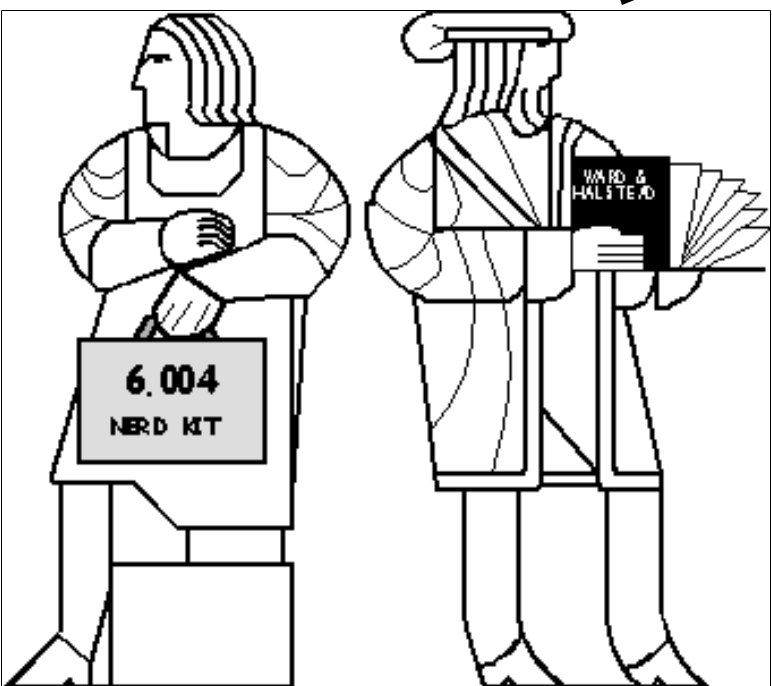
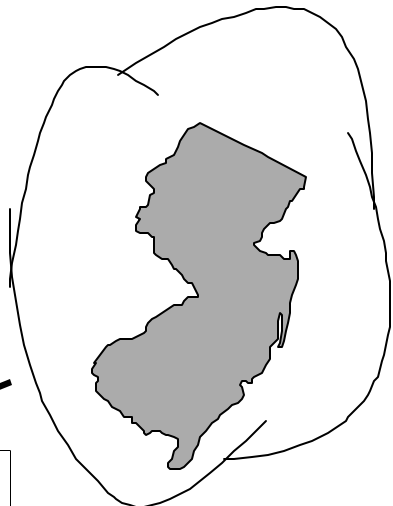
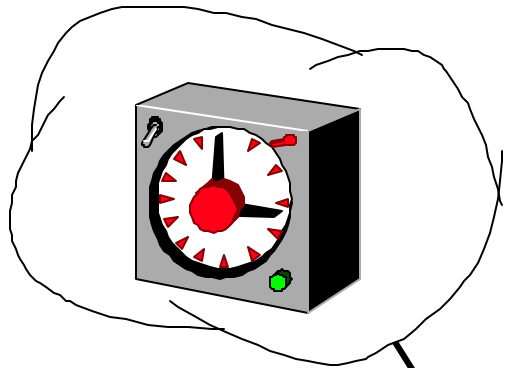
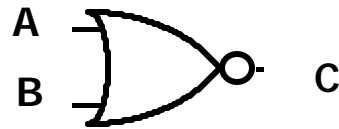


