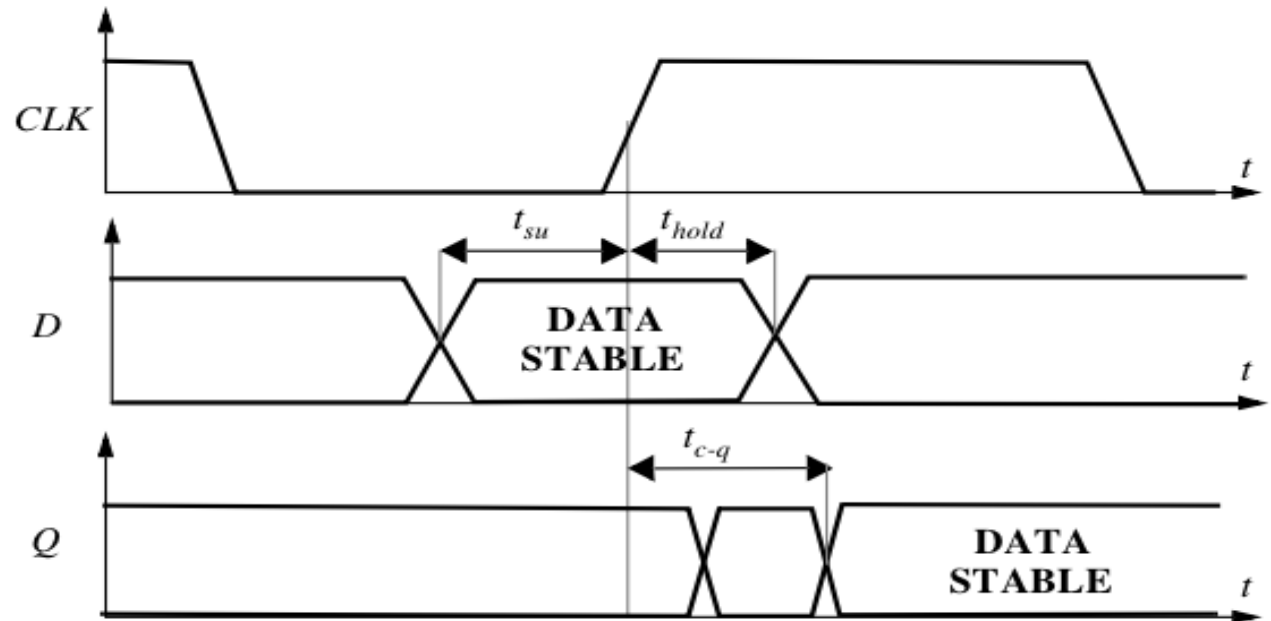
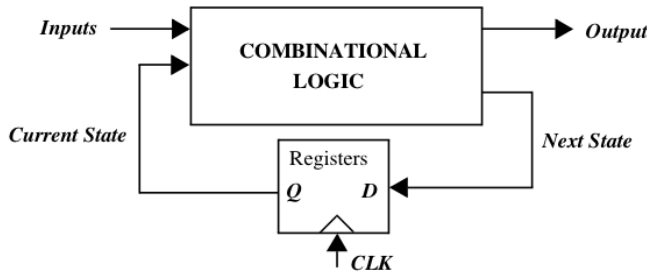
Minimum delays



Latch Timing Issues

- ❑ t_{su} = time data (D) must be valid before CLK edge
- ❑ t_{plogic} = worst case propagation delay of logic
- ❑ t_{c-p} = worst case propagation delay of latch

