EE273 Lecture 9
Advanced Signaling

February 12, 2001

William J. Dally
Computer Systems Laboratory
Stanford University
billd@csl.stanford.edu

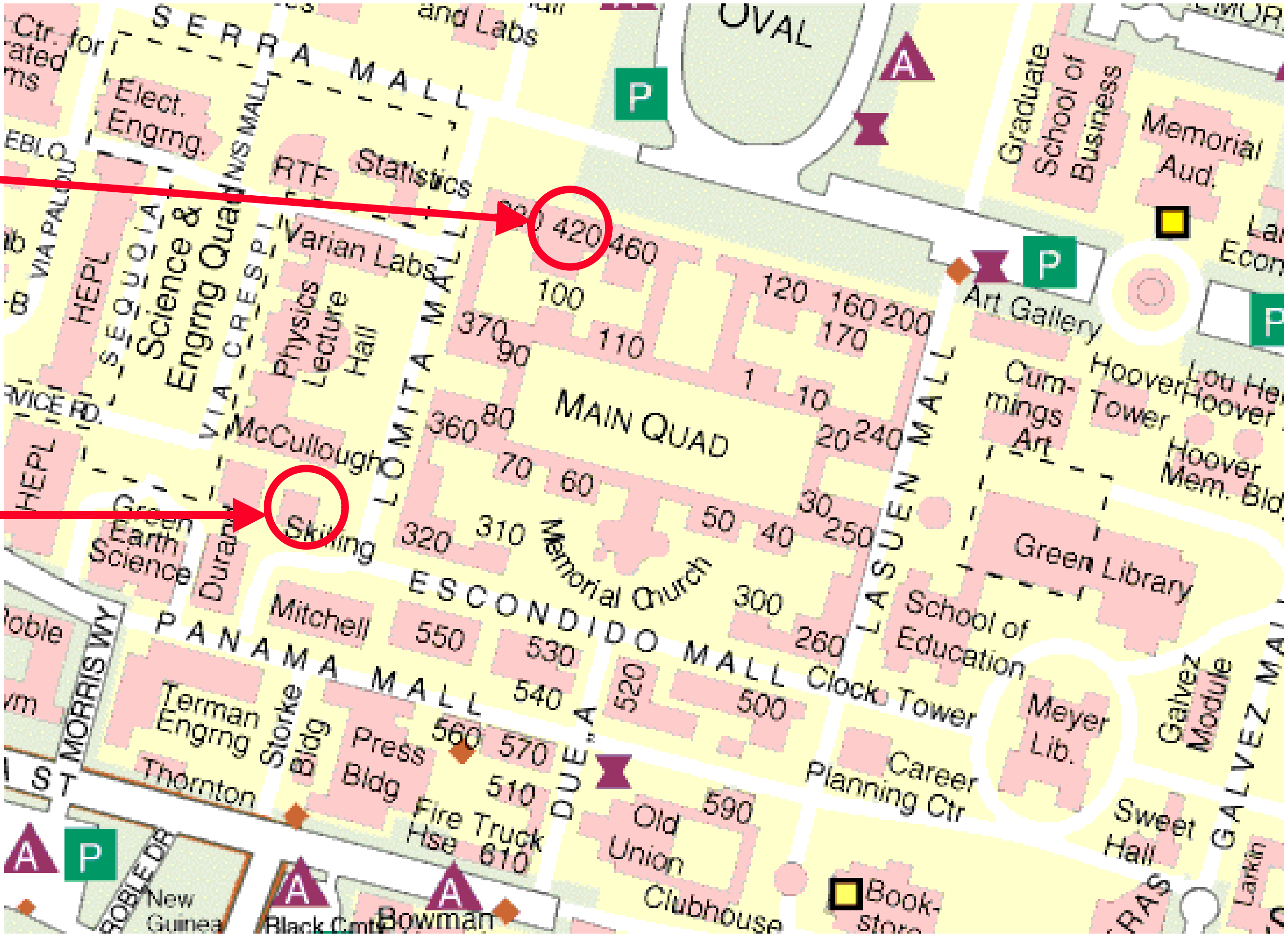
Logistics

- Midterm Tonight!
  - Room change – 420-041
  - 7:00PM to 9:00PM
  - Open notes, open book
- Reading
  - Sections 9.6.1 through 9.6.5
  - Complete before class on Wednesday 2/14