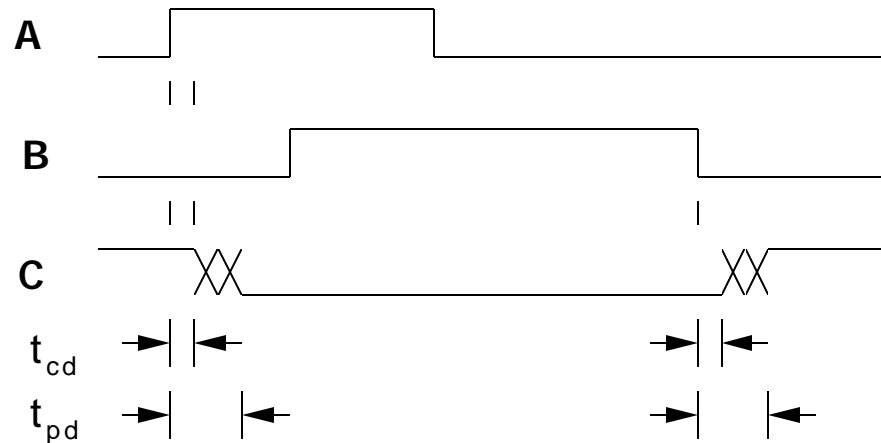
Timing & State



Combinational Timing Recap



A	B	C
0	0	1
0	1	0
1	0	0
1	1	0



Propagation Delay:

UPPER bound on interval between *valid* inputs and *valid* outputs.

Contamination Delay:

LOWER bound on interval between *invalid* inputs and *invalid* outputs.

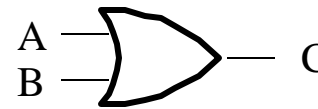
Always: $t_{cd} < t_{pd}$

Usually: t_{cd} assumed 0.

Combinational Hazards...

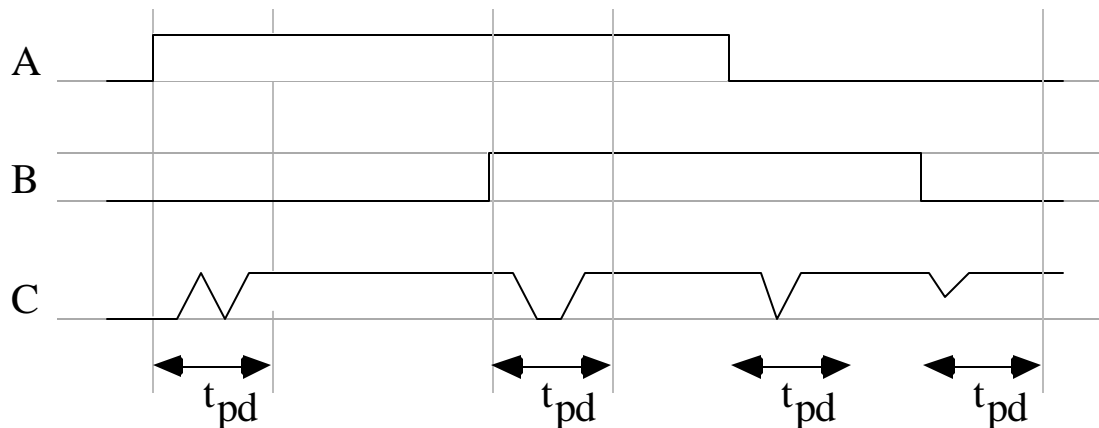
Note that combinational specification allows ANYTHING on outputs during t_{pd} ...

Consider a simple 2-input OR gate:



A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Can an OR gate get away with THIS...



HAZARDS:

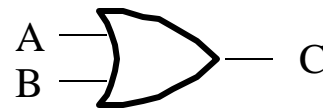
Extraneous output transitions from single input transitions

... and still be a valid combinational device?

Glitch-free Gate Criteria

We might raise our standards...

HAZARD-FREE OR GATE:

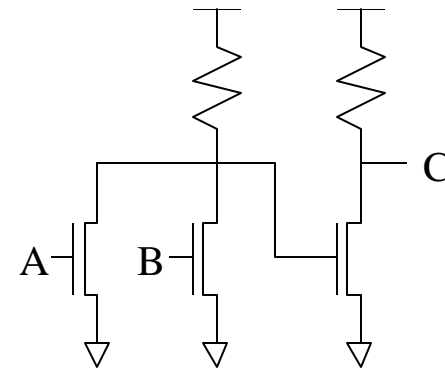


If either input is a valid 1 for at least t_{pd} , the output of the OR gate must be a valid 1.

A	B	C
0	0	0
1	X	1
X	1	1

Alternative truth table: X can be ANY input, including invalid.

Can we make 'em???
You Bet:



yet another abstraction ...