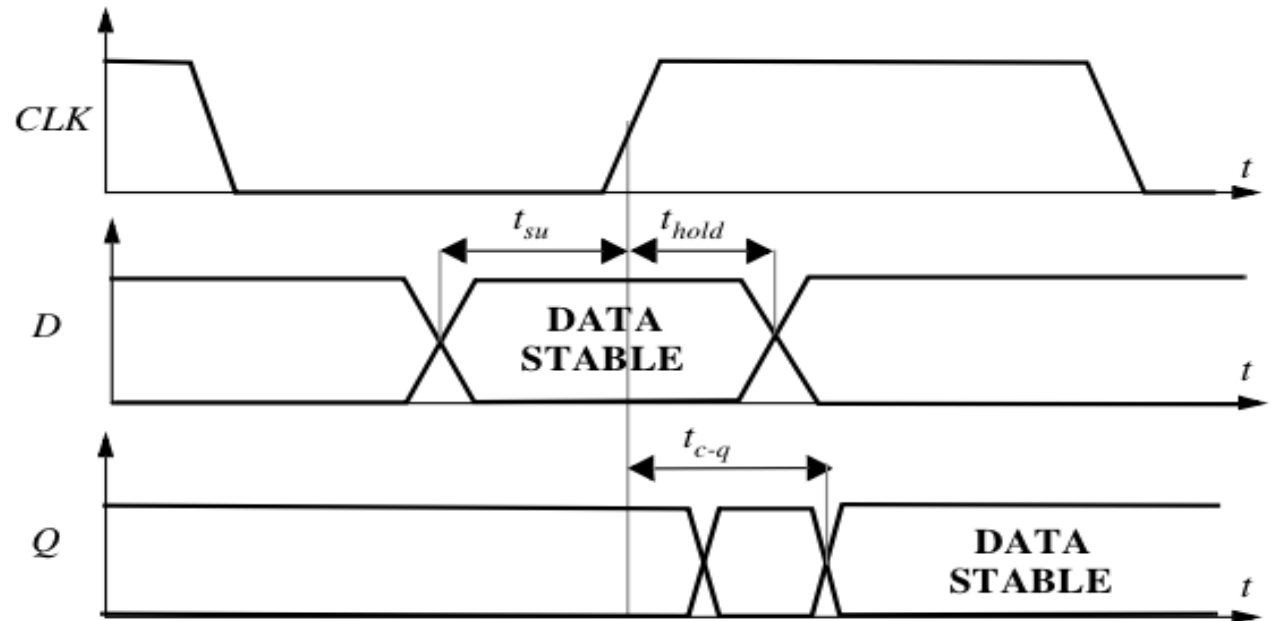
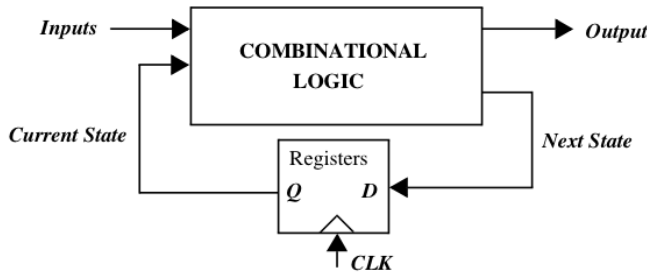
$$T \geq t_{c-q} + t_{plogic} + t_{su}$$



Latch Timing Issues

- ❑ $t_{cdregister}$ = minimum propagation delay of latch
- ❑ $t_{cdlogic}$ = minimum propagation delay of logic
- ❑ t_{hold} = time data (D) must stay valid after CLK edge

