ECE4740: Digital VLSI Design

Lecture 19: Dynamic latches/flip-flops

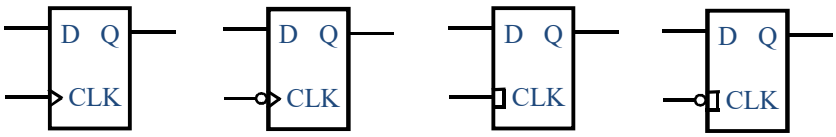
690

Recap

Timing, flip-flops, and latches

691

Common flip-flop and latch symbols



rising-edge triggered FF

falling-edge triggered FF

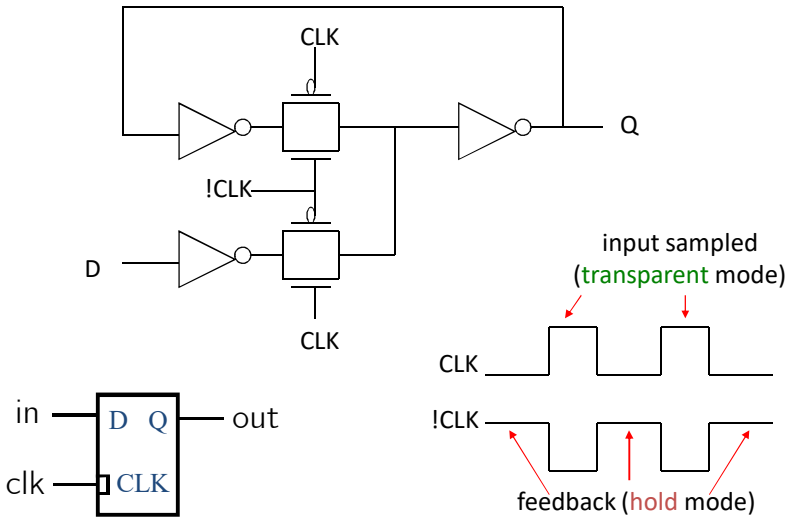
positive latch 1-transparent 0-hold

negative latch 0-transparent 1-hold

- Real-world flip-flops (and latches) may have more inputs and outputs, such as
  - Reset in, enable in, scan in, and !Q out