Midterm Location

The midterm
is here

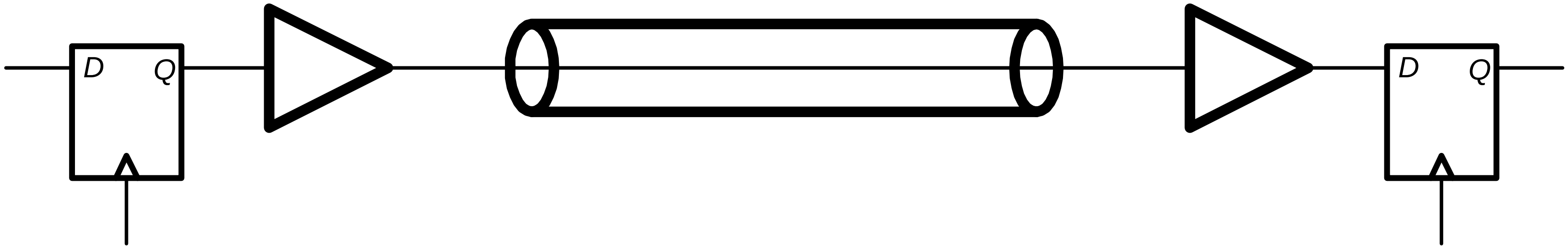
Right now we
Are here

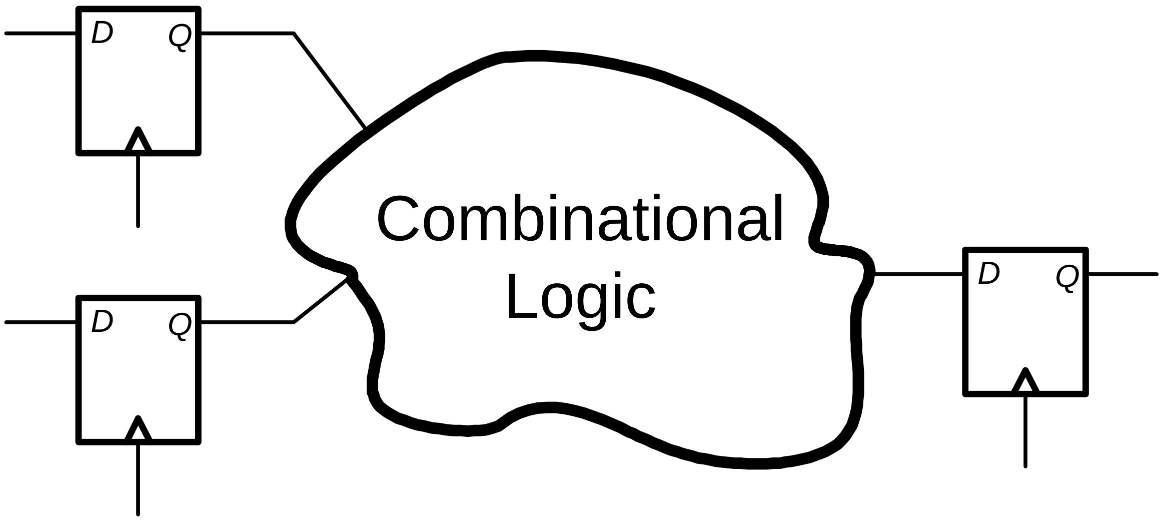


A Quick Overview

- Timing
  - defines when a value is present on a signal line
- Signals, values, and events
  - at an instant in time a signal holds a value
  - over time, a signal carries events (time-value pairs)
  - there are many ways of encoding events
- Clock domains
  - a group of signals *bundled* with a single clock forms a domain
  - need synchronizers to cross between domains
- Timing components
  - delay elements
  - combinational logic
  - clocked-storage elements
- Timing uncertainty
  - skew
  - jitter
- Synchronous Timing
  - all signals in one domain
  - narrow windows of allowed operating frequency
- Pipeline Timing
  - clock forwarded with signal
  - operates across wide range of frequencies

Timing
The Big Picture

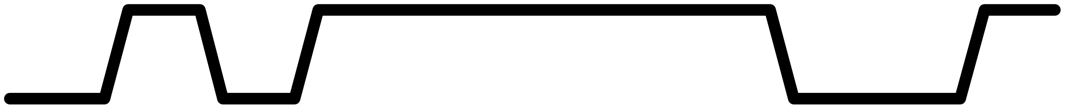




Combinational Logic

Was it 5 or was it 6?