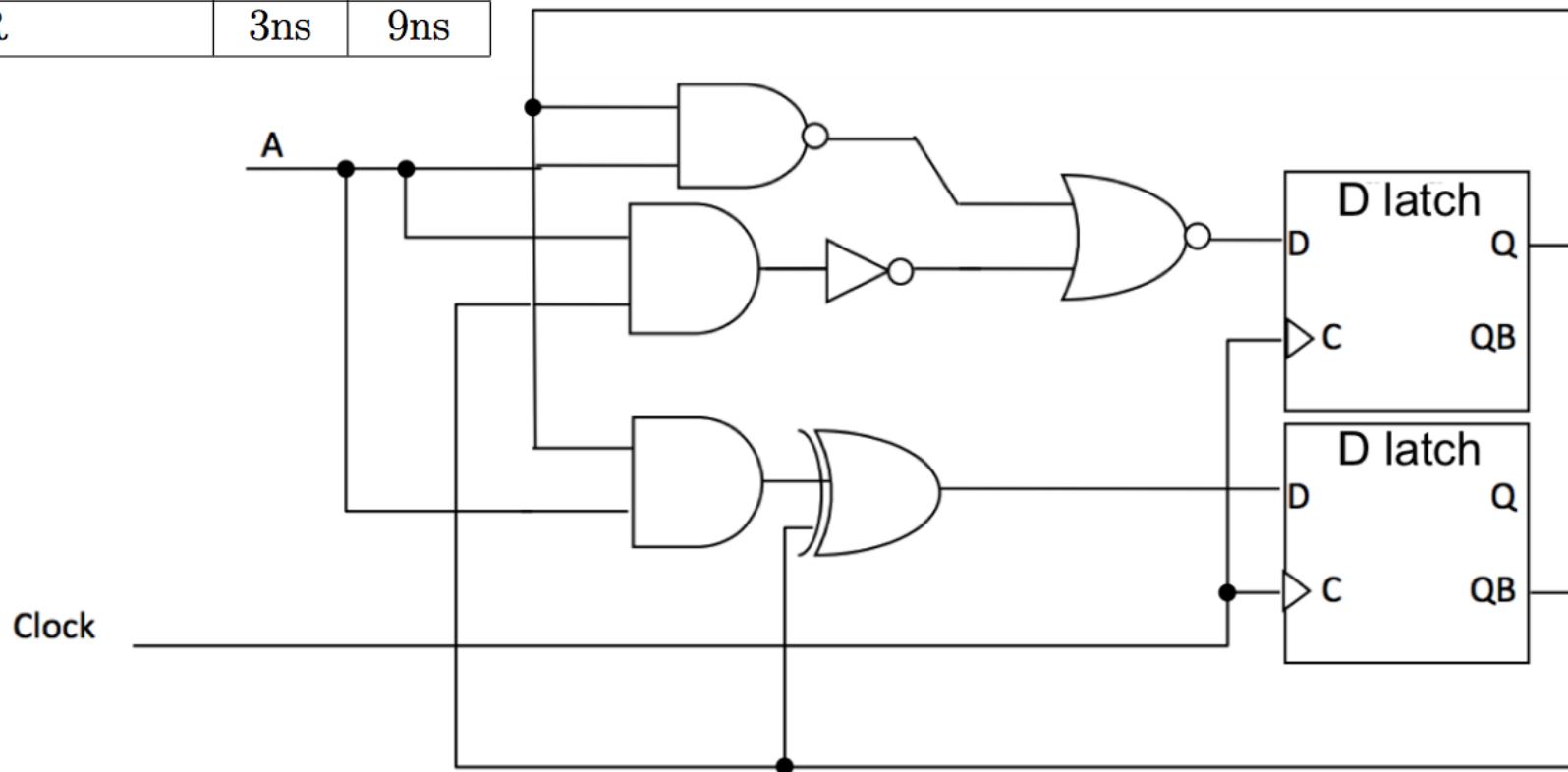
$$t_{cdregister} + t_{cdlogic} \geq t_{hold}$$



Timing Example (Preclass5)

	Min	Max
OR/AND	3ns	5ns
NOR/NAND	2ns	4ns
NOT	1ns	2ns
XOR	3ns	9ns

Latch:		Min	Max
	<i>Clock to Q</i>	2ns	3ns
	<i>Setup time</i>	7ns	
	<i>Hold time</i>	6ns	

