ESE3700: Circuit-Level Modeling, Design and Optimization for Digital Systems

Lec 16: Apr 1, 2024

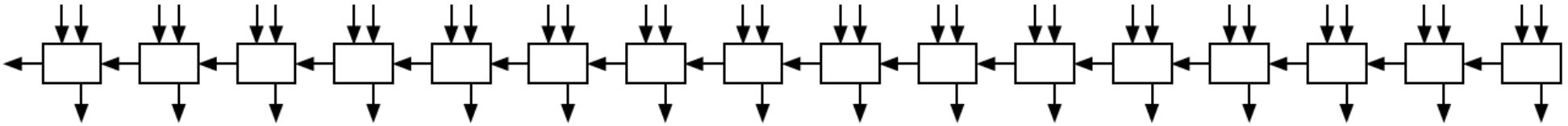
Synchronous Timing Discipline and Clocking

Today

- Managing Timing
- Latches
- Registers, clocking

Timing Setup (Preclass1&2)

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

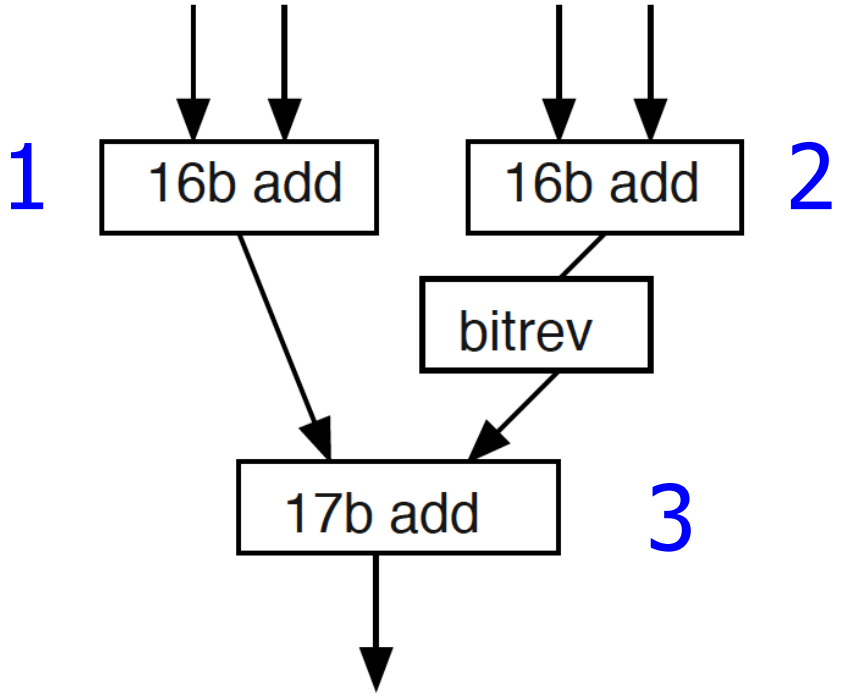


- Worst-case delay to output?
- Shortest delay to output?
  - assuming the output switches

Timing Setup (Preclass 3)

- New set of inputs every $20T_{bit}$
  - How does it behave?

- bitrev: Out[i]=In[17-i]
  - Assume delay of bitrev<<T_{bit}



Challenge

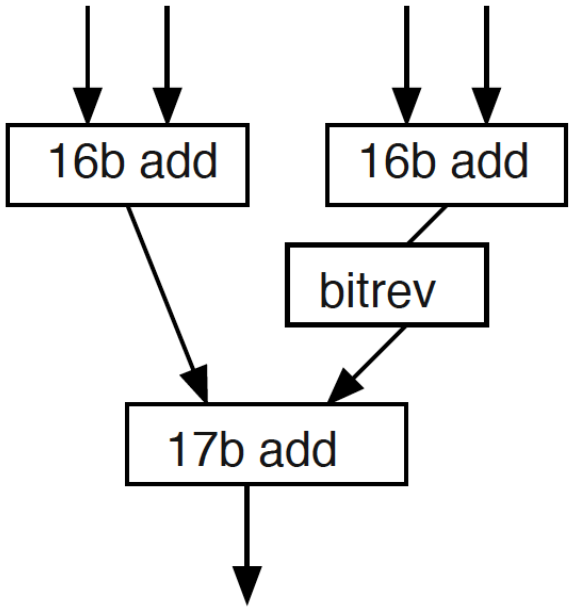
- Logic paths have different delays
  - E.g. different output bits in an adder
- Delay of signal is data dependent
  - E.g. length of carry propagation
- Delay is chip dependent
  - E.g. Threshold Variation
- Delay is environment dependent
  - E.g. Temperature

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

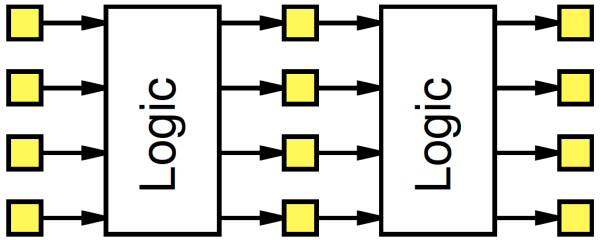
Logic Timing

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



Discipline

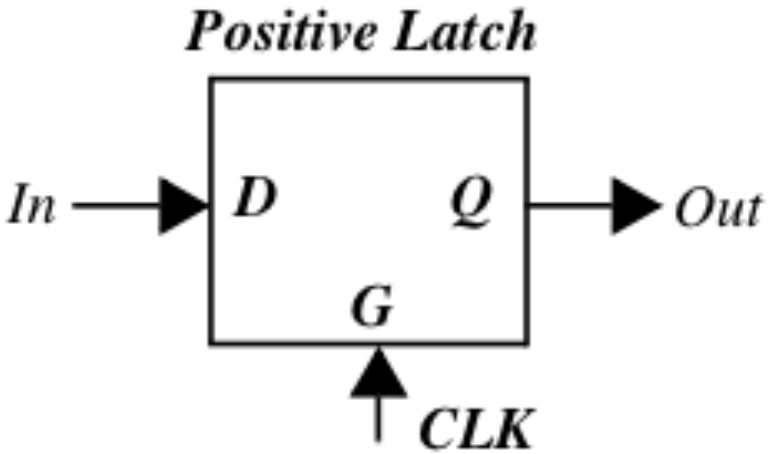
- Add circuit elements to
  - hold values
  - and change at coordinated point
    - Control when changes seen by circuit



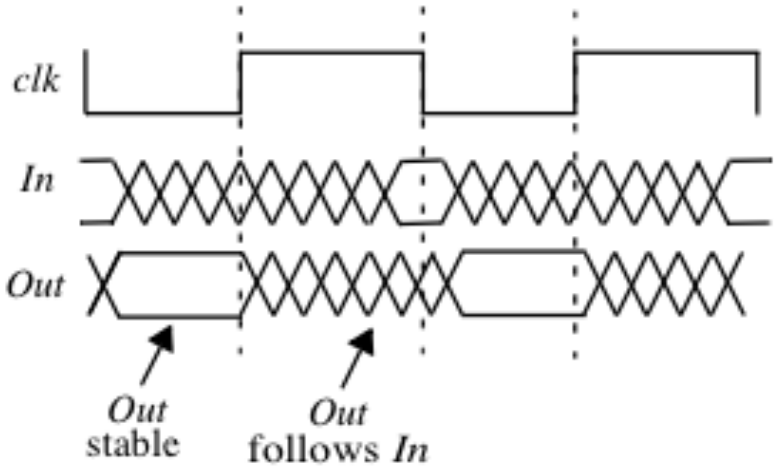
- Only have to make sure to
  wait long enough for all results
- Decouple outputs and inputs
  - timing of signal change (settling of output change)…
  - …from timing of signal usage (output used as input)

Latch

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

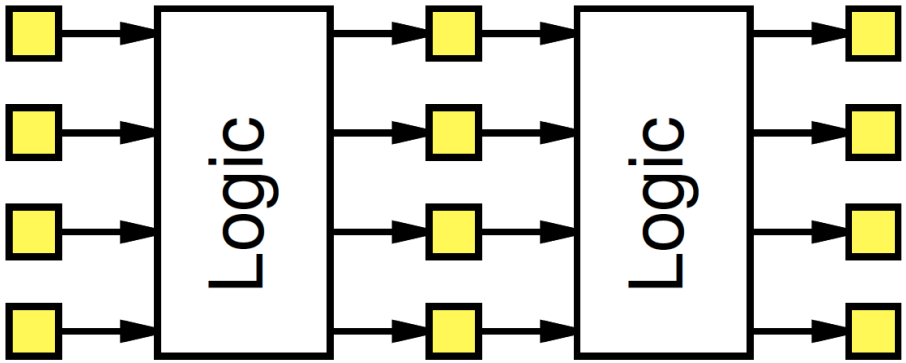


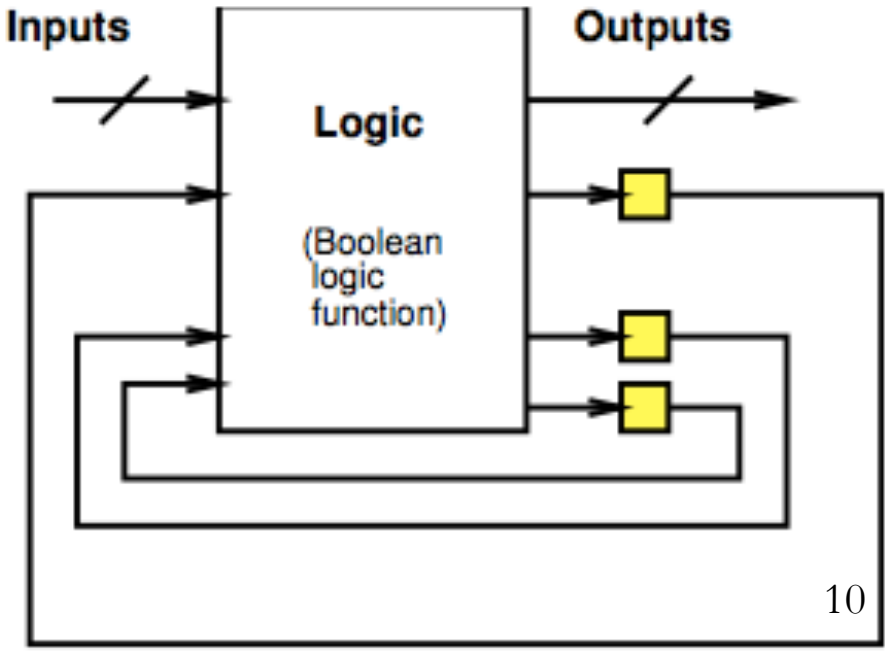
$$Q = \overline{CLK} \cdot Q + CLK \cdot In$$



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

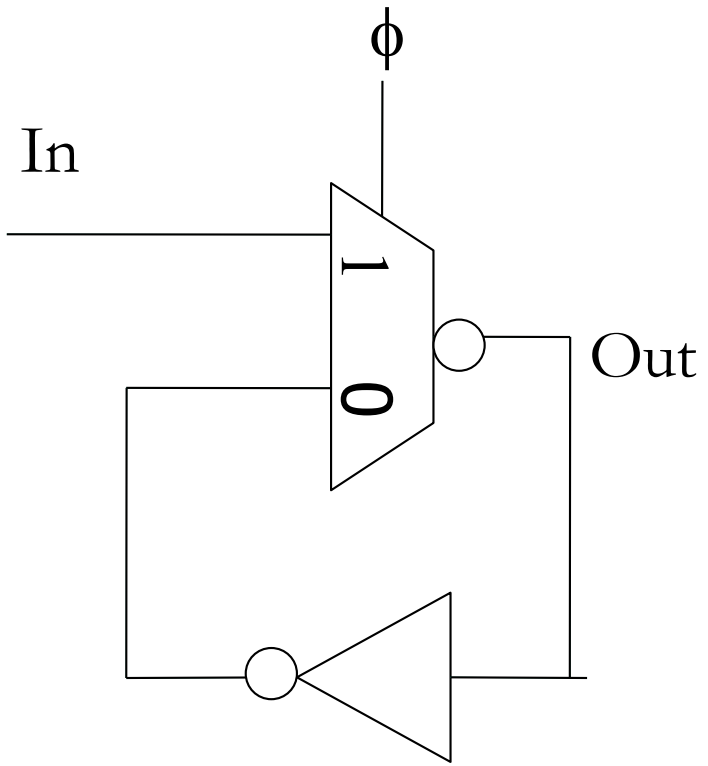
Latch (Preclass 4a)