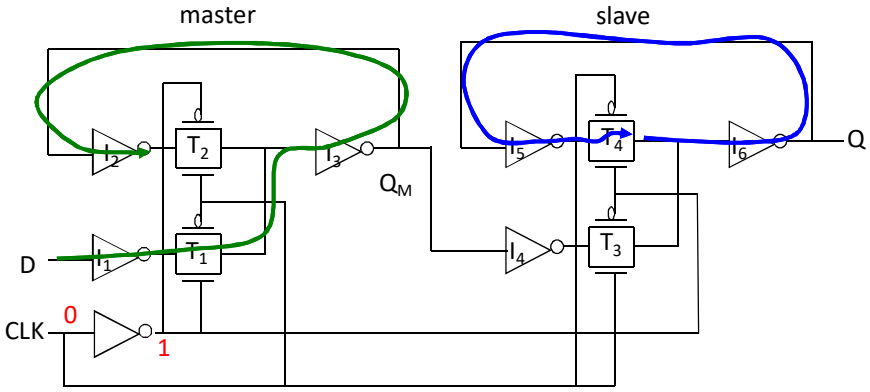
692

Positive latch: transparent if CLK=1



693

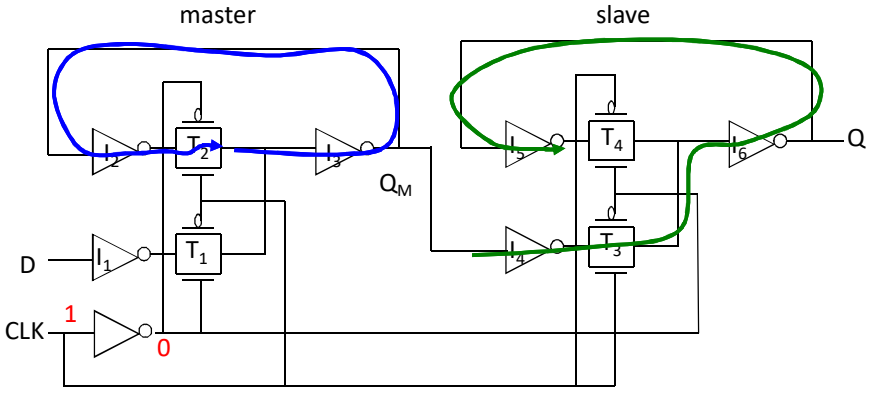
Positive-edge triggered MS flip-flop



CLK=0 → master transparent; slave hold

694

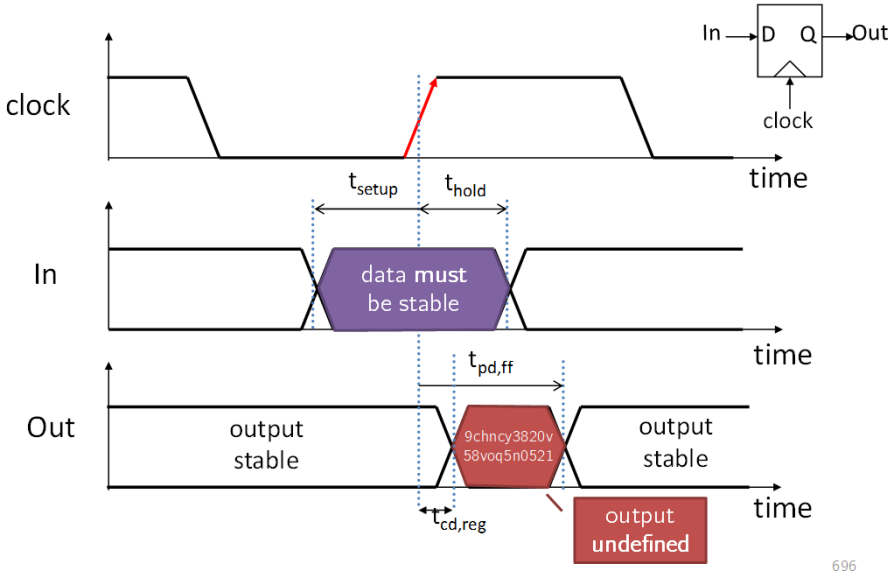
Positive-edge triggered MS flip-flop



CLK=1 → master hold; slave transparent

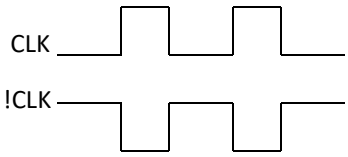
695

Setup and hold times

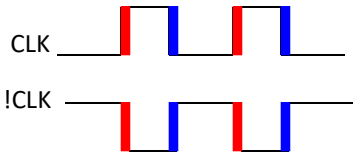


696

Non-ideal clocks: clock skew



ideal clocks



Non-ideal clocks

clock skew

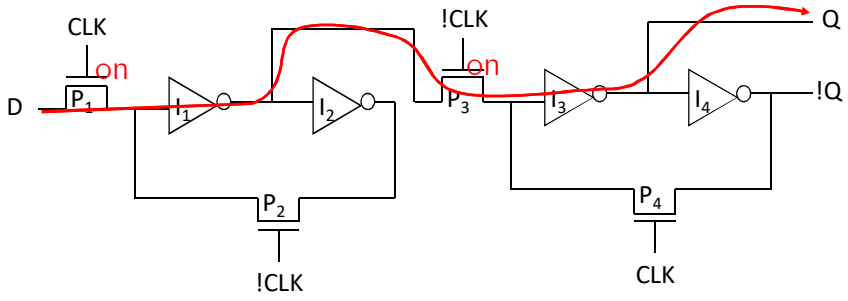
clock skew can happen due to uneven wire lengths, capacitances, different fan-outs, etc.

1-1 overlap

0-0 overlap

697

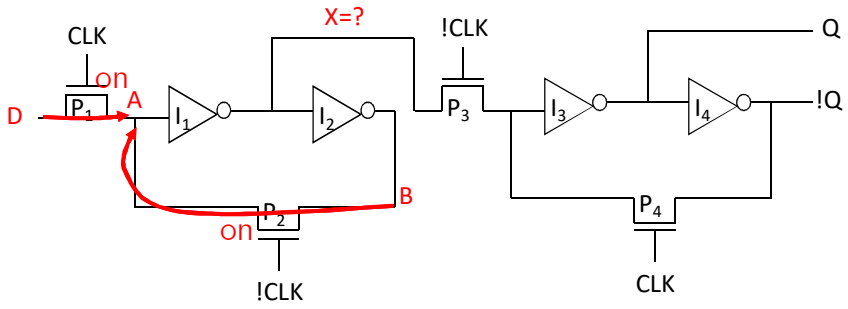
1-1 overlap is dangerous



- Direct path from D to Q during short time when both CLK and !CLK are high
  - Happens during 1-1 overlap