- Timing defines *when* a value is present on a line
  - how many 1s does a high pulse correspond to?



  - requires a convention to decide when one “1” ends and the next begins
  - most conventions mark boundaries with *transitions*
    - of the signal itself
    - of a related *clock* signal
  - uncertainty in the timing of these boundaries, not absolute delay, limits operating speed

Signals, Values, and Events

- At a given instant in time a signal has a *value*
  - “1” or “0” for a binary signal
- Over time, a signal has a sequence of *events*
  - e.g., 01011111001
  - each event is a *value* and a *time*
    - 0 at t=0, 1 at t=500ps, (0,1ns), (1,1.5ns), (1,2.0ns)
  - need to encode an event even when the signal doesn’t change value

Encoding Events

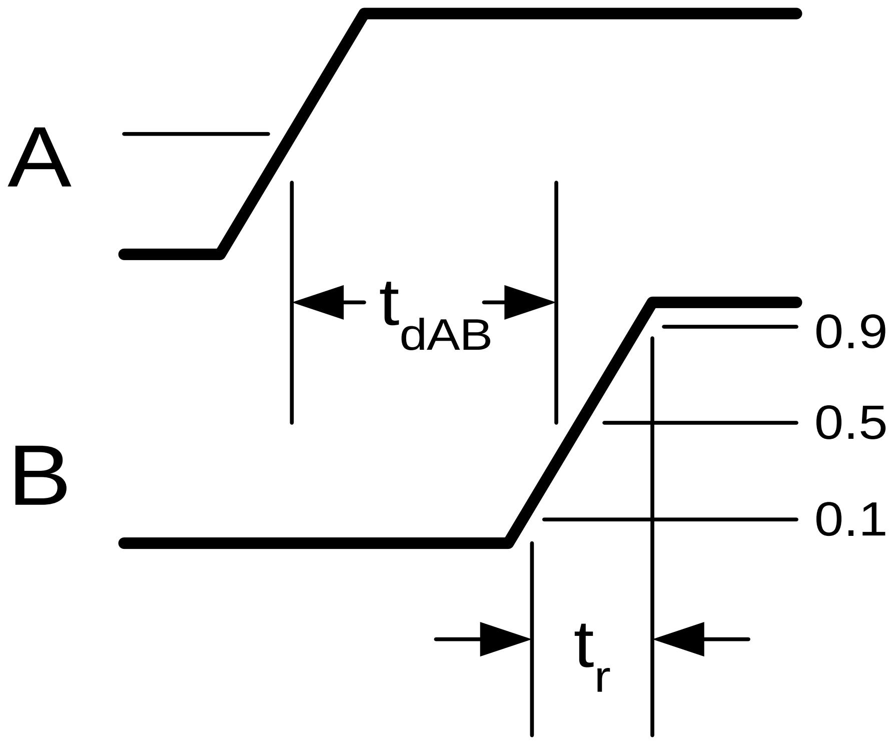
- There are many ways of encoding events
- *Periodic* encodings implicitly define when events happen
  - still need a transition every so often to keep synchronized
- *Aperiodic* encodings explicitly define when events happen
  - with a transition for every event
  - per-signal (e.g., dual-rail) vs. bundled
  - RZ vs NRZ

Clock Domains

- A group of signals bundled with one clock defines a *clock domain*
- Signals within a clock domain can be combined and sampled by the clock
- Care is required when switching between clock domains
  - – depends on relationship between the two clocks
    - mesochronous
    - plesiochronous
    - periodic
    - asynchronous
  - – a synchronizer is probably required

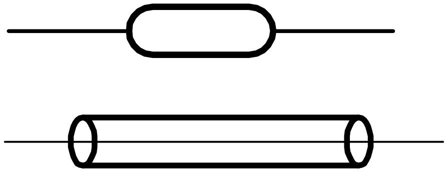
Delay and Rise/Fall Times

- delay (from 50% to 50%)
- rise time (from 10% to 90%) or 20 to 80 or 25 to 75

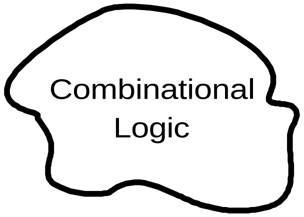


Timing Building Blocks

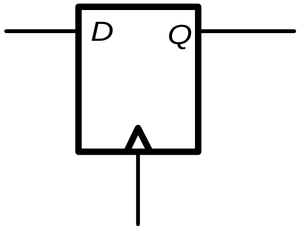
- delay elements
  - described by
    - nominal delay
    - timing uncertainty, skew and jitter