Recall the rules re *combinational devices*:

Combinational Device: Output guaranteed to be valid when all inputs have been valid for at least t_{pd} . *Any invalid input contaminates output!*

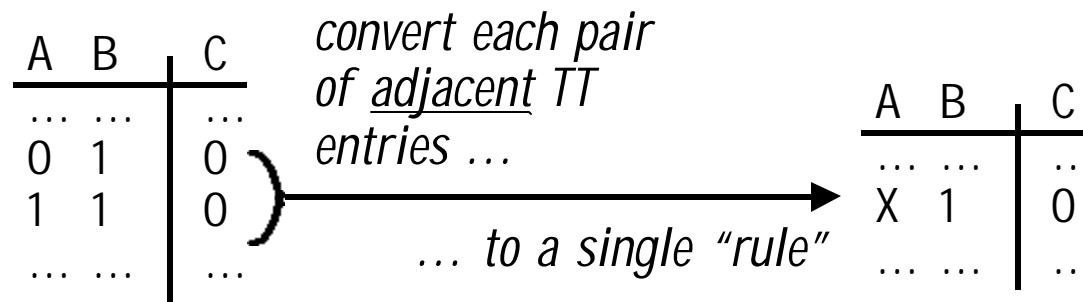
Among these, we distinguish *lenient* combinational devices:

LENIENT Combinational Device: Output guaranteed to be valid when any combination of inputs sufficient to determine output value has been valid for at least t_{pd} . *Tolerates transitions -- and invalid levels -- on irrelevant inputs!*

Observation: A *LENIENT* combinational device is hazard-free.

Lenient device specification

- 1: Introduce variables ("don't-cares") into the left side of the truth table. Repeatedly REDUCE truth table by combining lines with same output and differing in a single input:

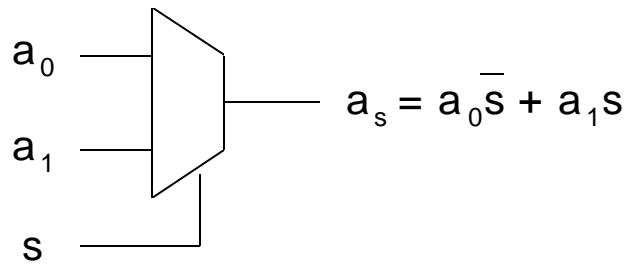


Compare:
Karnaugh Map
simplification

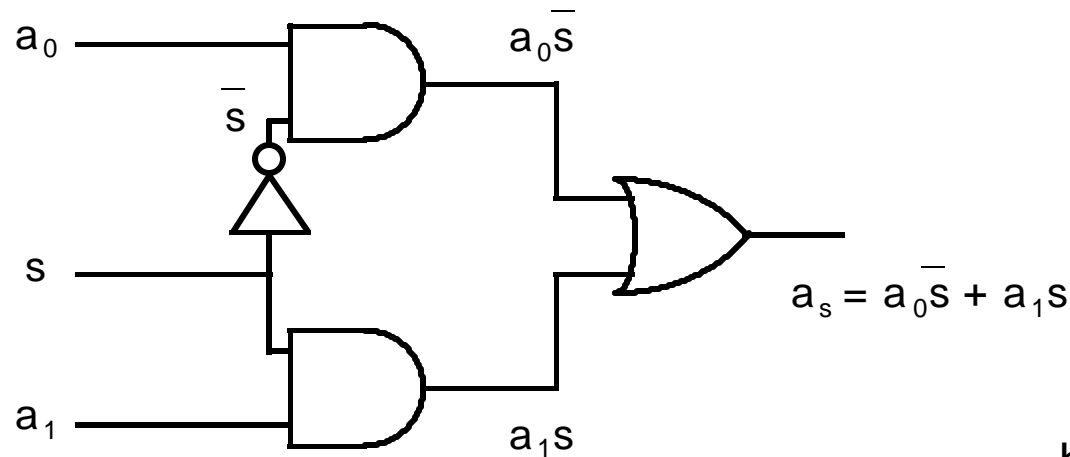
- 2: Interpretation of reduced truth table: If any input criterion is satisfied for tpd, specified output must be valid.