698

1-1 overlap is dangerous (cont'd)

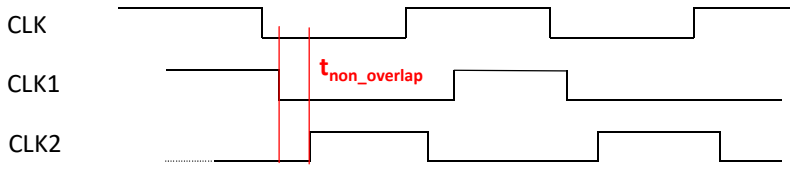


- Both B and D are driving A when CLK and !CLK are both high (1-1 overlap)

699

Generating a non-1-1-overlapping clock

- To avoid overlapping clocks 1-1 we need
  - tools for accurate timing analysis OR
  - non-1-1-overlapping clock signals
  - One can use SR-latch to generate such clocks



700

Building sequential logic with fewer transistors

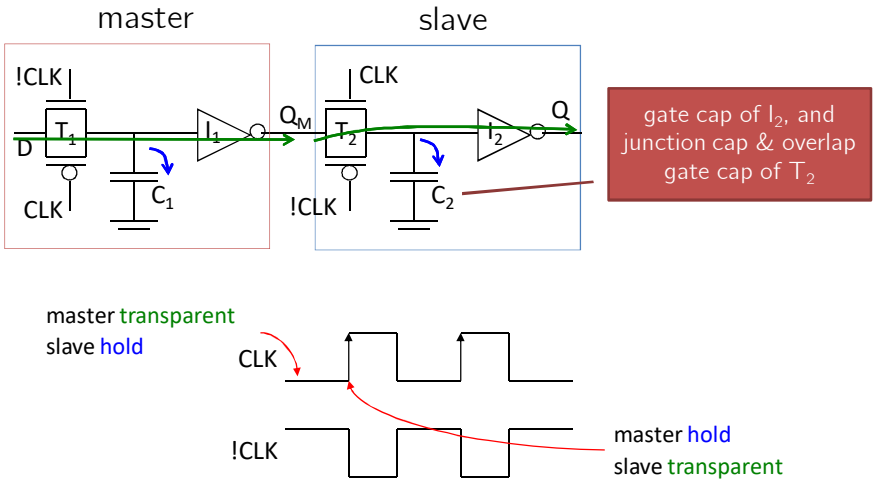
Dynamic latches and flip-flops

701

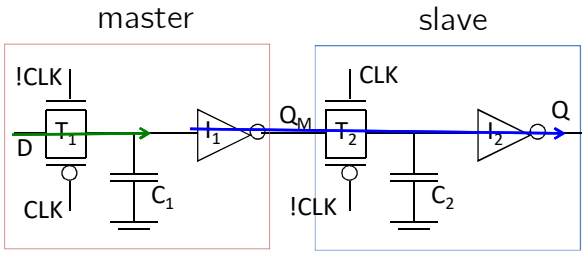
Static vs. dynamic storage cells

- Static cells use bistable element with feedback (regeneration)
  - Preserve state as long as power is on
- Static storage is preferred when updates are infrequent (clock gating etc.)
- Dynamic storage on parasitic capacitors
  - Preserve state only for milliseconds
- Dynamic storage cells are usually smaller, achieve higher speed and consume lower power

Dynamic edge-triggered flip-flop



Dynamic ET flip-flop (cont'd)

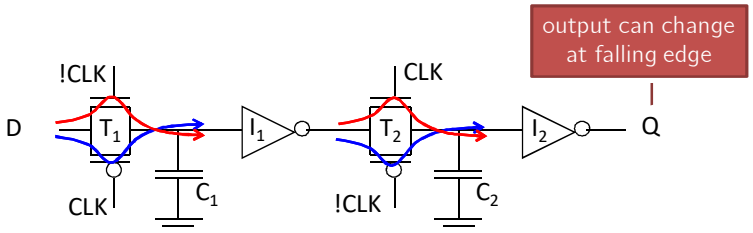


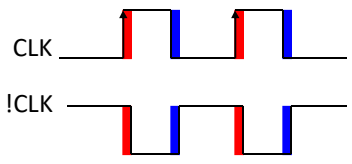
t_{su} = t_{pd_tx}
t_{hold} = zero
t_{pd} = $2t_{pd\_inv}+t_{pd\_tx}$