- ϕ=1 ➔ Out=/In
- ϕ=0 ➔ Out=Out

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

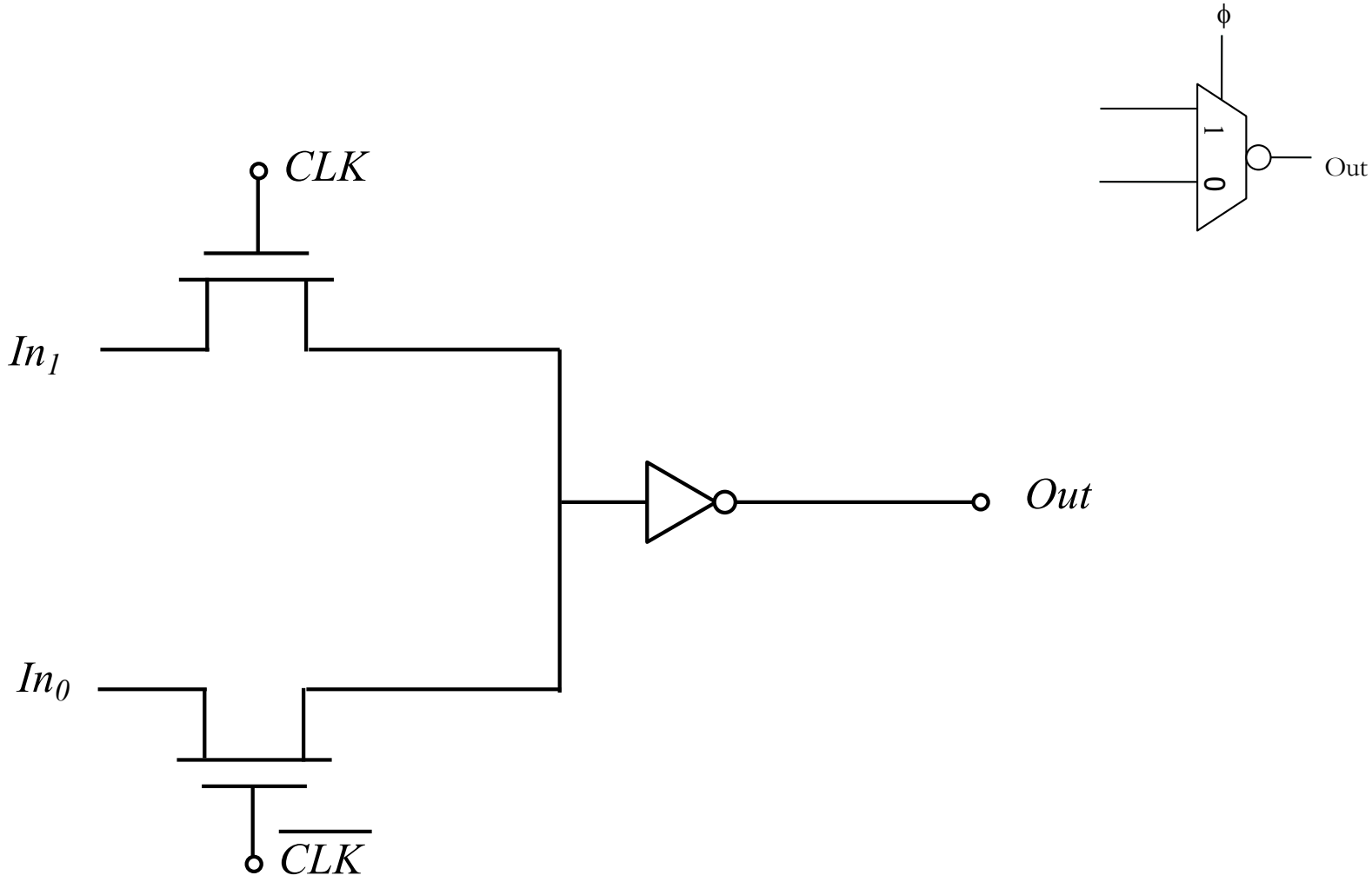
Muliplexer-based Latch from CMOS Logic

- ϕ=1 ➔ Out=/In
- ϕ=0 ➔ Out=Out

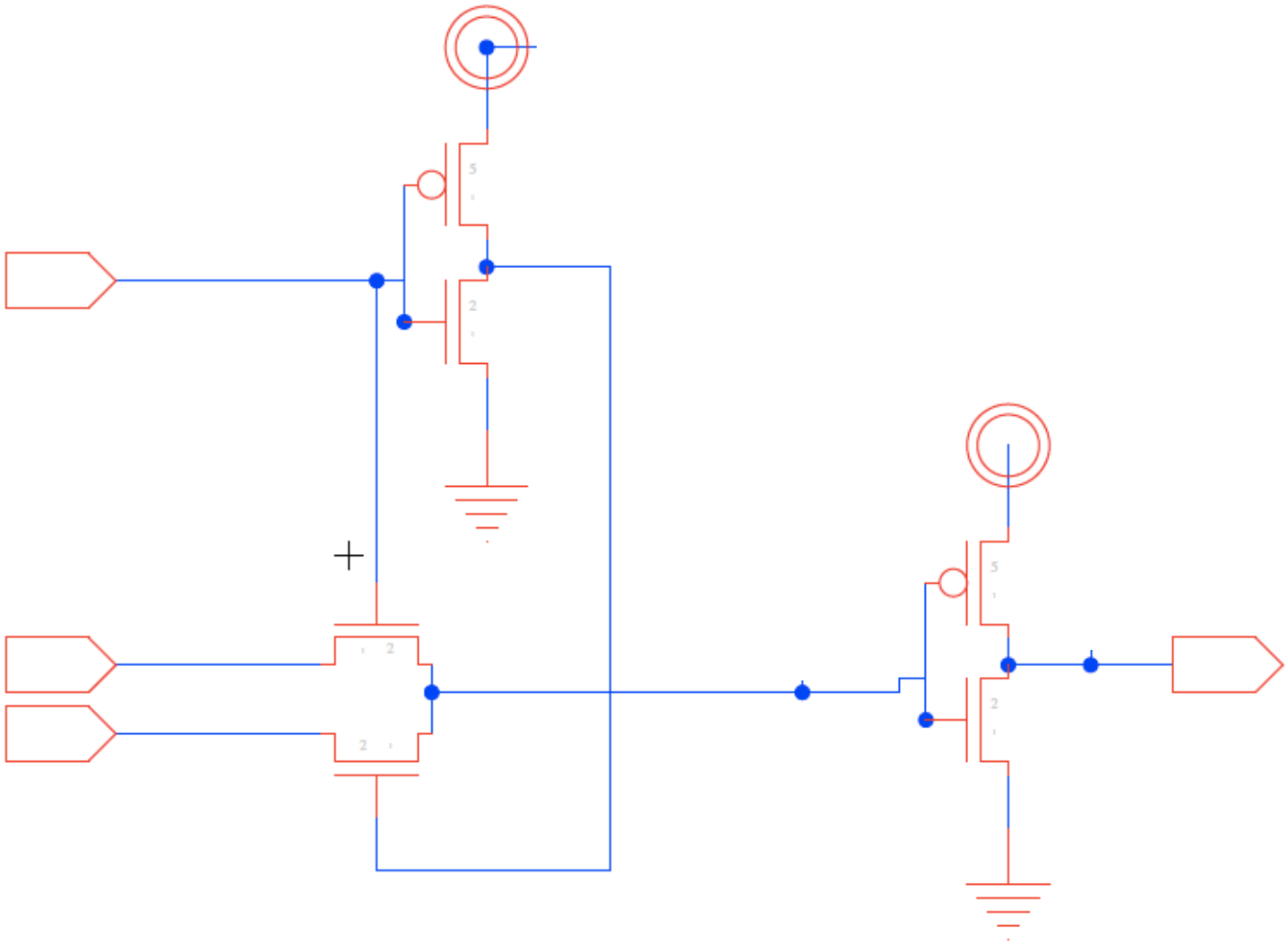


How to build the mux with pass logic?