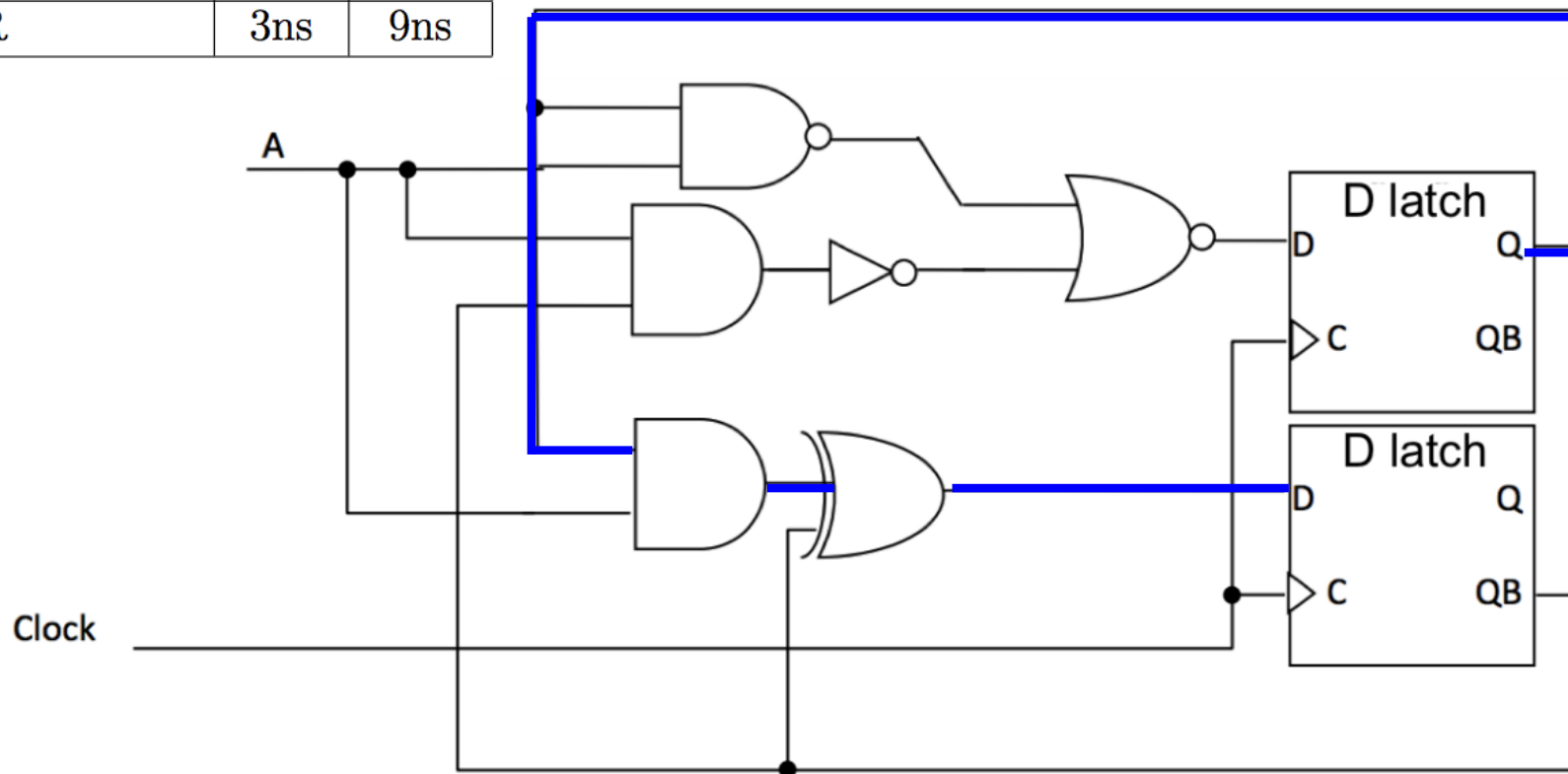


$$T \geq t_{c-q} + t_{plogic} + t_{su}$$

Timing Example (Preclass5)

	Min	Max
OR/AND	3ns	5ns
NOR/NAND	2ns	4ns
NOT	1ns	2ns
XOR	3ns	9ns

Latch:		Min	Max
	<i>Clock to Q</i>	2ns	3ns
	<i>Setup time</i>	7ns	
	<i>Hold time</i>	6ns	