- Requires only 8 transistors; clock load = 4
- Dynamic nodes need periodical refresh

704

Issue 1: race conditions





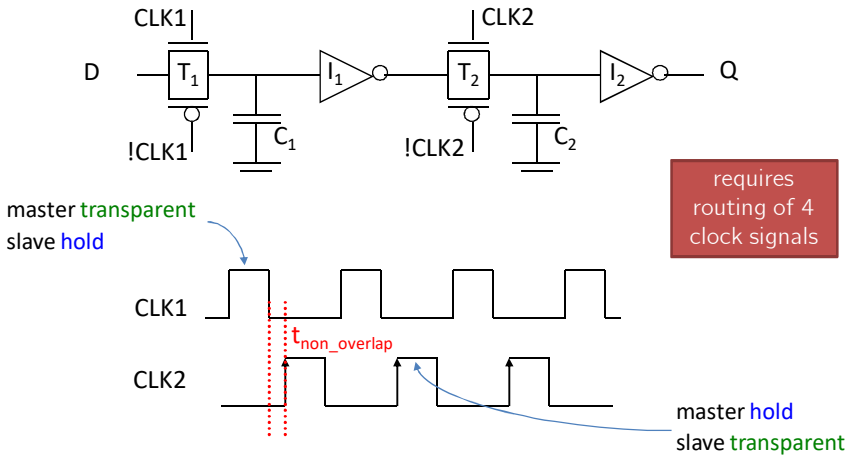
0-0 overlap race condition
$t_{overlap0\text{-}0} < t_{T1} + t_{I1} + t_{T2}$

1-1 overlap race condition
$t_{overlap1\text{-}1} < t_{hold}$

data must be stable during high phase

705

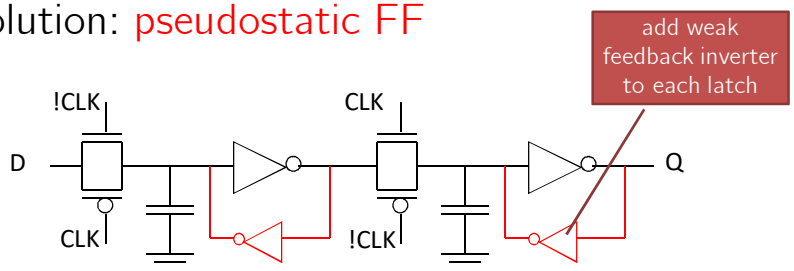
Solution: non-overlapping clocks



706

Issue 2: robustness

- Dynamic flip-flops suffer from
  - Coupling between signal nets and internal storage nodes (can destroy FF state)
  - Leakage currents cause state to leak with time
- Solution: pseudostatic FF