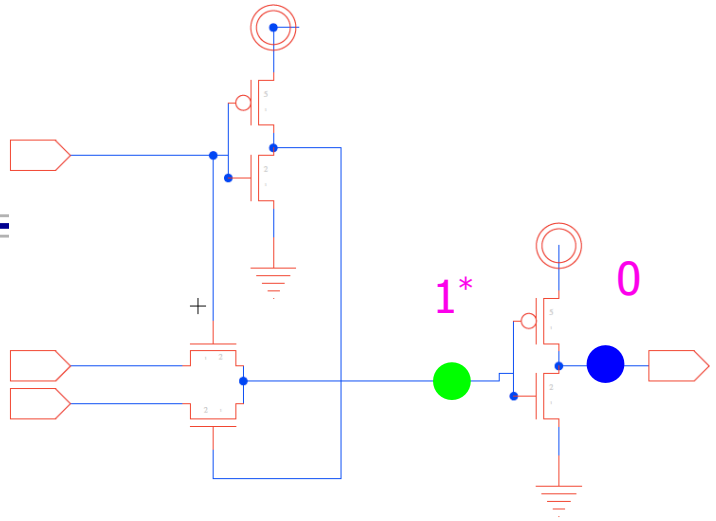
Mux at Transistor Level

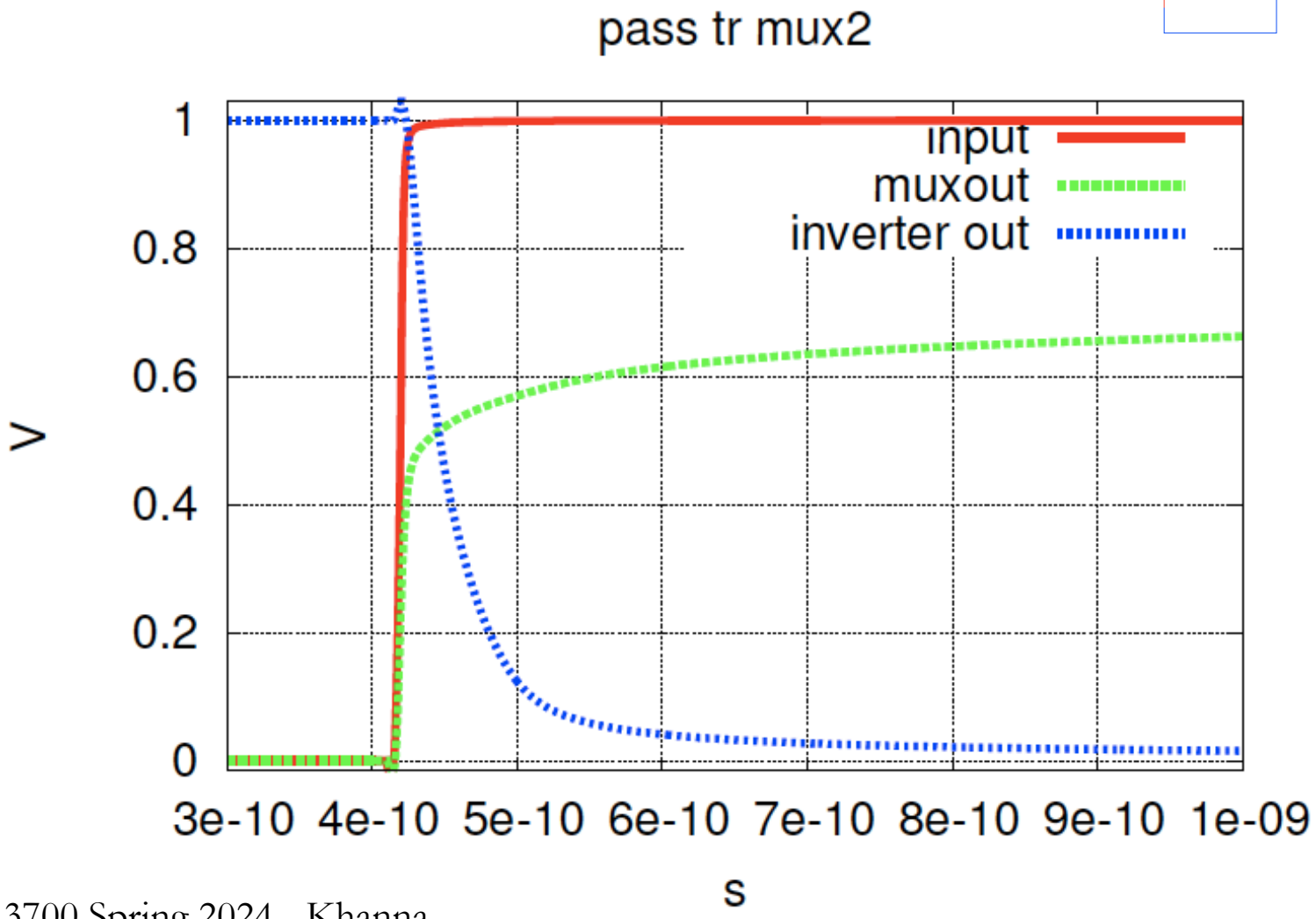


MuxL

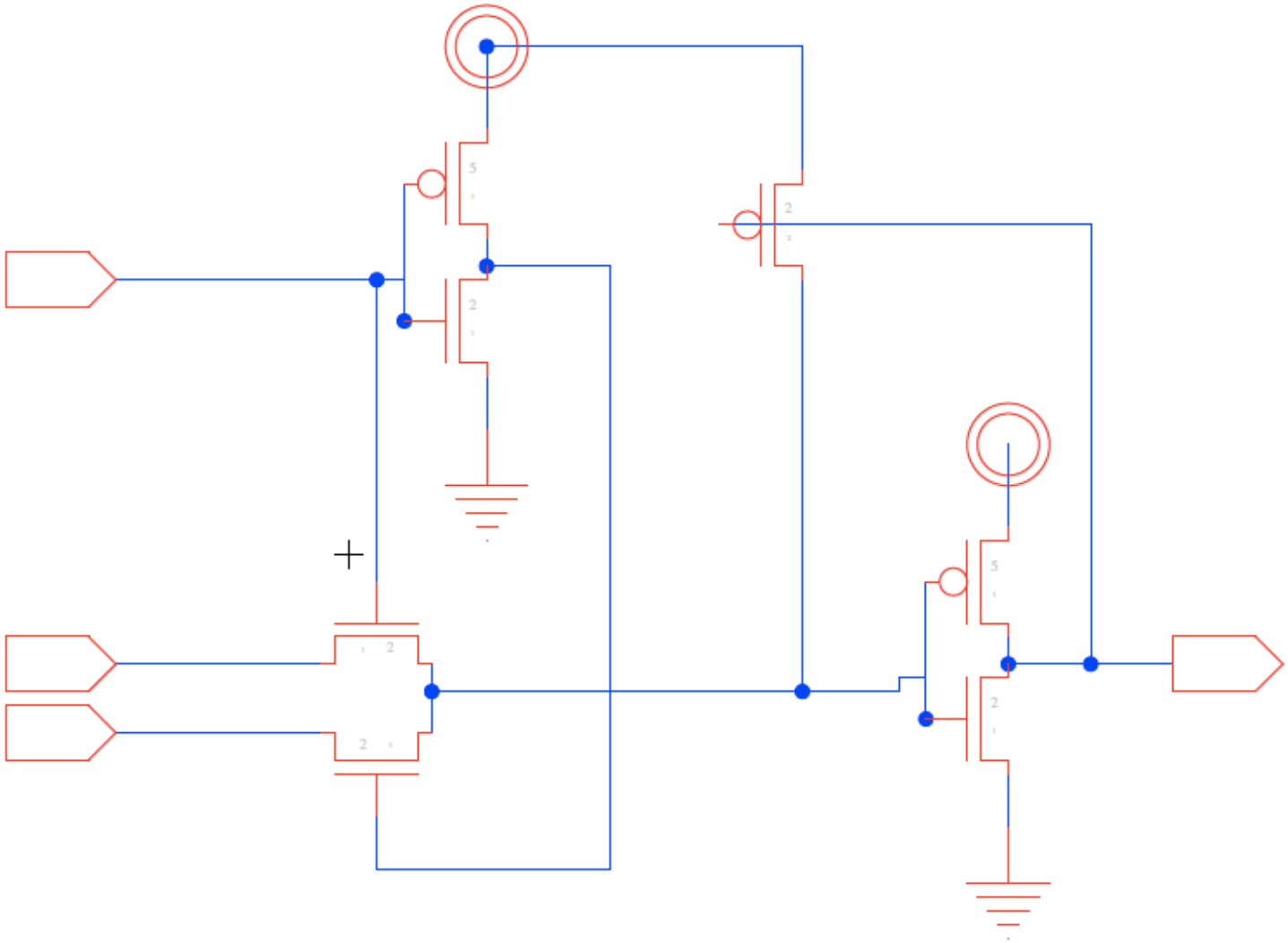


Without level restorer

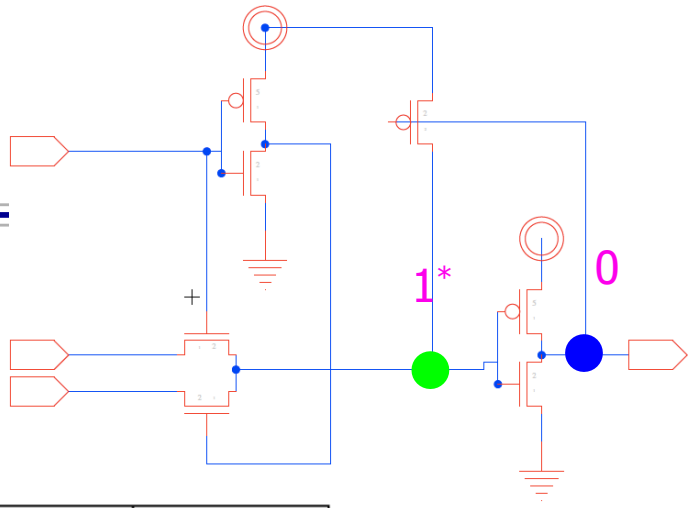


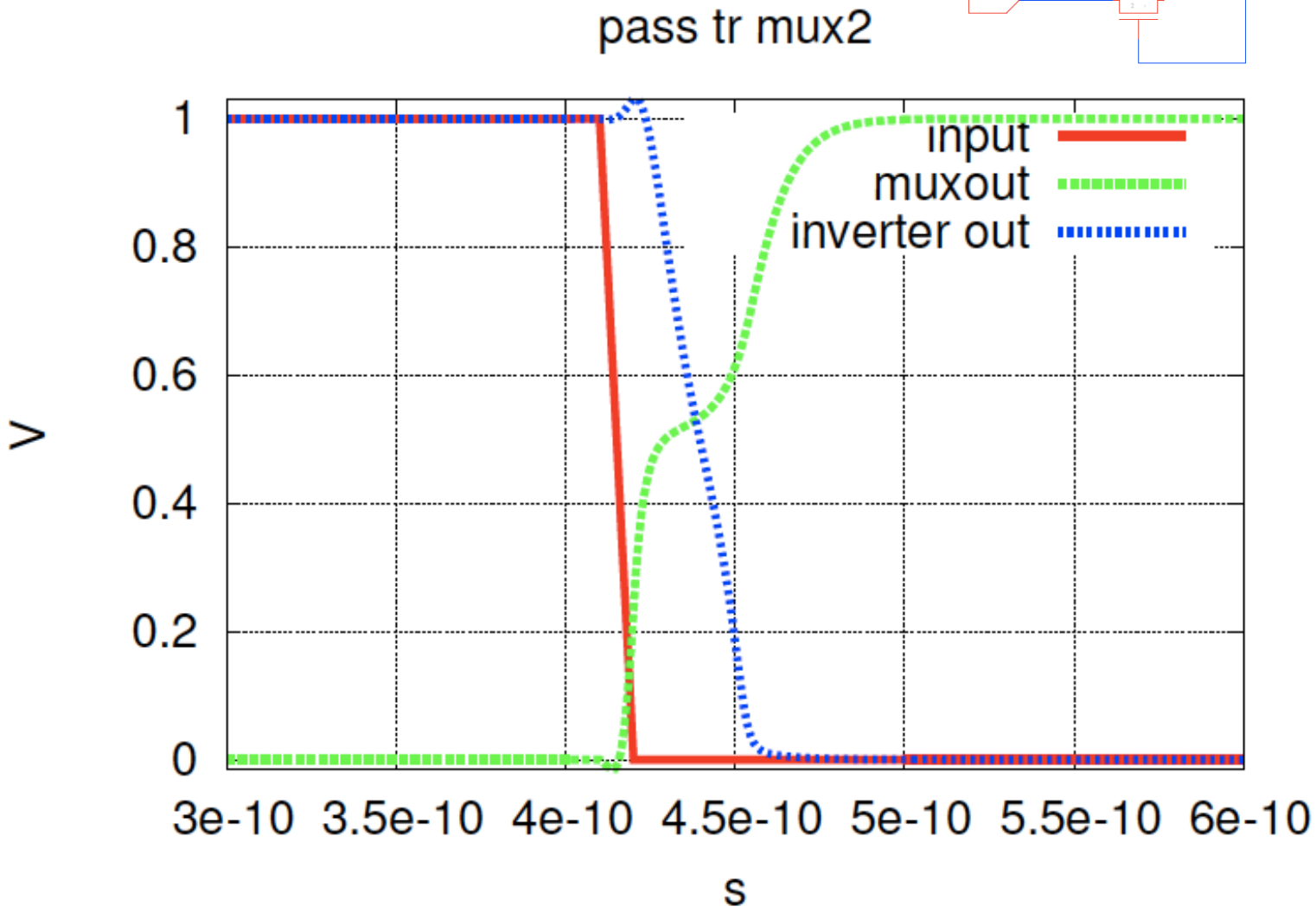


MuxL Level Restorer ("Staticizer")



With level restorer



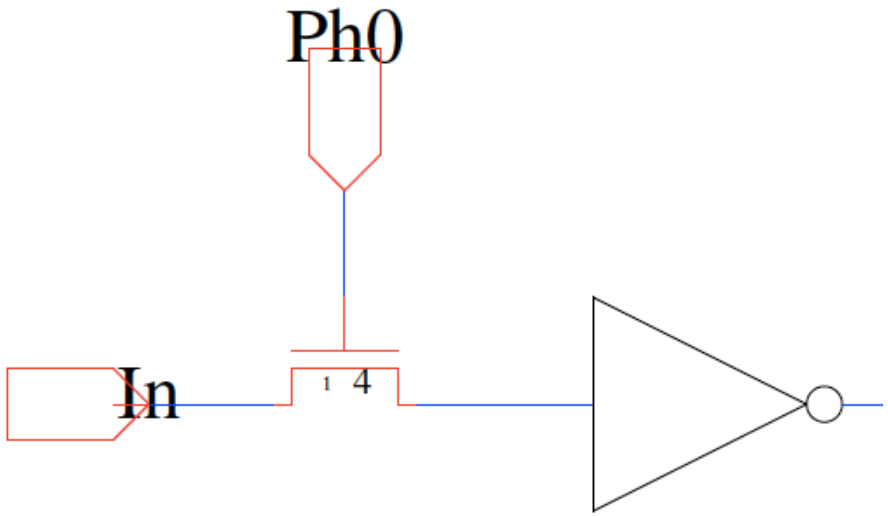


Latch (Preclass 4b)