- combinational logic
  - contamination delay
  - propagation delay

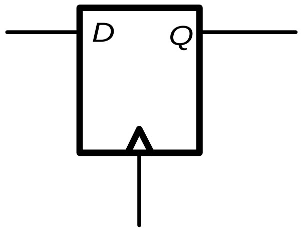


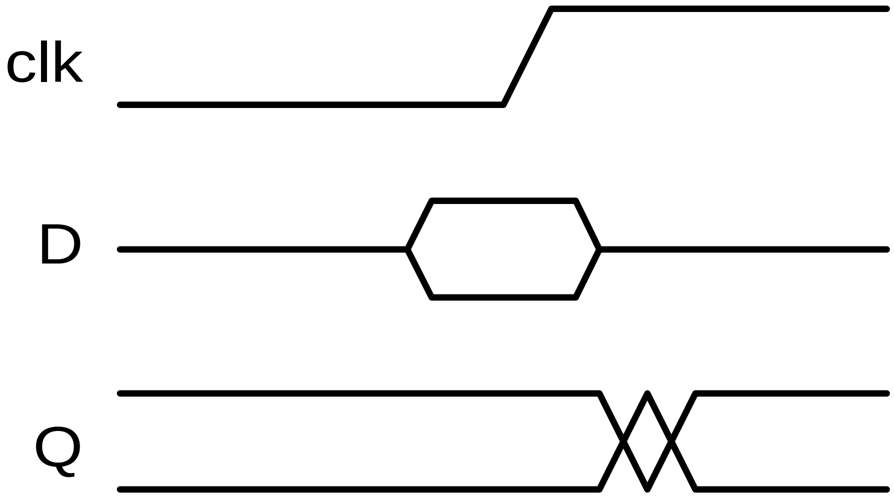
- clocked storage elements
  - align a signal to a clock
    - signal *waits* to be sampled by a clock
    - output is held steady until the next clock



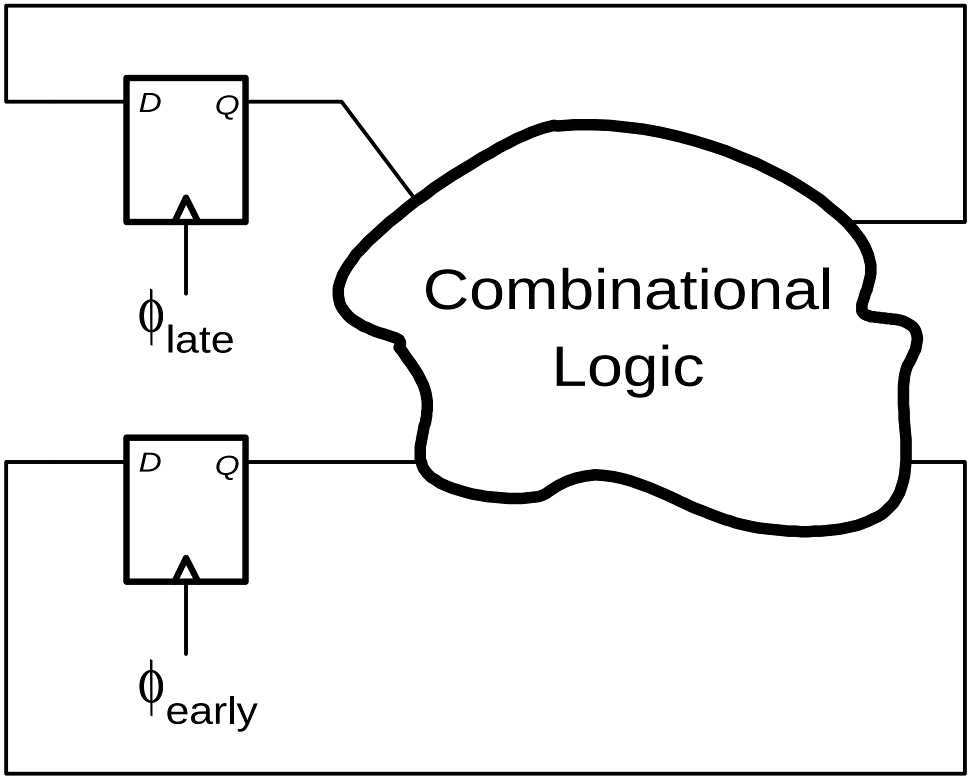
The Edge-Triggered Flip-Flop

- Conceptually simplest clocked storage element
- Samples data on rising edge of clock
  - data must remain valid during an *aperture* time during sampling
- Output is held steady until next clock
  - output is steady until a *contamination delay* after clock
  - output has correct value a *propagation delay* after clock