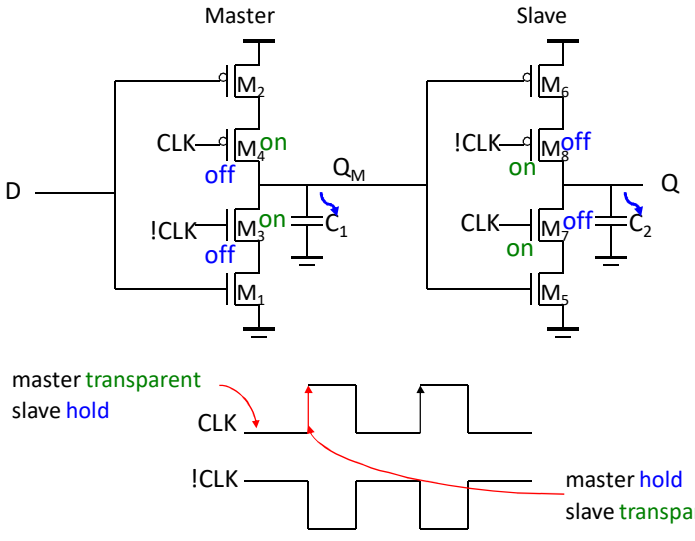
707

A clock-skew insensitive approach

The C²MOS register

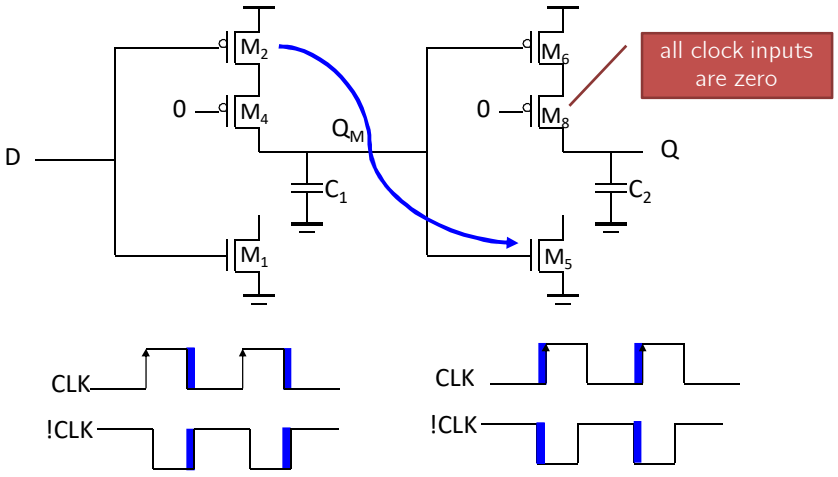
708

C²MOS (clocked CMOS) ET FF



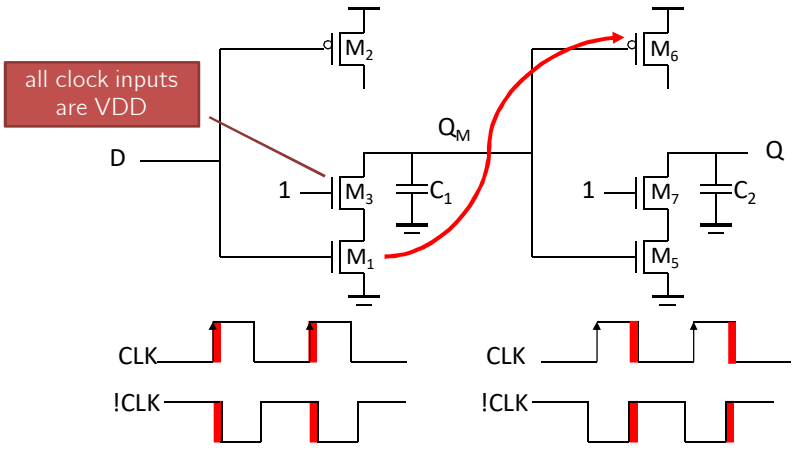
709

C²MOS FF: 0-0 overlap



710

C²MOS FF: 1-1 overlap

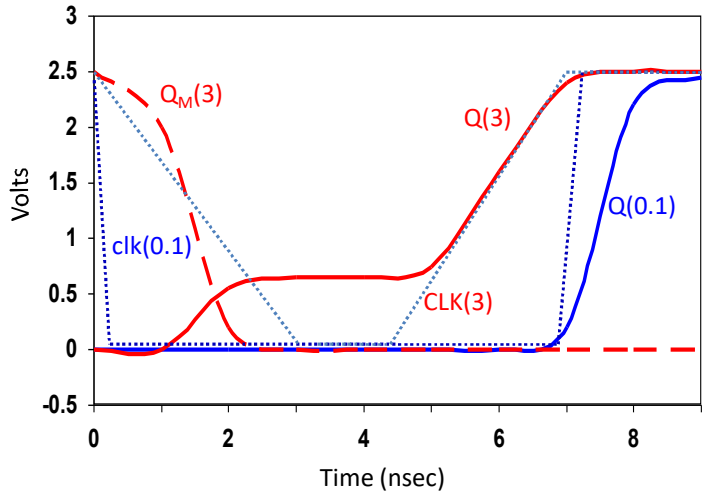


1-1 overlap constraint

$t_{overlap1\text{-}1} < t_{hold}$

711

(Slope matters: transient response)



For a
0.1 ns clock

For a
3 ns clock
(race condition
exists)

712

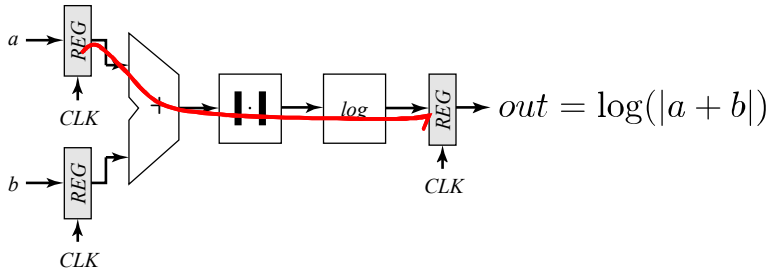
Image adapted from: Digital Integrated Circuits (2nd Edition) by Rabaey, Chandrakasan, Nikolic