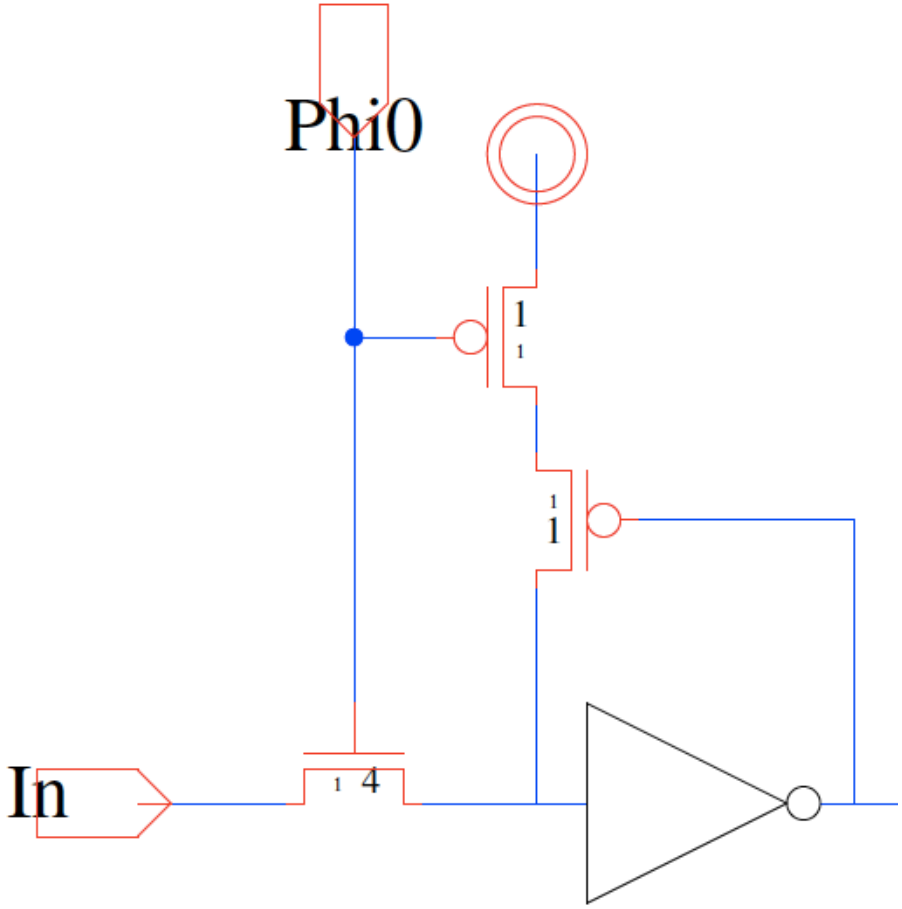
- ϕ=1 ➔ Out=/In
- ϕ=0 ➔ Out=Out
- ϕ transitions 1➔0 Out holds value

- How do we build a latch from pass transistors?

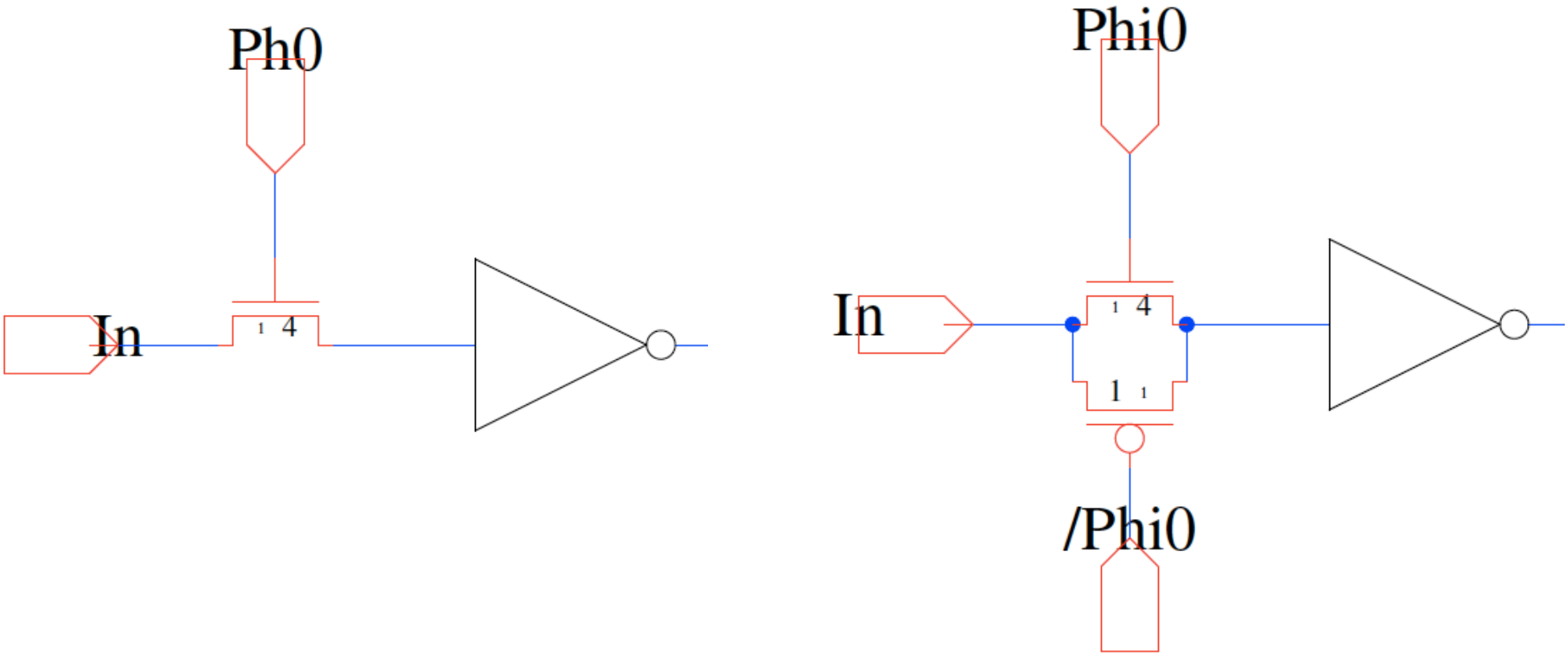
Simple Latch



Latch with Level Restore



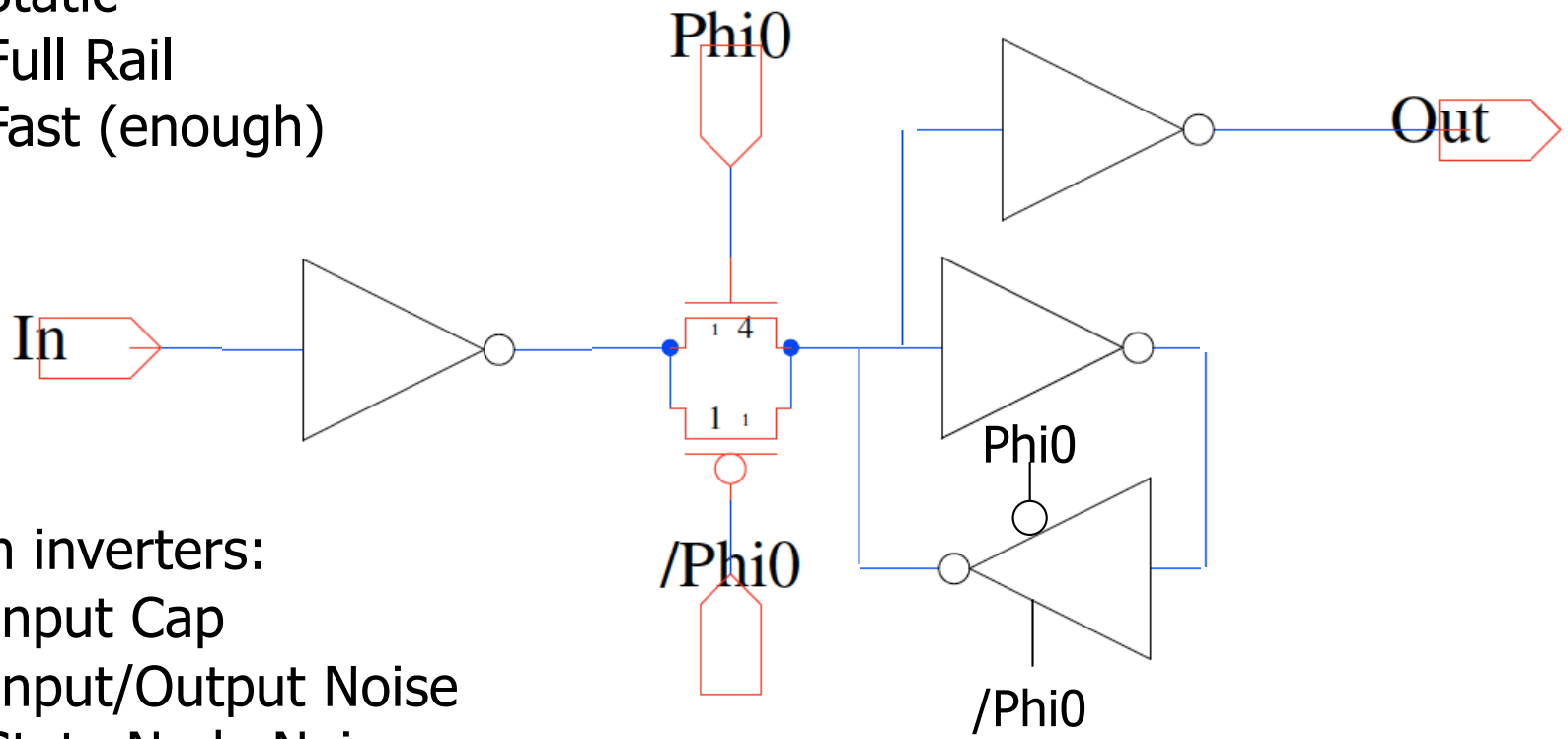
What is the difference?



Typical Static Latch

Advantages: VERY ROBUST

- Static
- Full Rail
- Fast (enough)



Isolation inverters:

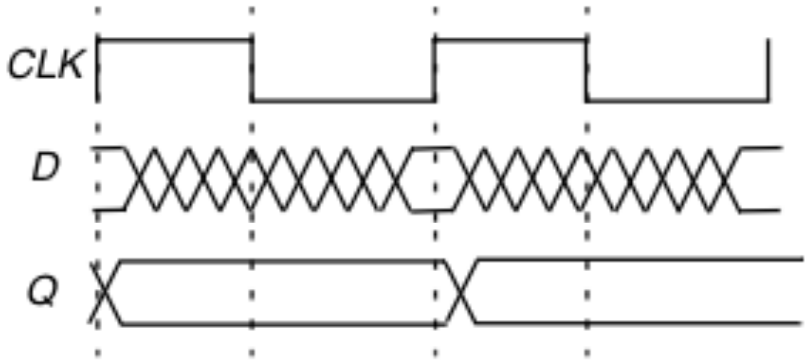
- Input Cap
- Input/Output Noise
- State Node Noise

Disadvantages:

- Large
- High clock loading

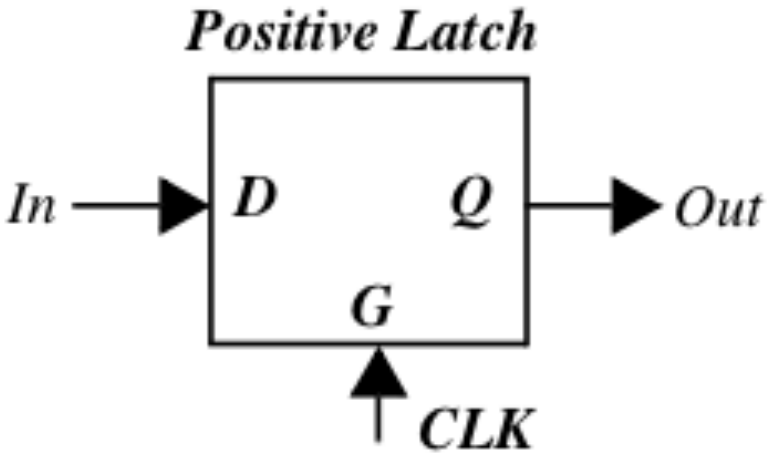
Register

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

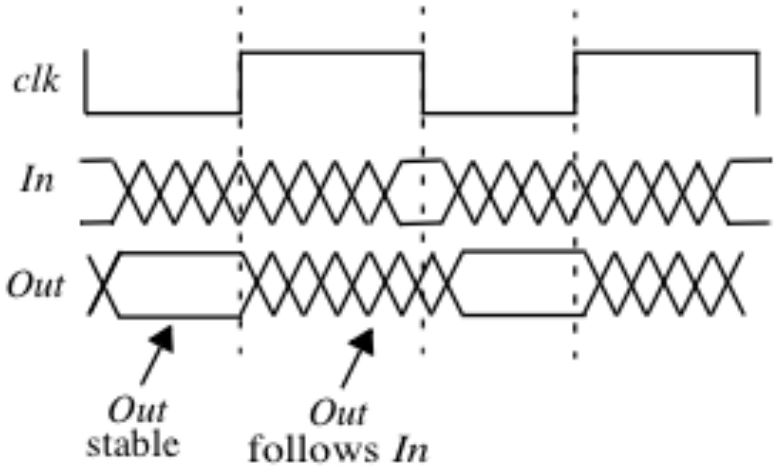


Latch

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



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

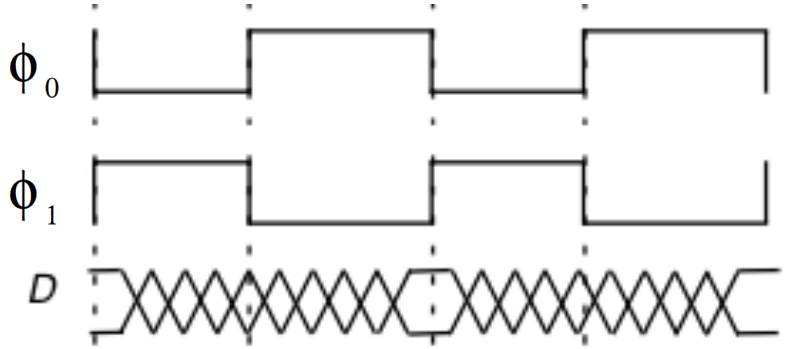


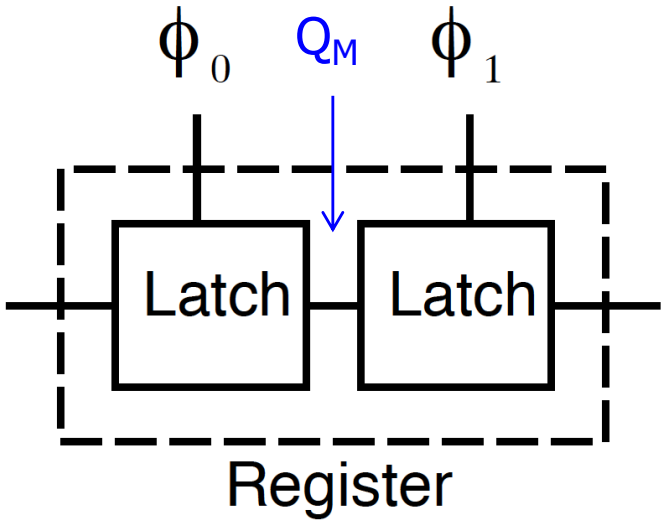
Shift Register

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

Positive-Edge Triggered Register

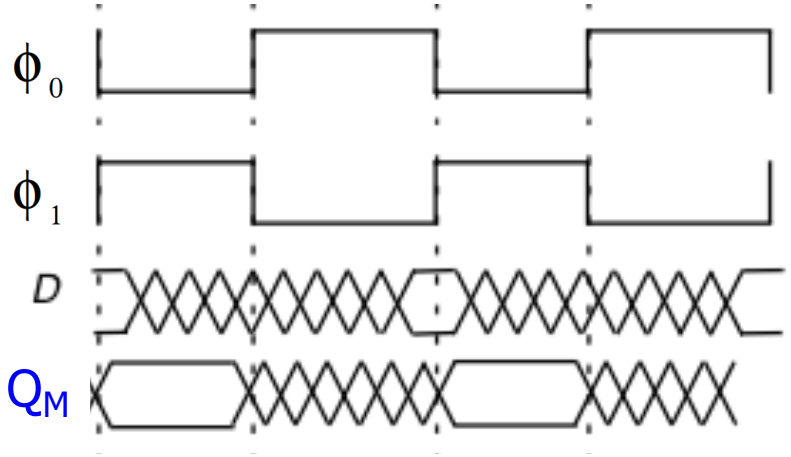
- Build register from pair of latches
- What happens when ϕ_0 is high?
- What happens when ϕ_1 is high?

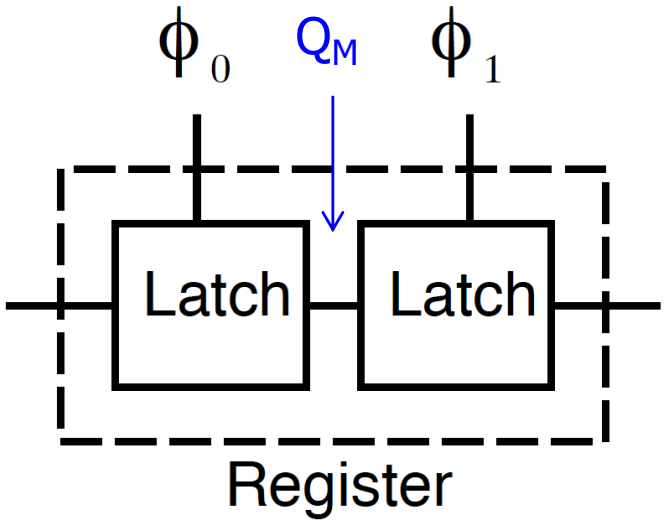




Positive-Edge Triggered Register

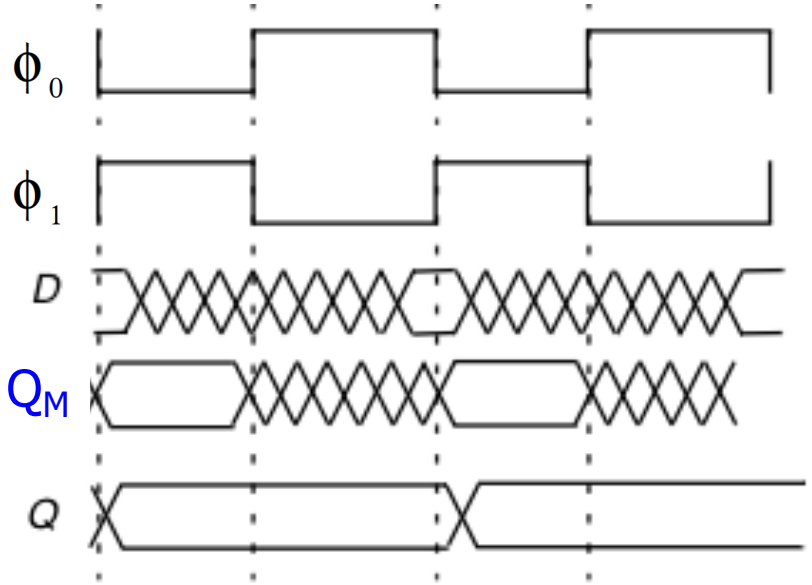
- Build register from pair of latches

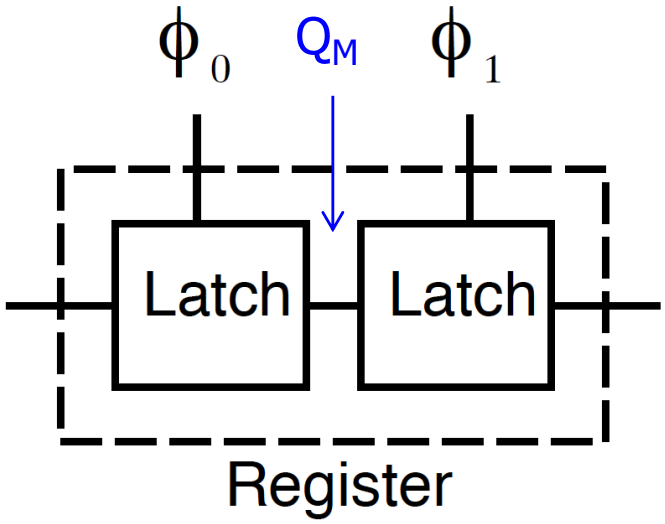




Positive-Edge Triggered Register

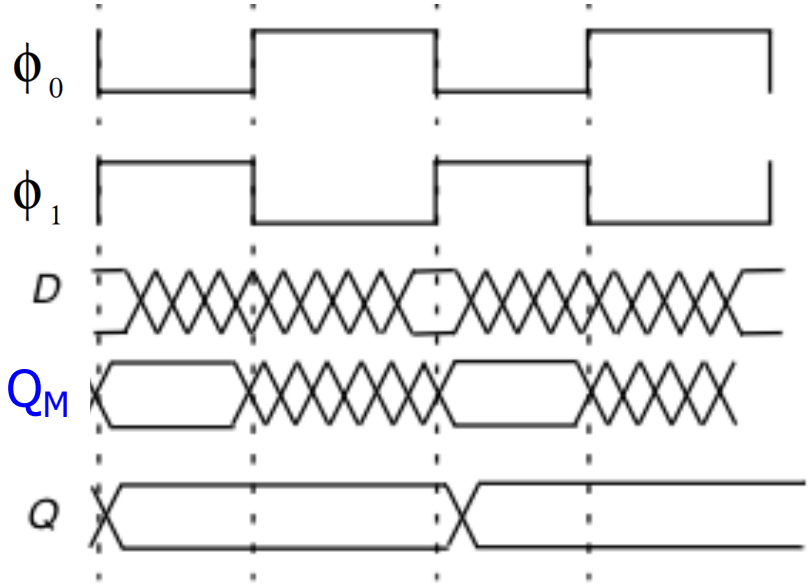
- Build register from pair of latches
- What could go wrong if clocks overlap?

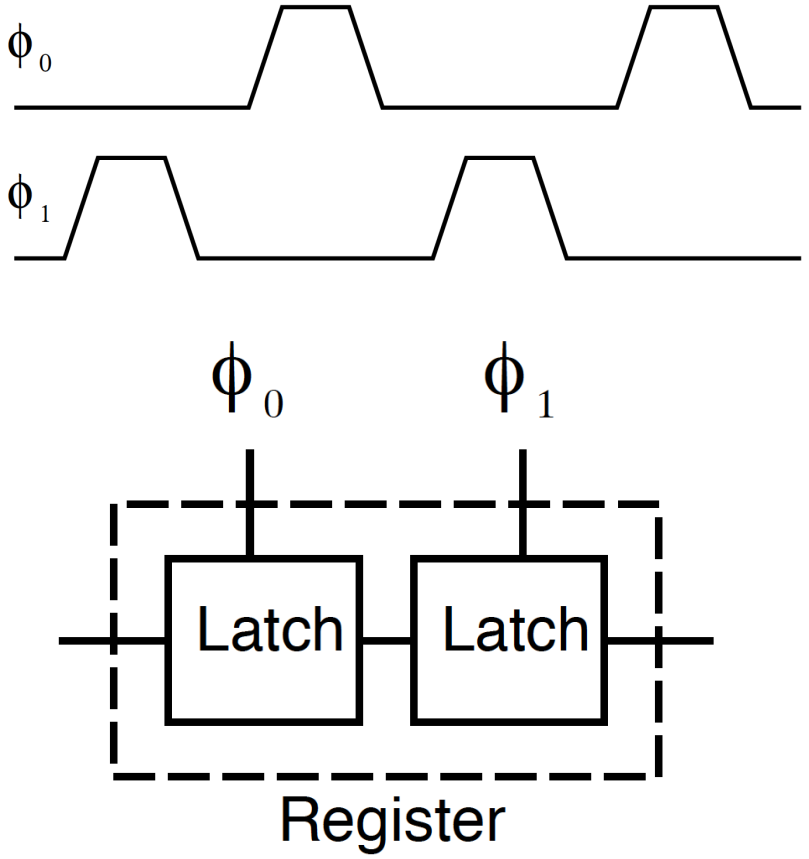




Positive-Edge Triggered Register

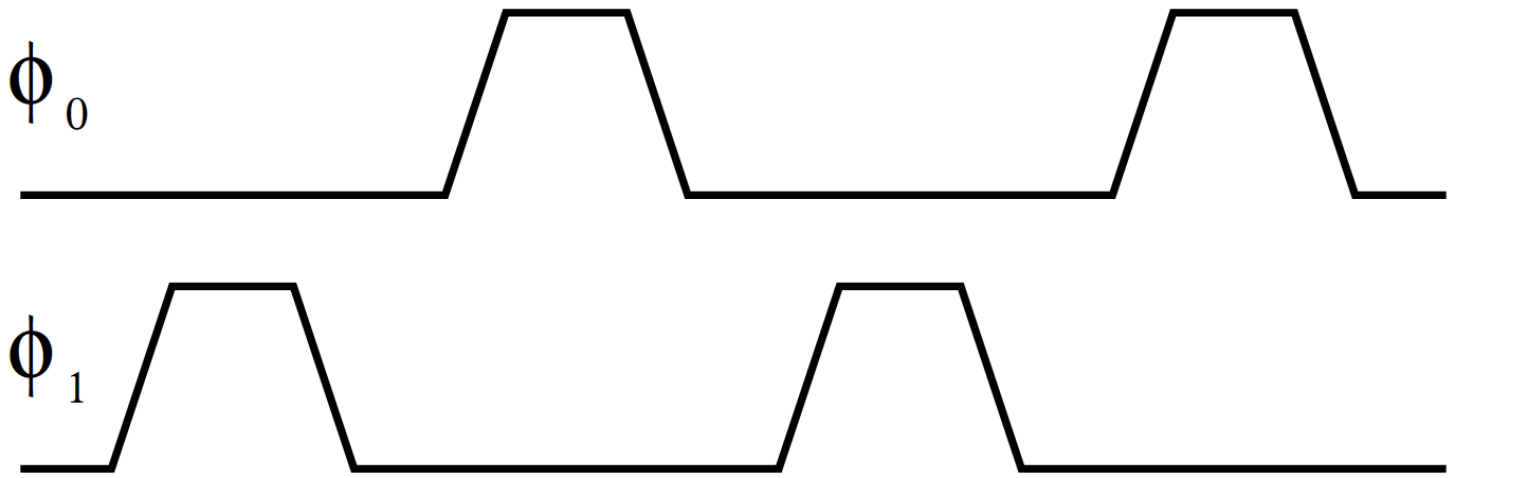
- Build register from pair of latches
- Control with non-overlapping clocks