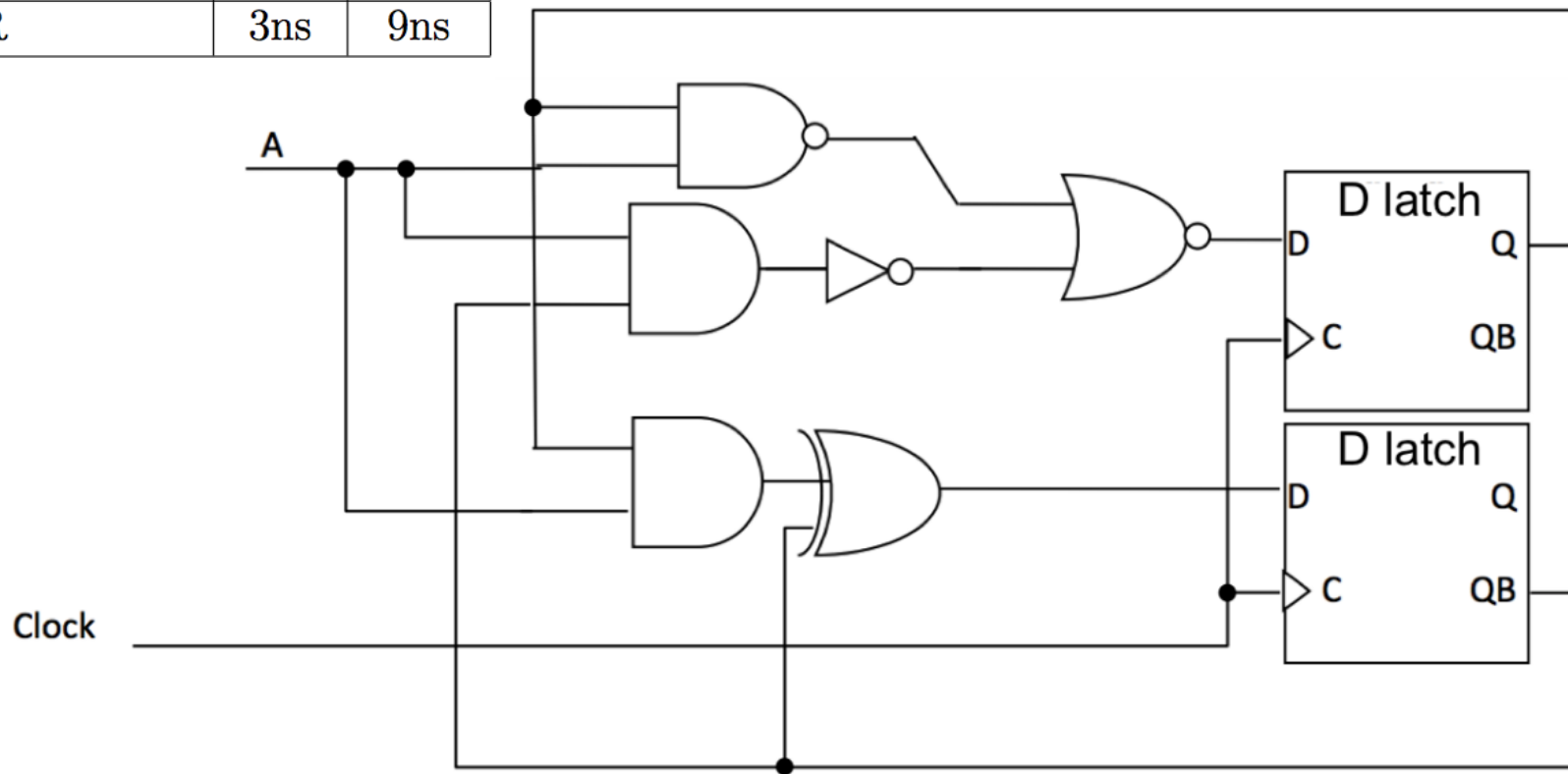


$$T \geq t_{c-q} + t_{plogic} + t_{su}$$

Timing Example (Preclass5)

	Min	Max
OR/AND	3ns	5ns
NOR/NAND	2ns	4ns
NOT	1ns	2ns
XOR	3ns	9ns

Latch:		Min	Max
	<i>Clock to Q</i>	2ns	3ns
	<i>Setup time</i>	7ns	
	<i>Hold time</i>	6ns	