Question: Is the use of lenient components in an acyclic circuit sufficient to guarantee hazard-free operation?

Hazards & Circuits



2-input Multiplexor



Karnaugh Map

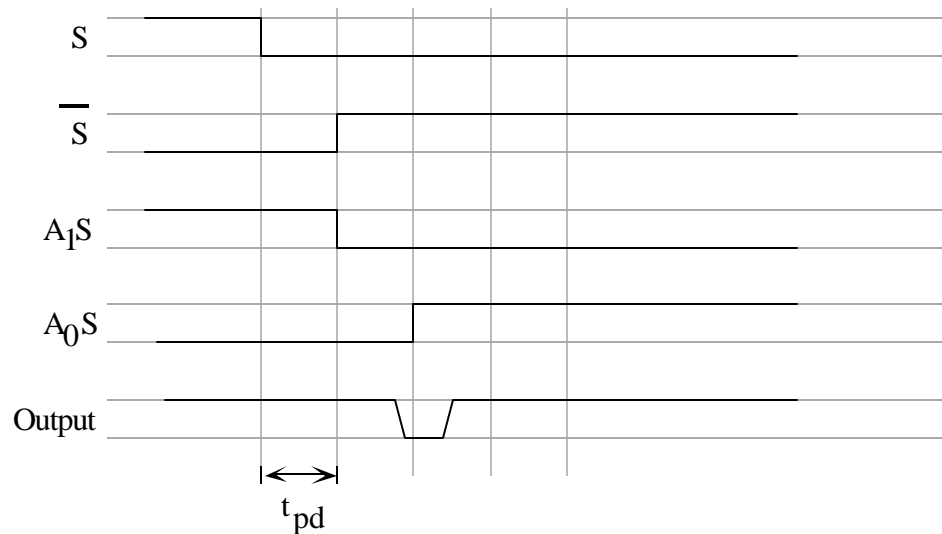
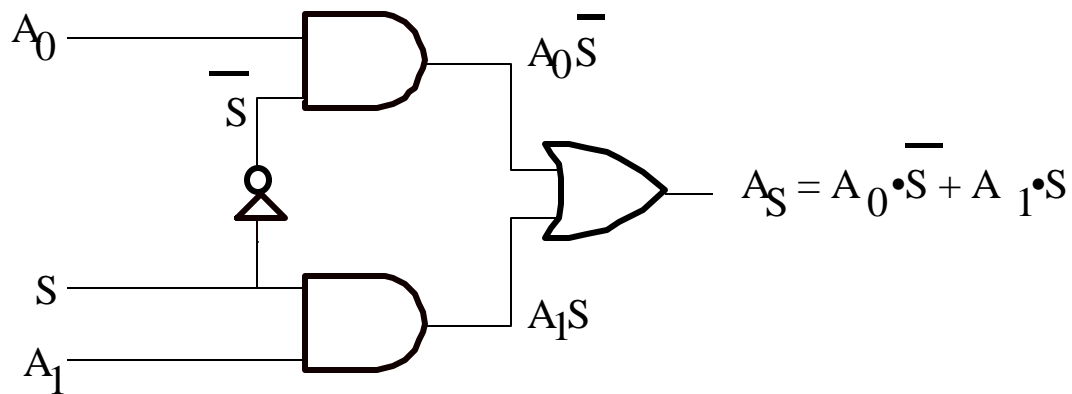
		$a_0 a_1$			
		00	01	11	10
s	0	0	0	1	1
	1	0	1	1	0

Try experiment:

hold $a_0 = a_1 = 1$, wiggle s .
What appears at output?