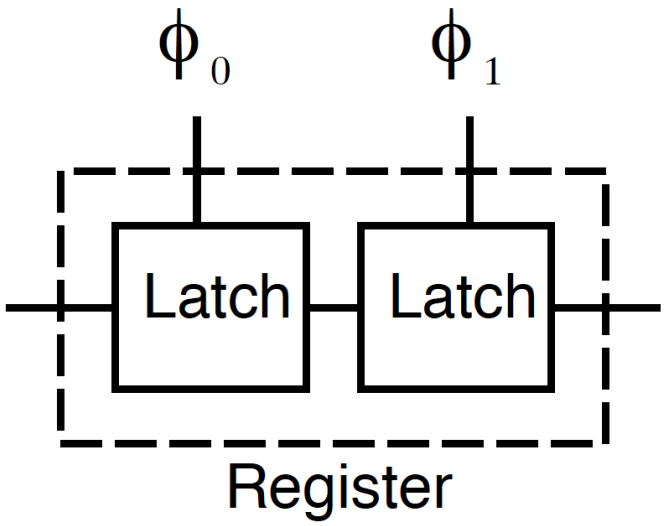




Two Phase Non-Overlapping Clocks

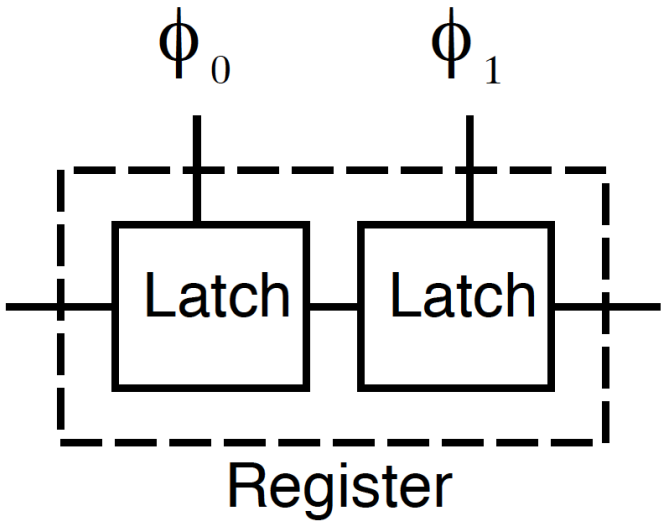
- What timing constraints do we have?





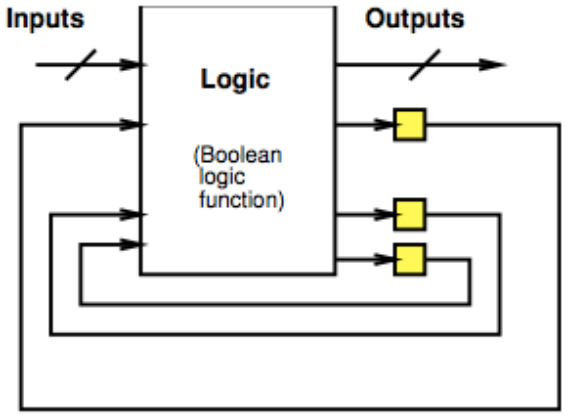
Latch Timing Issues

- What timing constraints do latches impose?
  - When can ϕ change? (ie. clk switching in relation to data)
    - Setup and hold time
  - How long must ϕ be high?
    - minimum clk period
  - Delay when ϕ is high?