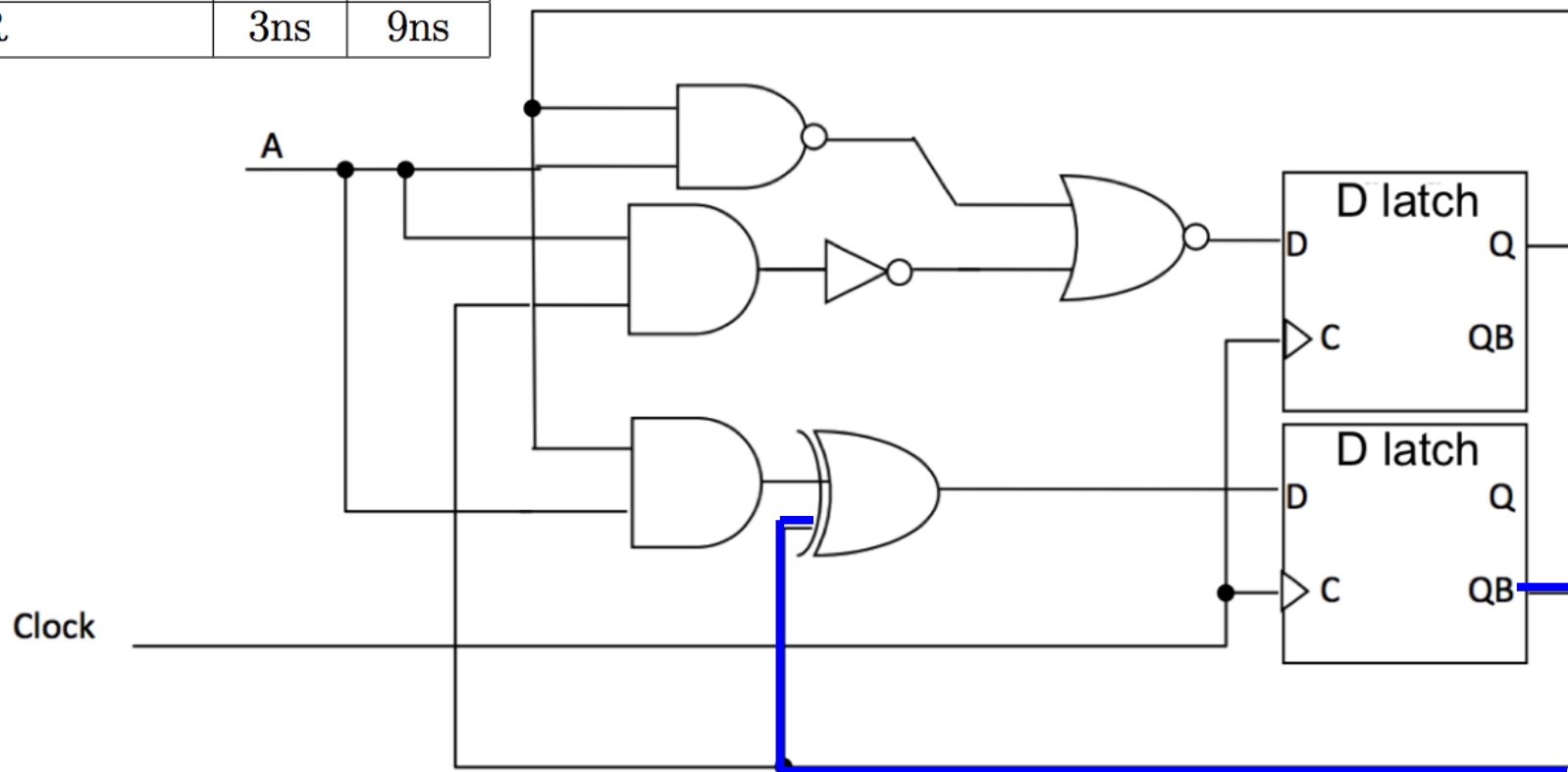


$$t_{cdregister} + t_{cdlogic} \geq t_{hold}$$

Timing Example (Preclass5)

	Min	Max
OR/AND	3ns	5ns
NOR/NAND	2ns	4ns
NOT	1ns	2ns
XOR	3ns	9ns

Latch:		Min	Max
	<i>Clock to Q</i>	2ns	3ns
	<i>Setup time</i>	7ns	
	<i>Hold time</i>	6ns	