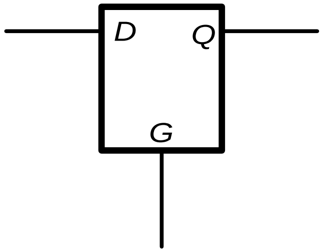


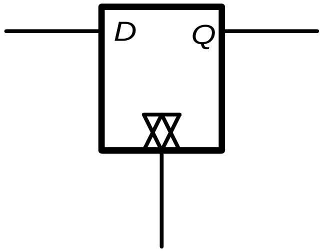
Maximum and Minimum Delay Constraints



Other Clocked Storage Elements

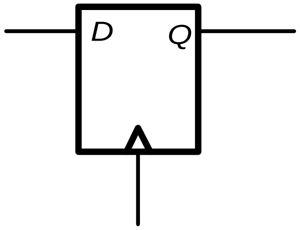
- level sensitive latch
  - passes data when enable high
  - holds output when enable low
- dual edge-triggered flip-flop
  - samples data on both edges of the clock
  - allows clock to run at *same* frequency as the data

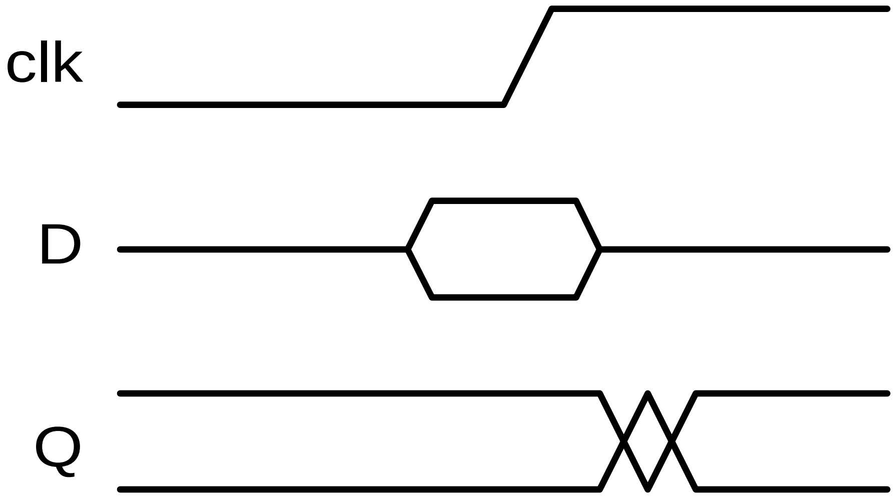




Properties of Clocked Storage Elements

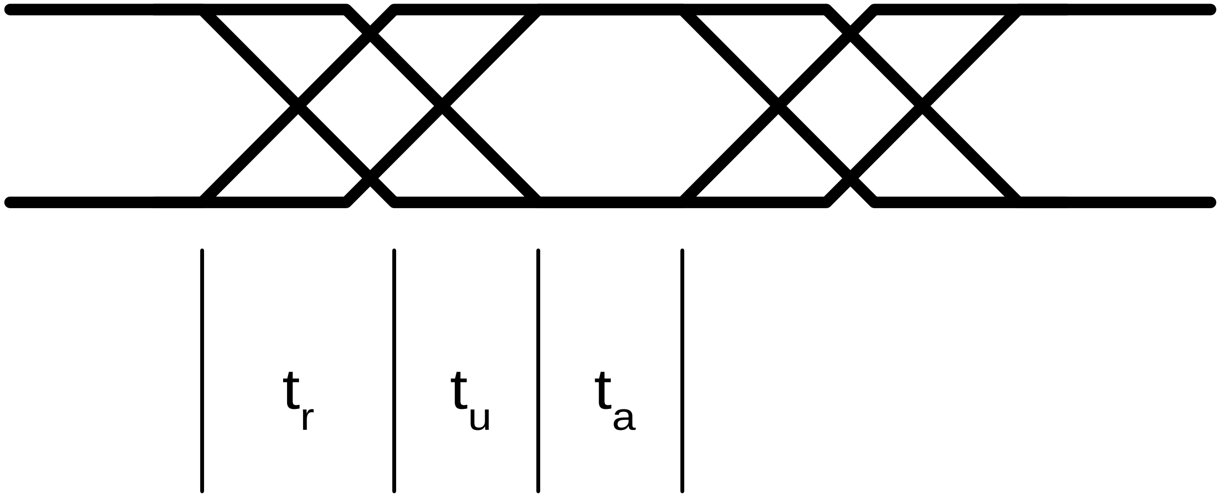
- Setup, hold, and delay
- Aperture time