For high-throughput designs

Pipelining & retiming

713

Consider the timing of this circuit

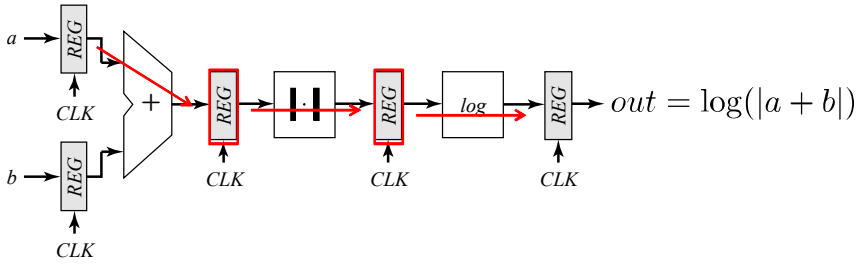


- Critical path:
  $T_{min}=t_{pd,ff}+t_{pd,add}+t_{pd,abs}+t_{pd,log}+t_{su,ff}$

714

Image taken from: Digital Integrated Circuits (2nd Edition) by Rabaey, Chandrakasan, Nikolic

Pipelining reduces critical path

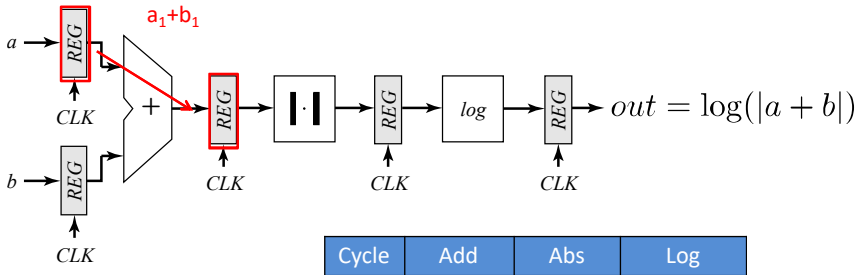


- Insert pipeline registers (flip-flops)
- Shortens critical path!
  $T_{pipe,min}=t_{pd,ff}+\max\{t_{pd,add}, t_{pd,abs}, t_{pd,log}\}+t_{su,ff}$

715

Image taken from: Digital Integrated Circuits (2nd Edition) by Rabaey, Chandrakasan, Nikolic

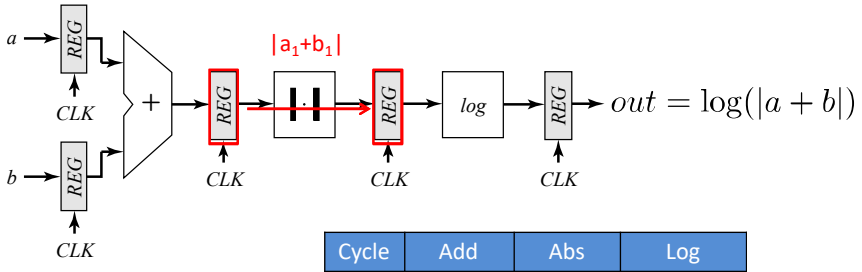
Pipelining



Cycle	Add	Abs	Log
1	a_1+b_1		
2	a_2+b_2	$\lvert a_1+b_1\rvert$	
3	a_3+b_3	$\lvert a_2+b_2\rvert$	$\log\lvert a_1+b_1\rvert$
4	a_4+b_4	$\lvert a_3+b_3\rvert$	$\log\lvert a_2+b_2\rvert$
…	…	…	…

716

Pipelining (cont'd)

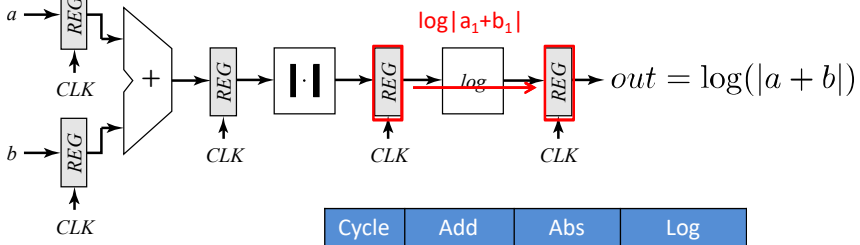


new data item inserted in pipeline

Cycle	Add	Abs	Log
1	a_1+b_1		
2	a_2+b_2	$\lvert a_1+b_1\rvert$	
3	a_3+b_3	$\lvert a_2+b_2\rvert$	$\log\lvert a_1+b_1\rvert$
4	a_4+b_4	$\lvert a_3+b_3\rvert$	$\log\lvert a_2+b_2\rvert$
…	…	…	…