Combinational Composition Revisited

... a hazardous exercise



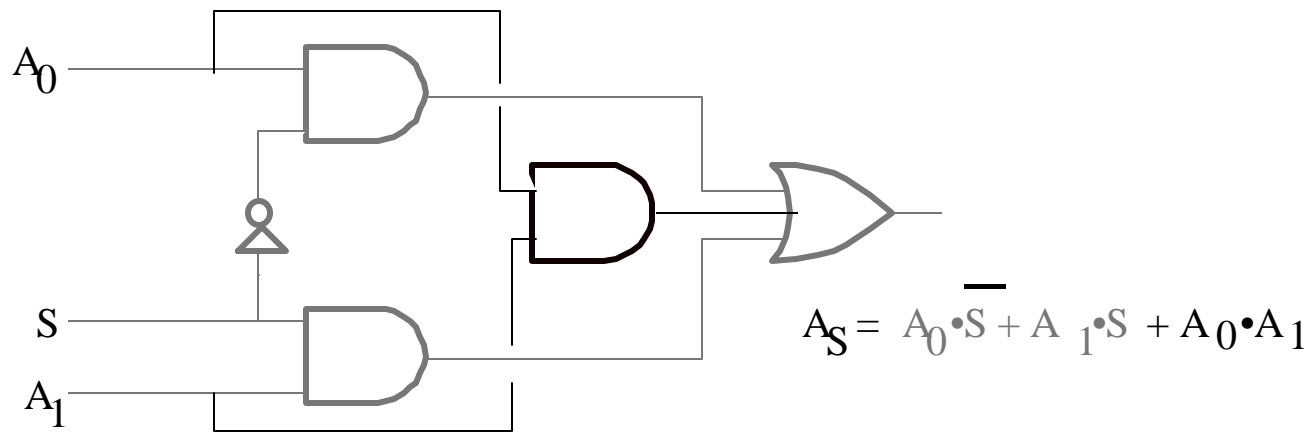
Problem: output dependent on

along two paths through circuit.

		$a_0 a_1$			
		00	01	11	10
s	0	0	0	1	1
	1	0	1	1	0

Hazard-Free Combinational Circuits

We can, however, choose a lenient (hence hazard free) implementation (assuming lenient components):



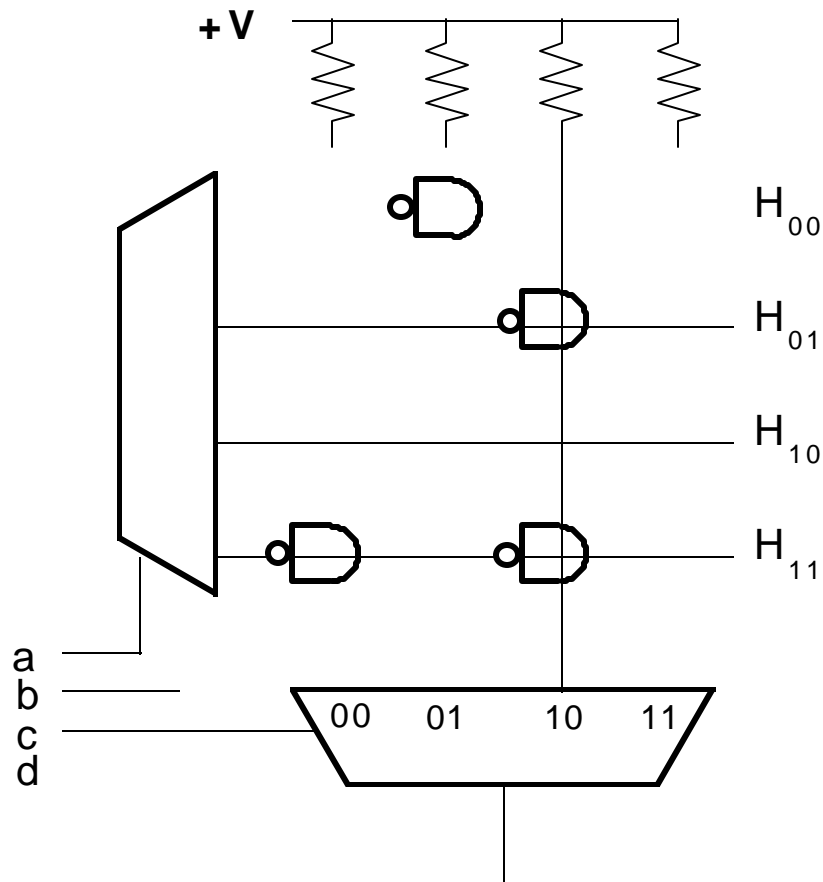
		$a_0 a_1$			
		00	01	11	10
s	0	0	0	1	1
	1	0	1	1	0

GENERAL TECHNIQUE: include EVERY maximal patch -- "prime implicant" -- in sum-of-products implementation.