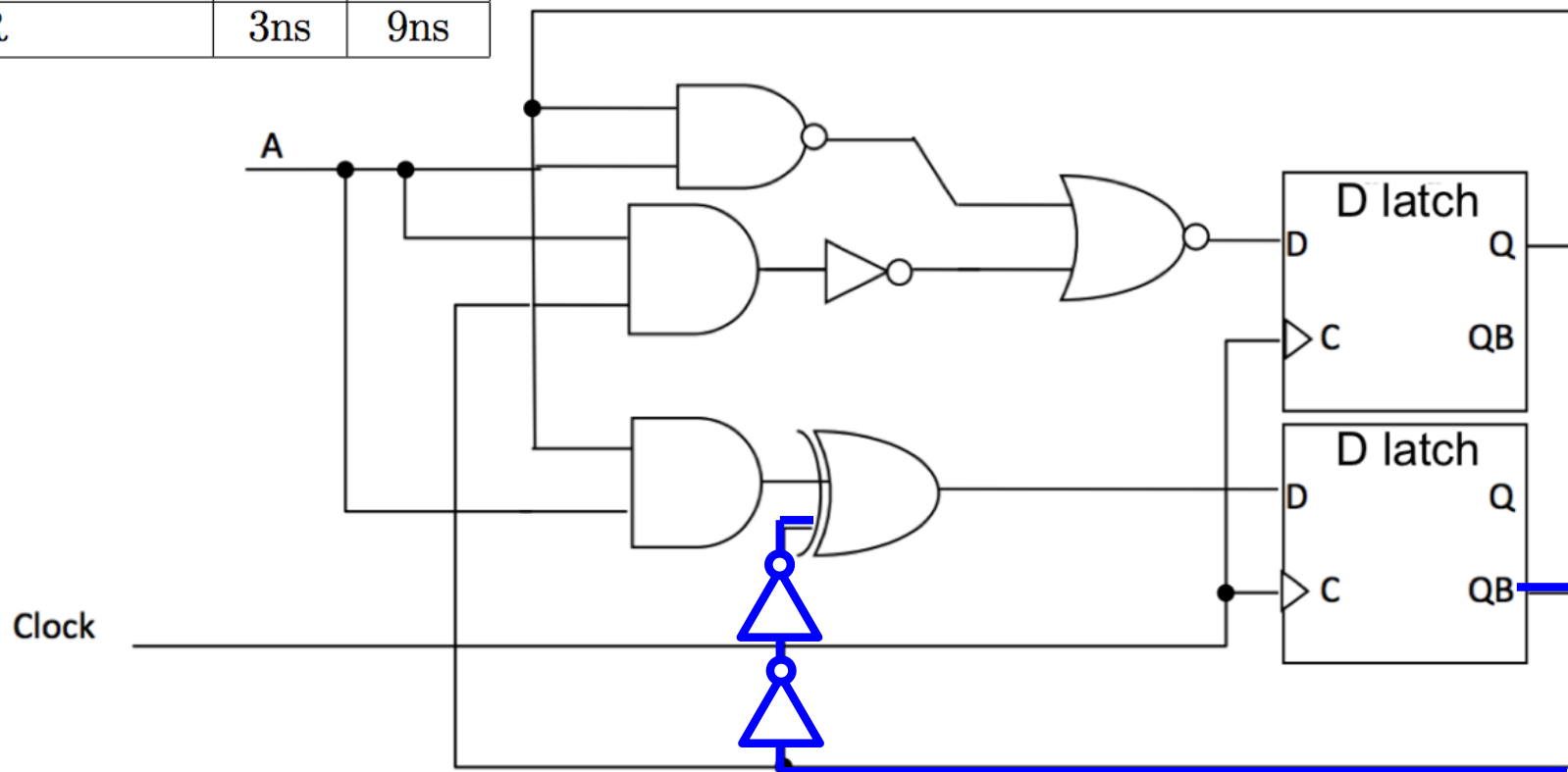


$$t_{cdregister} + t_{cdlogic} \geq t_{hold}$$

Timing Example (Preclass5)

	Min	Max
OR/AND	3ns	5ns
NOR/NAND	2ns	4ns
NOT	1ns	2ns
XOR	3ns	9ns

Latch:		Min	Max
	<i>Clock to Q</i>	2ns	3ns
	<i>Setup time</i>	7ns	
	<i>Hold time</i>	6ns	



$$t_{cdregister} + t_{cdlogic} \geq t_{hold}$$



Ideas

- ❑ Synchronize circuits
 - to external events
 - disciplined reuse of circuitry
- ❑ Leads to clocked circuit discipline
 - Uses state holding element
 - Prevents
 - Timing assumptions
 - (More) complex reasoning about all possible timings



Admin

- HW6 posted
 - Can do all of it now
 - Due **Friday** 11/12