717

Pipelining (cont'd)

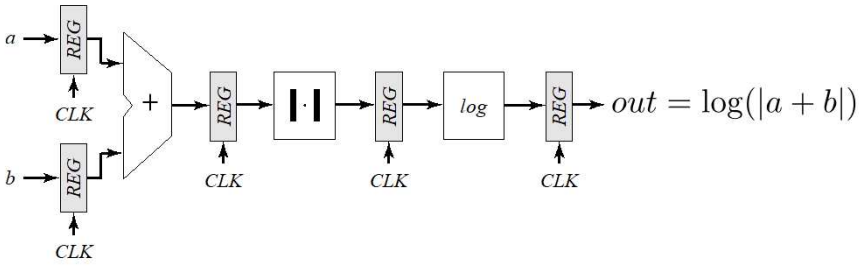


Cycle	Add	Abs	Log
1	a_1+b_1		
2	a_2+b_2	$\lvert a_1+b_1\rvert$	
3	a_3+b_3	$\lvert a_2+b_2\rvert$	$\log\lvert a_1+b_1\rvert$
4	a_4+b_4	$\lvert a_3+b_3\rvert$	$\log\lvert a_2+b_2\rvert$
...	...	...	...

new data item inserted in pipeline

718

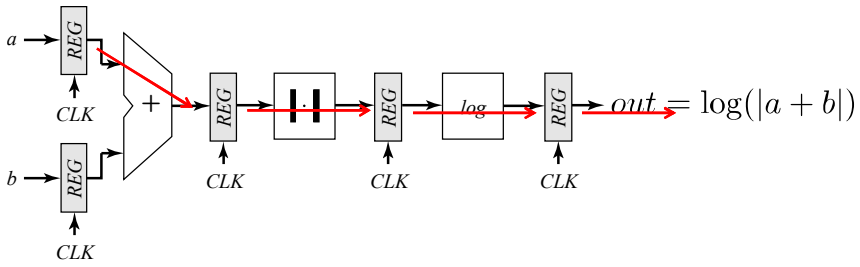
Pipelining improves throughput!



- Processes 1 data item per clock cycle at higher f_{max} → higher throughput (time per data item $T_{min,pipe}$)
- Ideally: $T_{min,pipe} = t_{pd,ff}+t_{pd,logic}/N+t_{su,ff}$ with N stages
- Throughput limit: $T_{min,pipe} \geq t_{pd,ff}+t_{su,ff}$

719

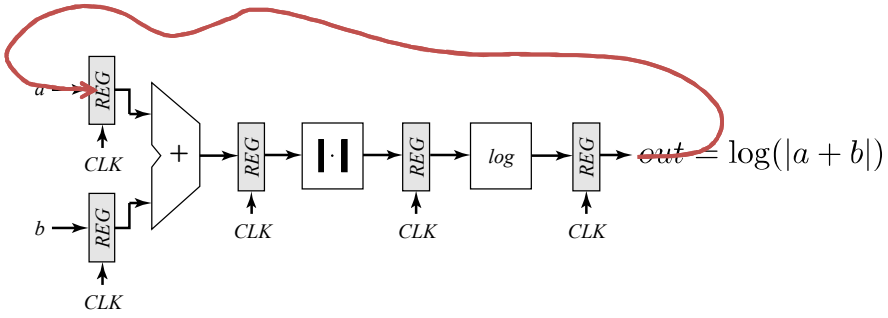
Pipelining introduces latency



- Latency = # of cycles for data to propagate from input to output
- Latency = 4 (four rising clock edges)