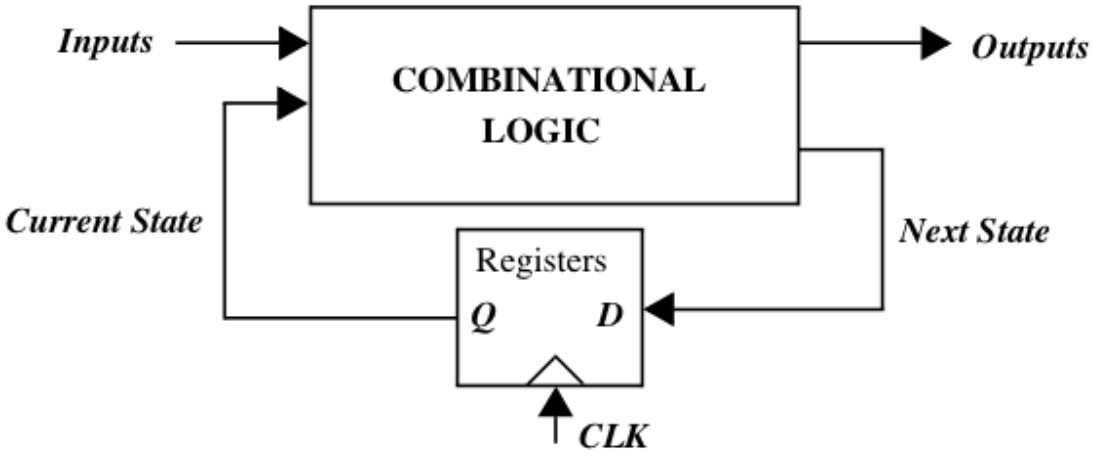


Clocking Discipline

- 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

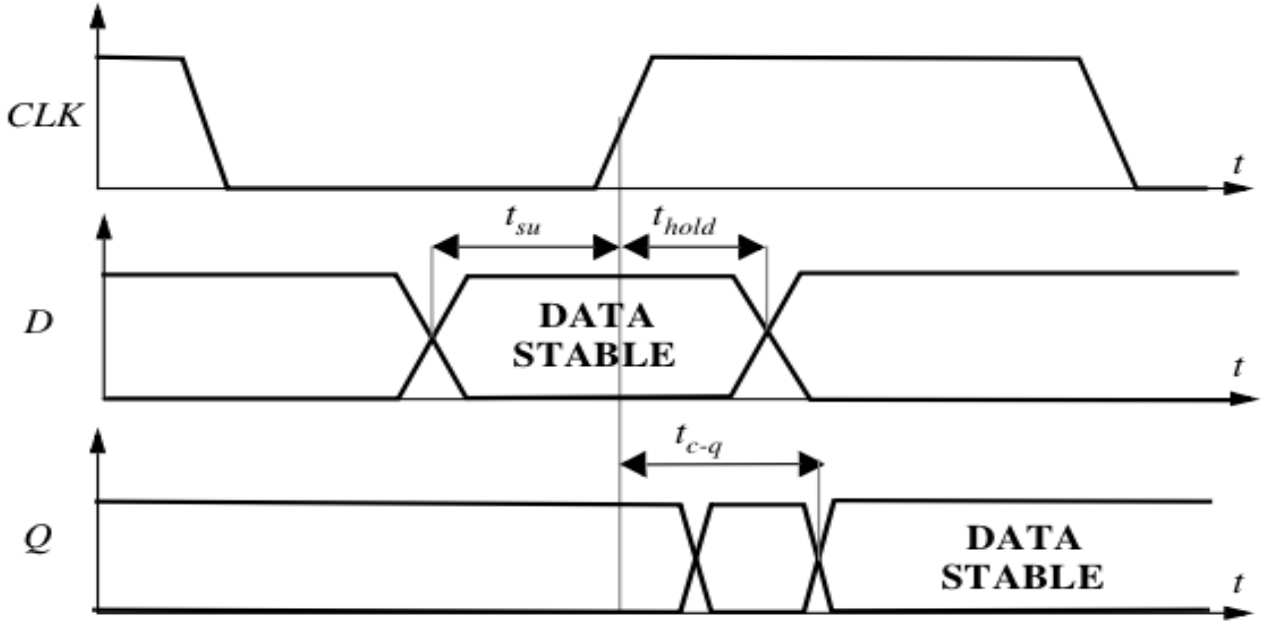


Worst case delays

$$T \geq t_{c\text{-}q} + t_{p\log ic} + t_{su}$$

$$t_{cd\text{register}} + t_{cd\text{logic}} \geq t_{hold}$$

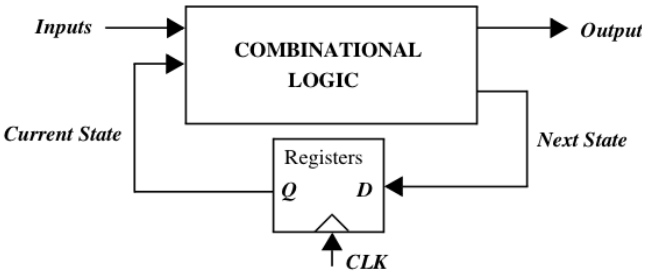
Minimum delays

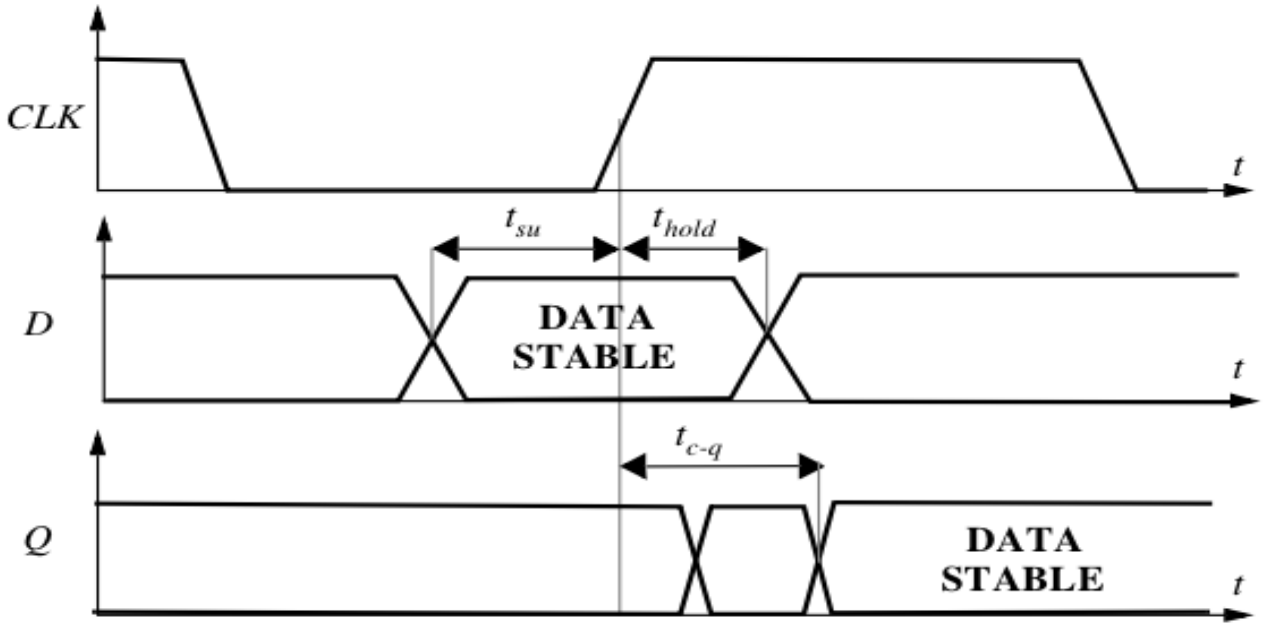


Latch Timing Issues

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

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

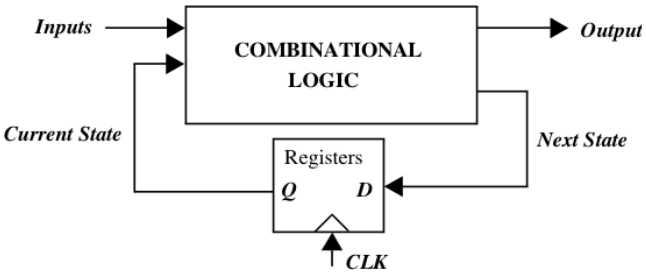


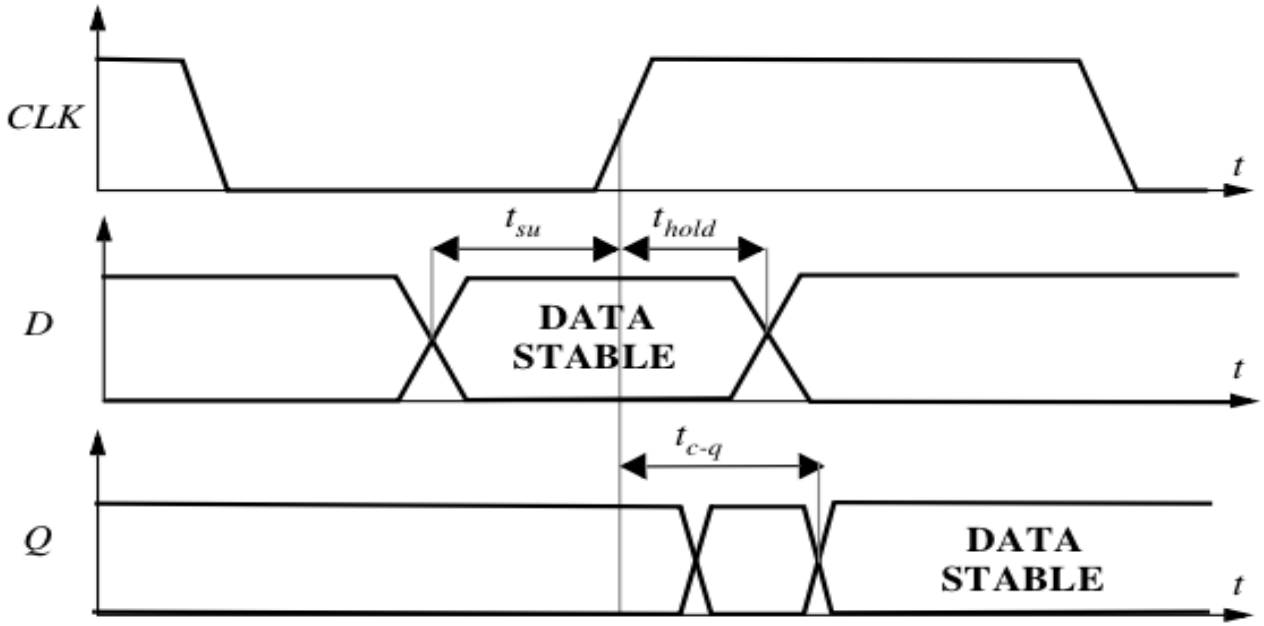


Latch Timing Issues

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

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

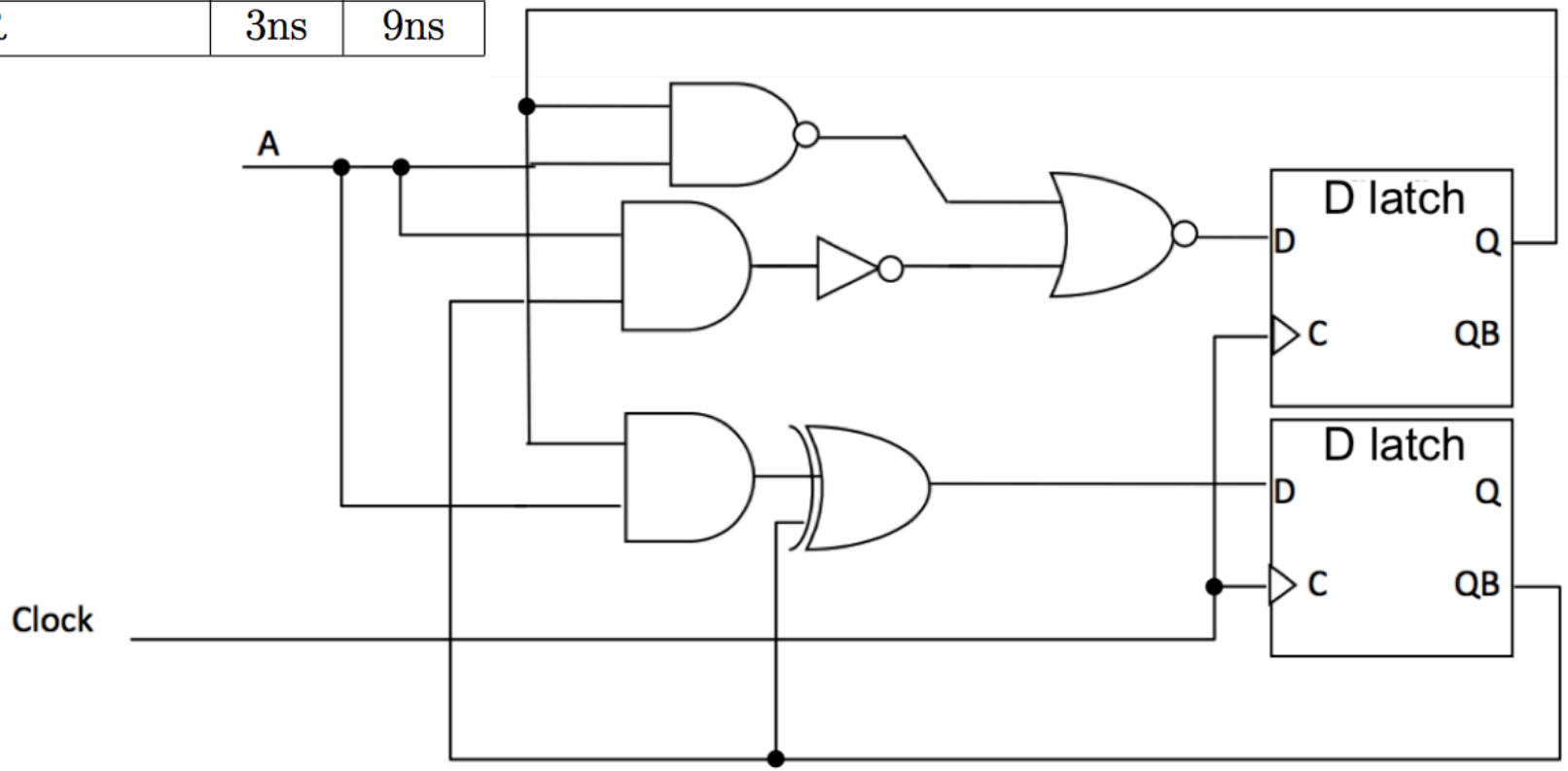




Timing Example (Preclass5)

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

Latch:		Min	Max
	Clock to Q	2ns	3ns
	Setup time	7ns	
	Hold time	6ns	

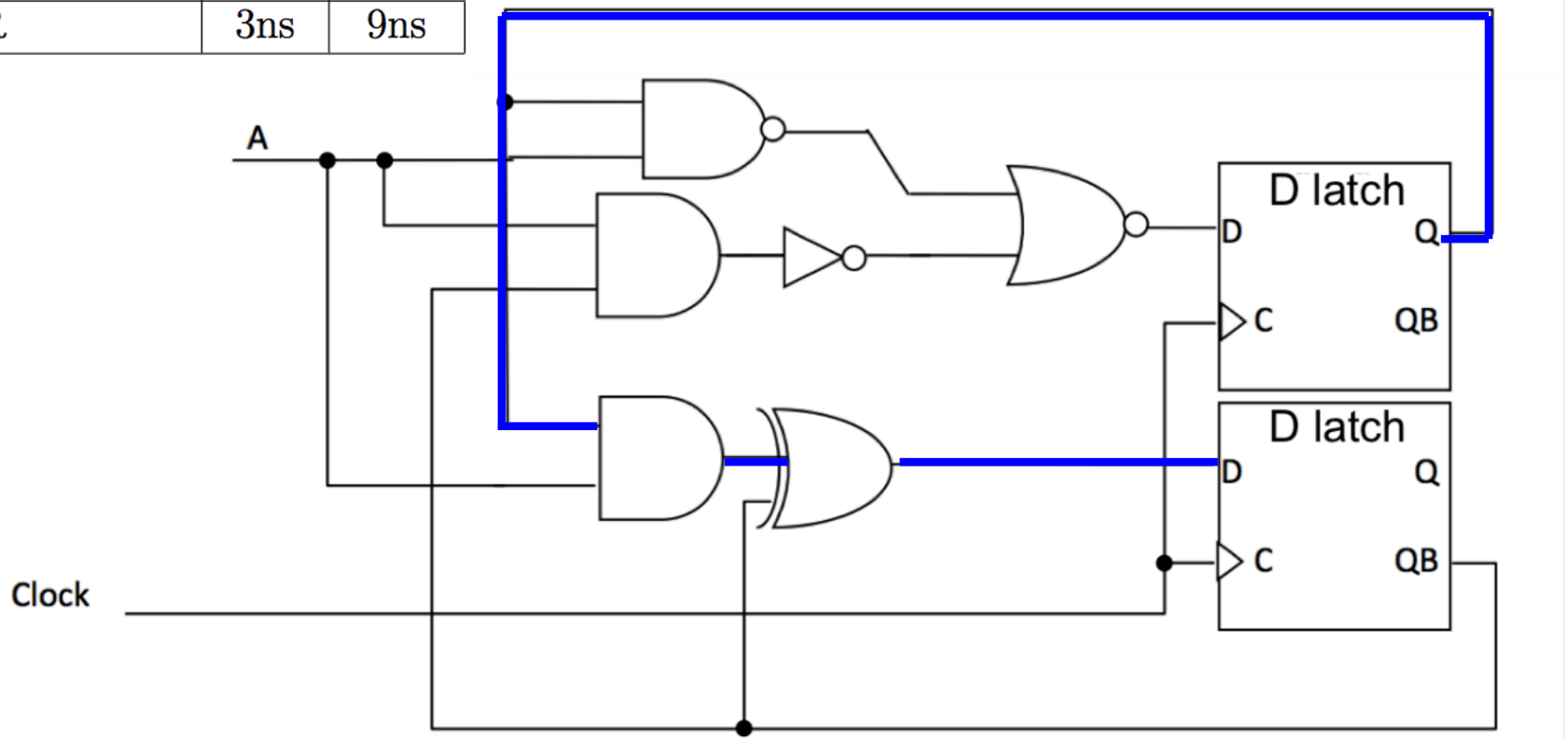


$$T \geq t_{c\text{-}q} + t_{plogic} + t_{su}$$

Timing Example (Preclass5)