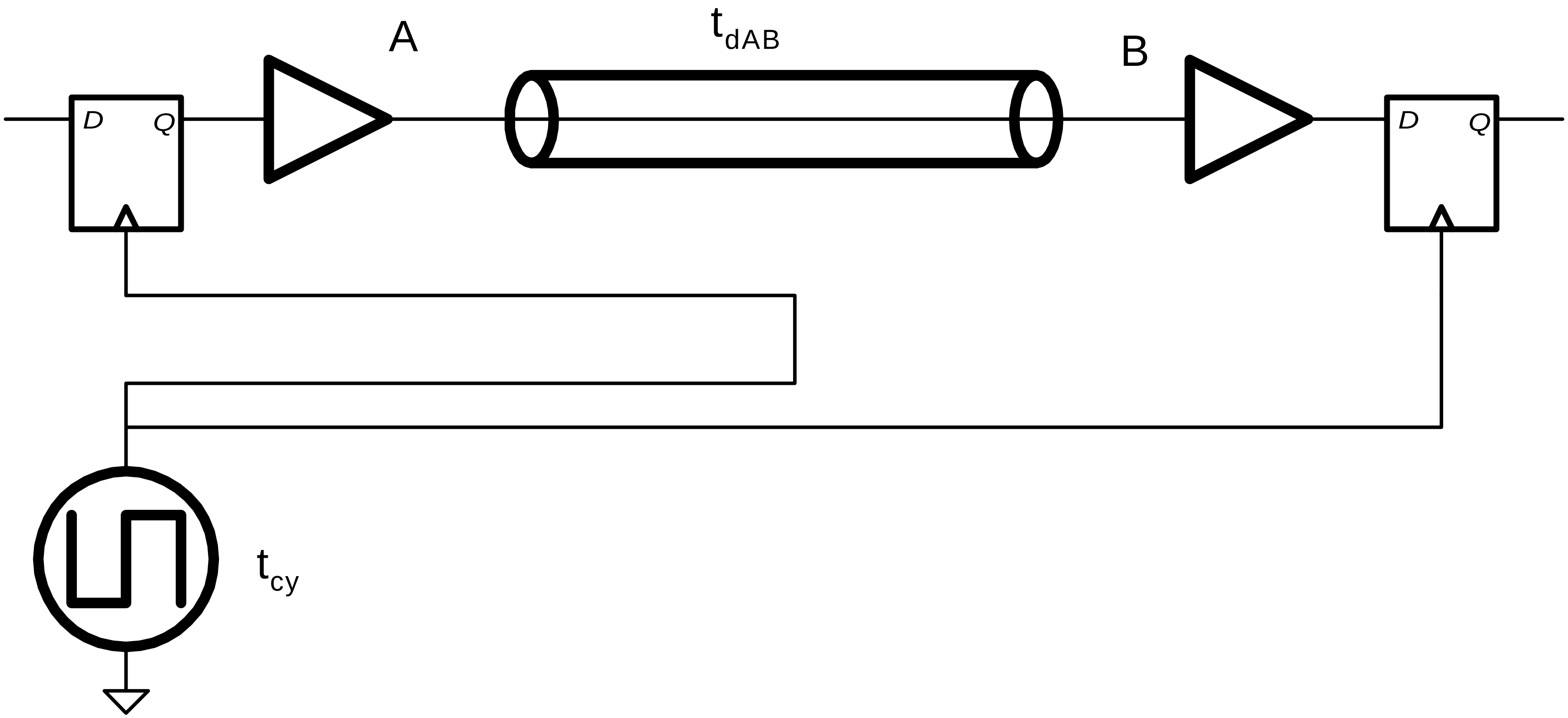
The Eye Diagram Revisited

- Operating rate is limited by 3 factors
  - transition (rise) time
  - aperture time
  - timing uncertainty, skew and jitter