720

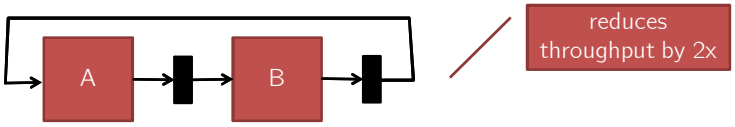
The feedback problem



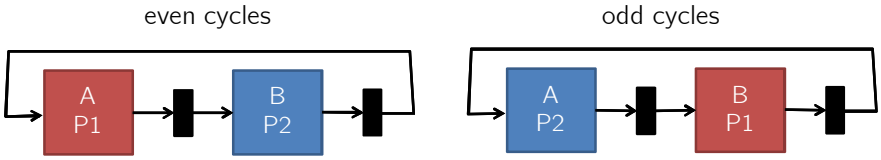
- If feedback path is present, latency will reduce throughput (circuit has to wait for data)
- Problem in processors and application specific integrated circuits (data dependencies)

721

Solution: Pipeline interleaving

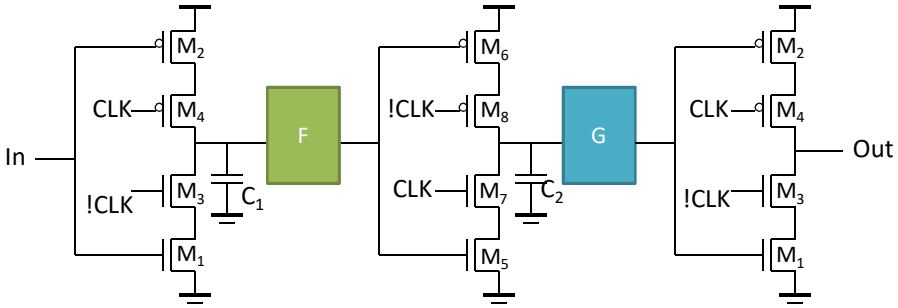


- Idea: Process independent problems in an interleaved manner in the same hardware



722

Pipelining using C^{2}MOS

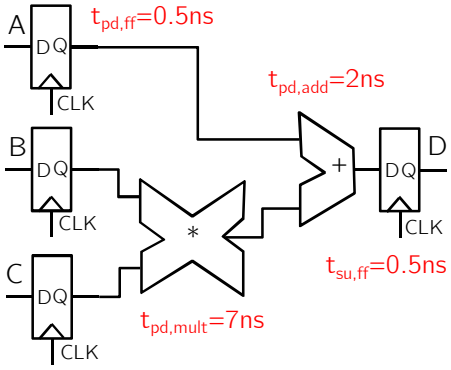


- Circuit is race-condition free (NORA) if functions F and G are non-inverting!

723

Your turn: pipeline a MAC unit

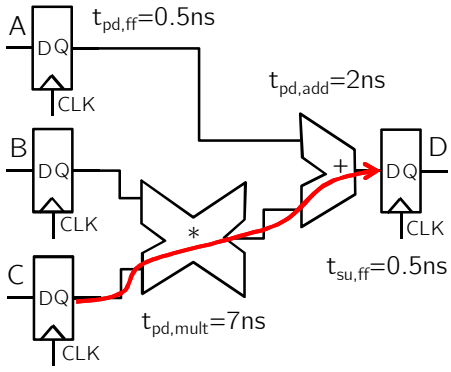
multiply-accumulate (MAC) unit: D=B*C+A



- What is the max. clock frequency?
- Where is the critical path?
- Insert a single pipeline stage
- What is the max. clock frequency after pipelining?

724

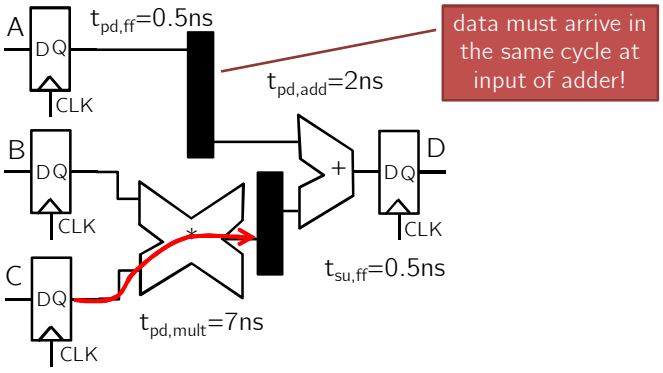
Critical path and max. clock freq.



- $T_{min}=t_{pd,ff}+t_{pd,mult}+t_{pd,add}+t_{su,ff}=10ns$
- $f_{max}=100MHz$

725

Pipelining: max. clock freq. now?



- $T_{min,pipe}=t_{pd,ff}+t_{pd,mult}+t_{su,ff}=8ns$
- $f_{max}=125MHz$

726