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

Latch:		Min	Max
	Clock to Q	2ns	3ns
	Setup time	7ns	
	Hold time	6ns	

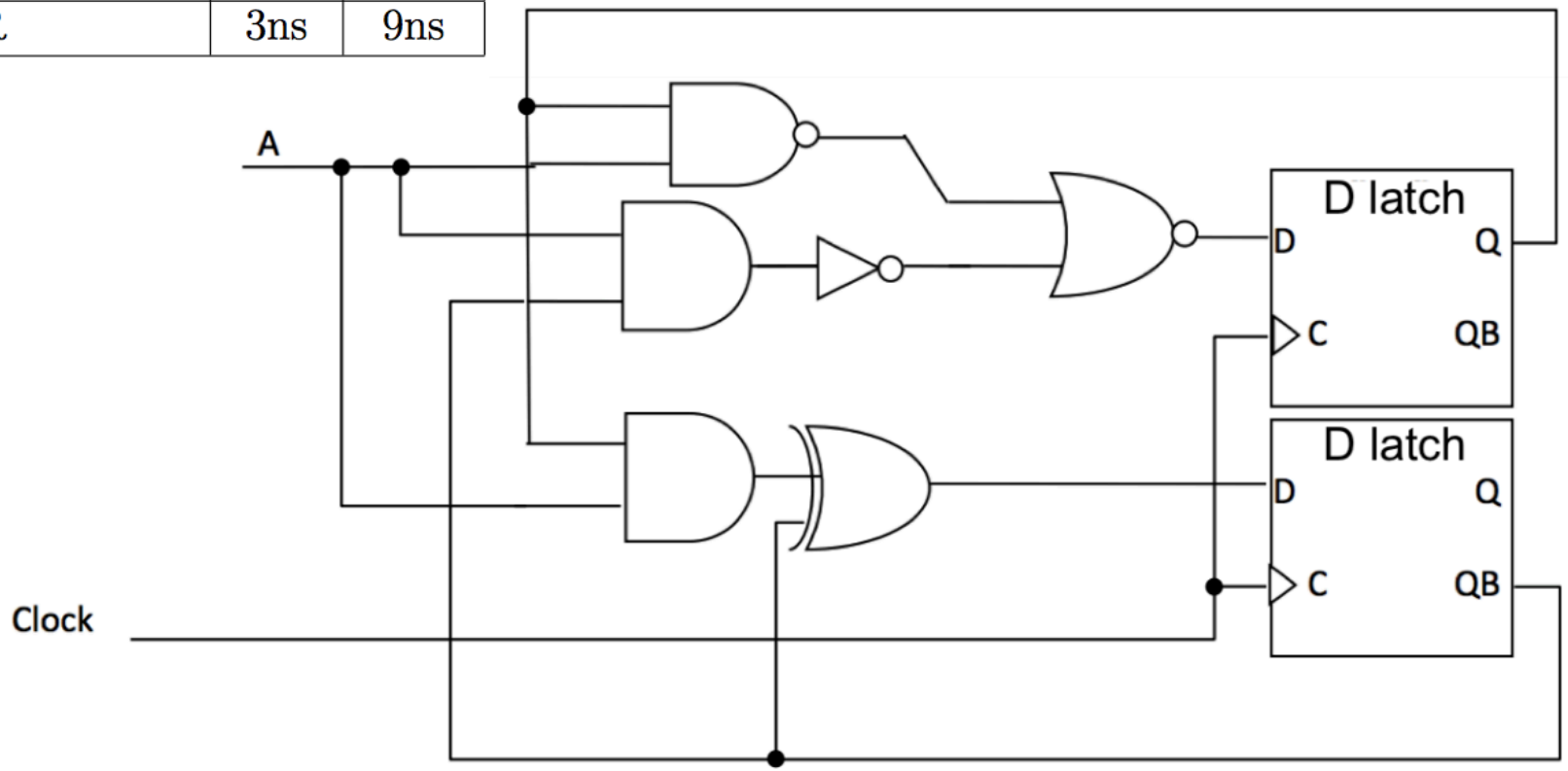


$$T \geq t_{c\text{-}q} + t_{plogic} + t_{su}$$

Timing Example (Preclass5)

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

Latch:		**Min**	**Max**
	Clock to Q	2ns	3ns
	Setup time	7ns	
	Hold time	6ns	

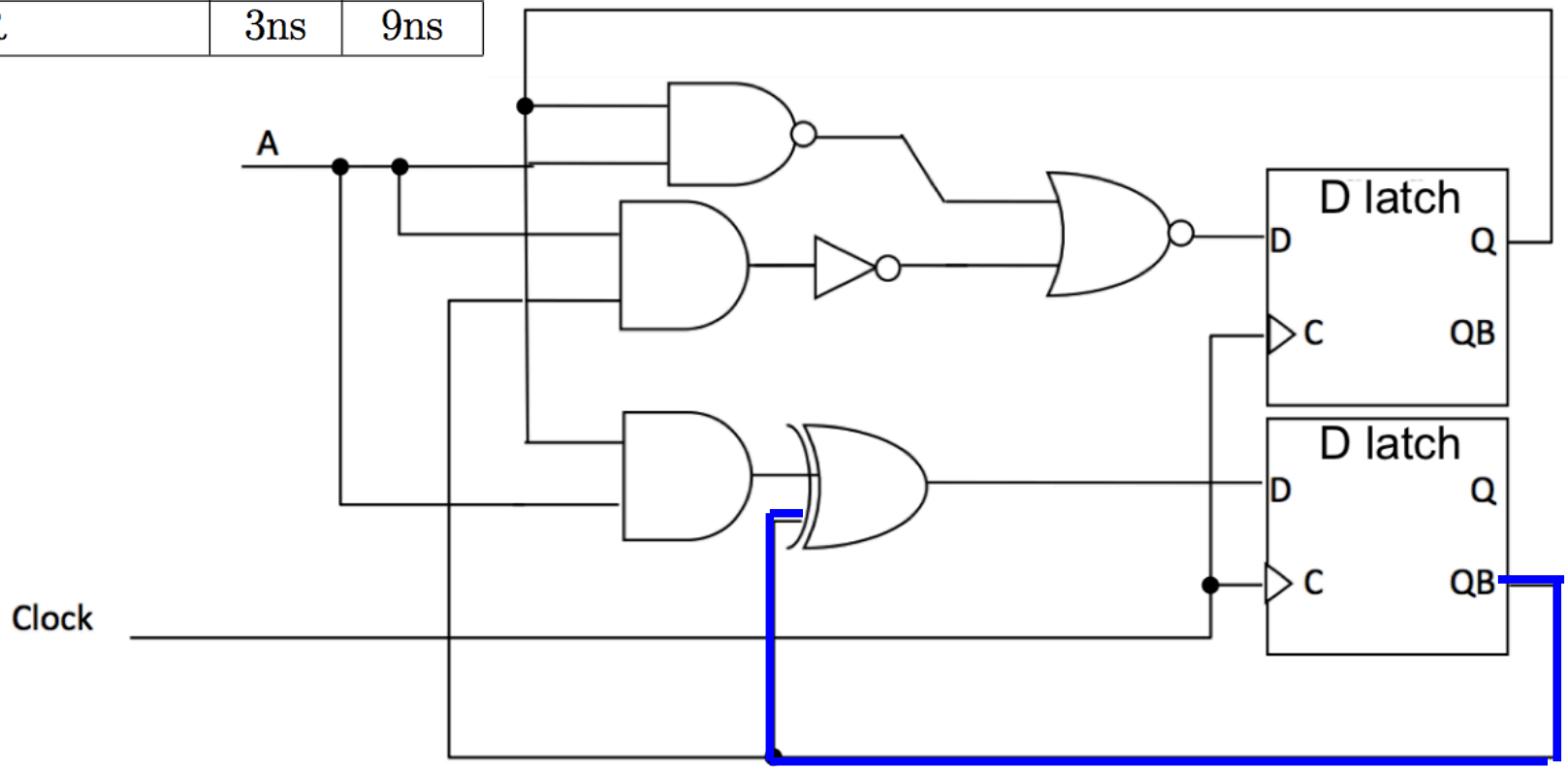


$$t_{cd\text{register}} + t_{cd\text{logic}} \geq t_{hold}$$

Timing Example (Preclass5)

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

Latch:		**Min**	**Max**
	Clock to Q	2ns	3ns
	Setup time	7ns	
	Hold time	6ns	

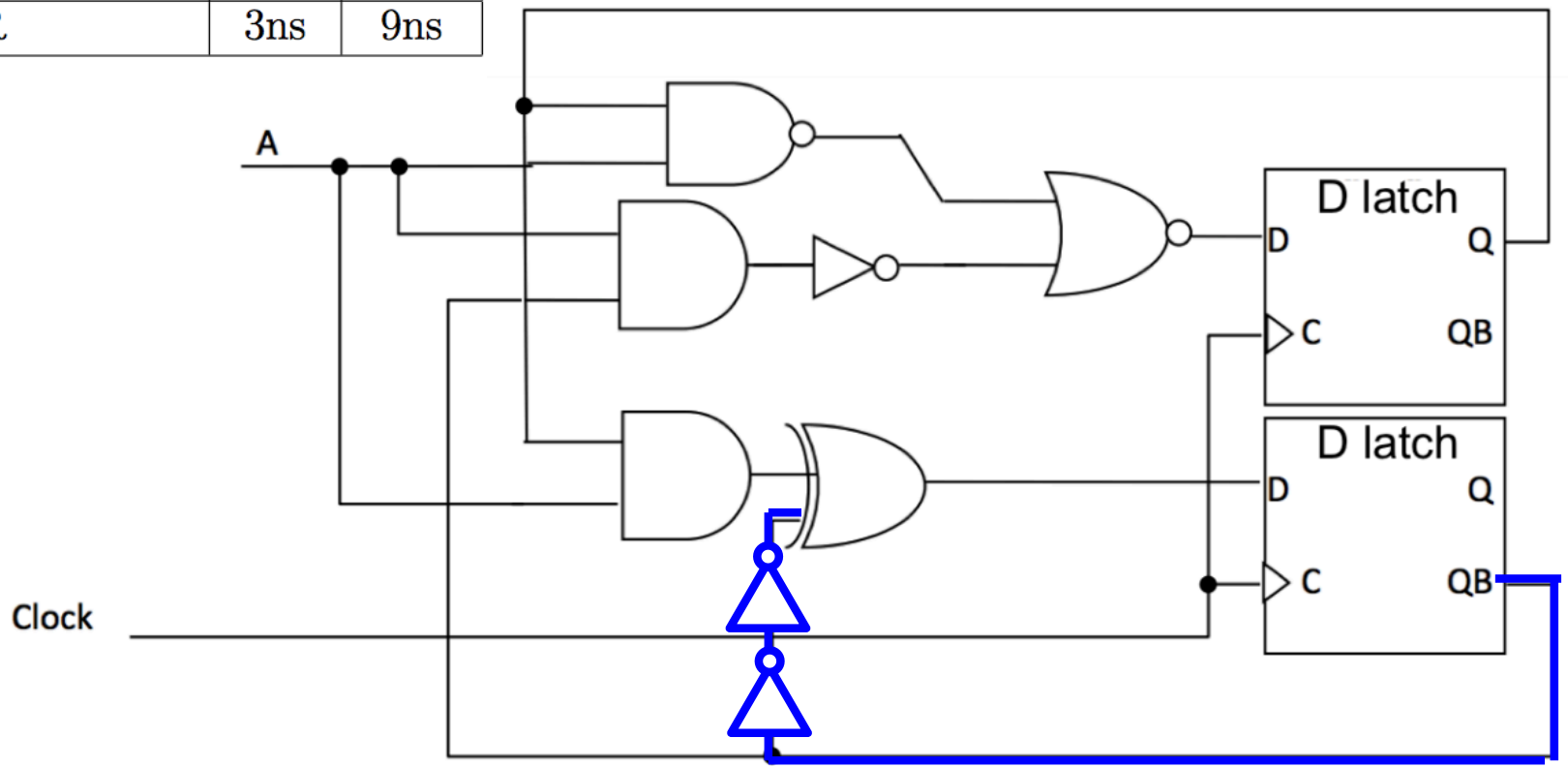


$$t_{cd\text{register}} + t_{cd\text{logic}} \geq t_{hold}$$

Timing Example (Preclass5)

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

Latch:		**Min**	**Max**
	Clock to Q	2ns	3ns
	Setup time	7ns	
	Hold time	6ns	



$$t_{cd\text{register}} + t_{cd\text{logic}} \geq t_{hold}$$

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

- Wednesday 4/3 Midterm 2 (this week)
  - During class in Moore 216
  - Lectures 1-14
  - Closed note, calculator allowed
  - All old exams online
    - 2010-2021
  - TA review session
    - 4/1 7-9pm in Towne 307 (tonight)
- HW6 release 4/3 (after Midterm 2)
  - Can do all of it now
  - Due **Wednesday 4/10**

Acknowledgement

- Prof. André DeHon (University of Pennsylvania)
- Prof. Jing Li (University of Pennsylvania)