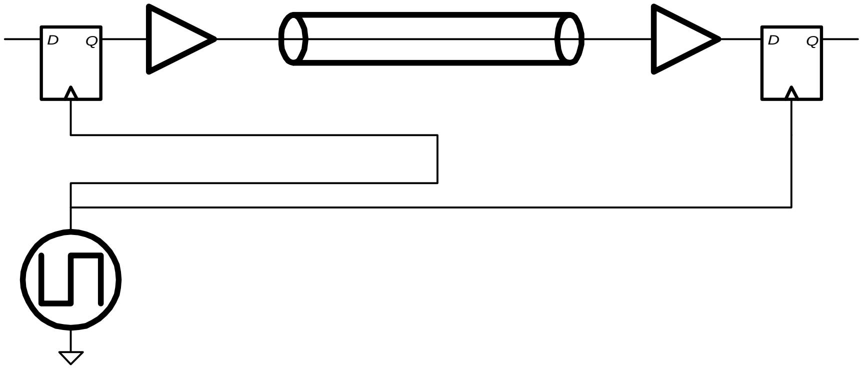
Synchronous Timing

- Consider a link using synchronous timing



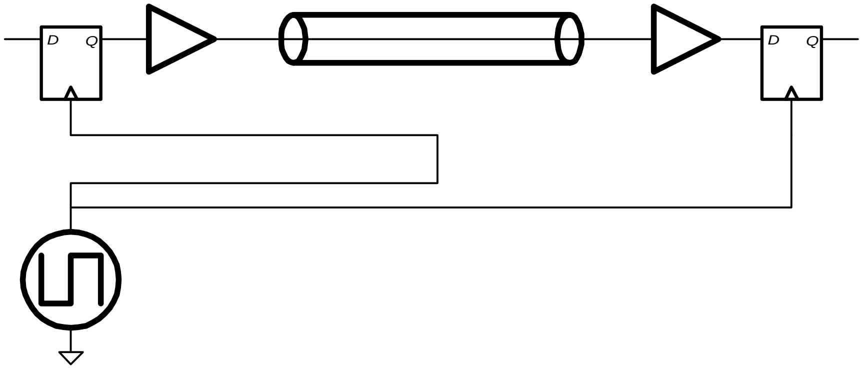
Synchronous Timing
Skew and Jitter

- Sources of uncertainty:
  - skew (across multiple lines) of line delay
  - jitter of Tx, Rx, and Line delay
  - skew and jitter of global clock
    - usually large because of high fanout
- For best performance want to *center* the sampling clock on the *eye*
- Line delay must be an odd multiple of half the clock!



Synchronous Timing Example

- Suppose line is 1m long
  - 6ns with Tx and Rx delay
- Rise time is 100ps
- Aperture time is 50ps
- Skew+Jitter is 100ps (p-p)
- How fast can the link go?
- What frequencies can it operate at?
  - at 4Gb/s, 24 bits on the wire
  - must slow to 23.5 bits on the wire (3.92Gb/s)
  - t_{cy} = 255ps ± 200fs
  - next window is at 22.5 bits (3.75Gb/s)

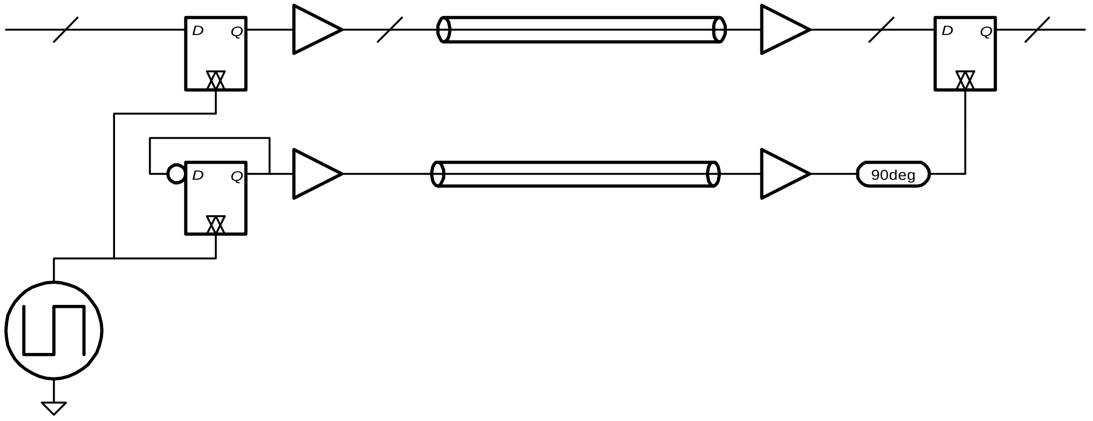


The problem with Synchronous Timing

- At high speeds and long lines it only operates over very narrow ranges of clock frequency
  – impractical to control delays and frequencies to precision required
  – unable to switch frequency of operation
    - can’t margin test system

Pipeline Timing

- Delay the clock by the same amount as the data *plus* half a bit cell
- System will work from DC to maximum frequency $1/(t_r+t_a+t_u)$
- Defines a new clock domain at the far end of the line

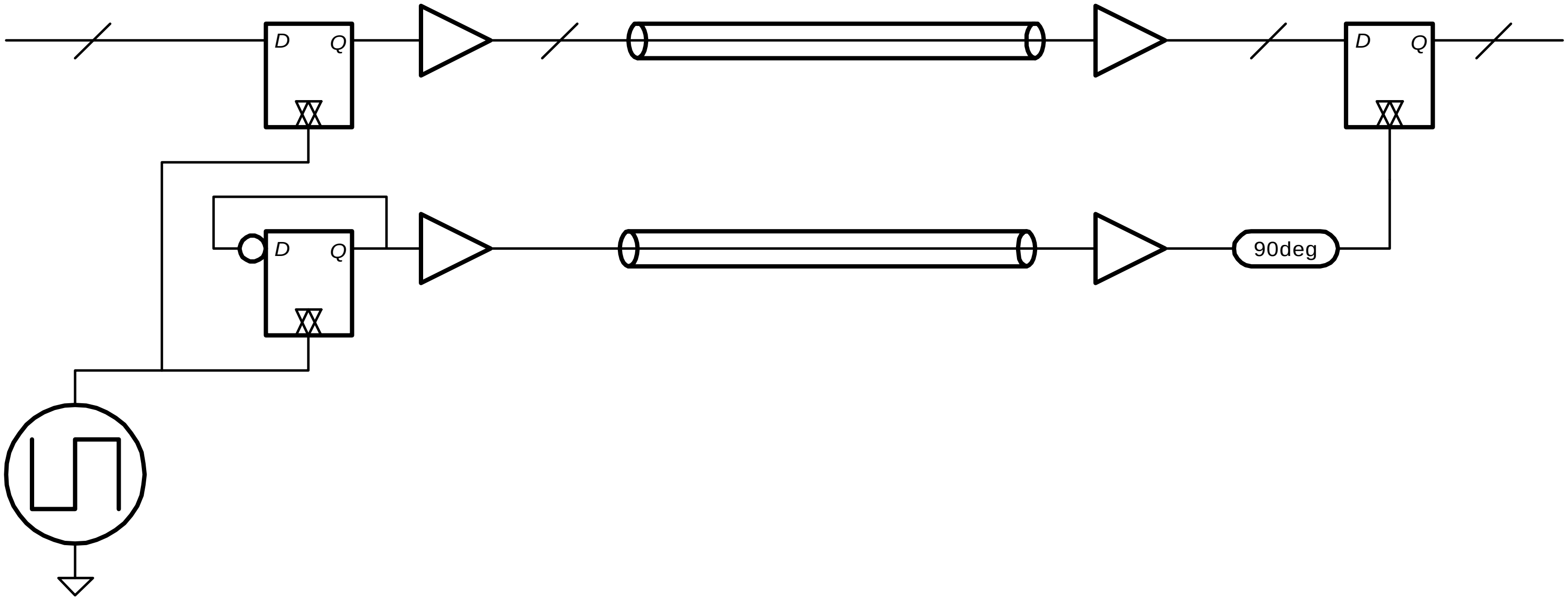


Pipeline Timing Example

t_r=100ps, t_a=50ps

t_j=10% of active delay, t_s=10% of active delay, 1% of wire delay

t_{dCQ}=200ps, t_{buf}=200ps, t_{clk_buf}=600ps



Next Time

- Closed-loop timing