... cf Truth Table "rules" for output=1.

ROM Hazards

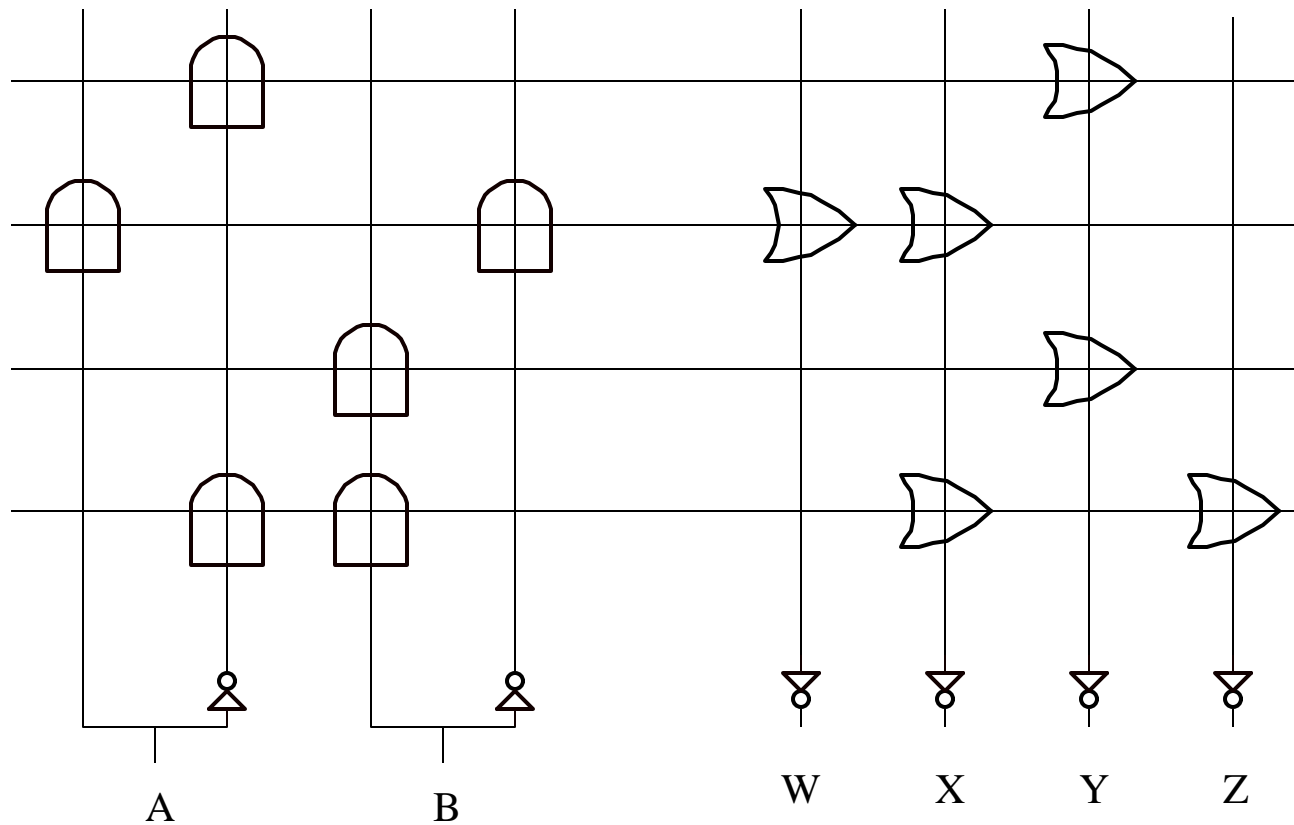
Recall our ROM sketch:



In general:

ROMs _____
hazard free.

PLA Hazards...



In general, PLAs _____ hazard free.

Hazards in perspective...

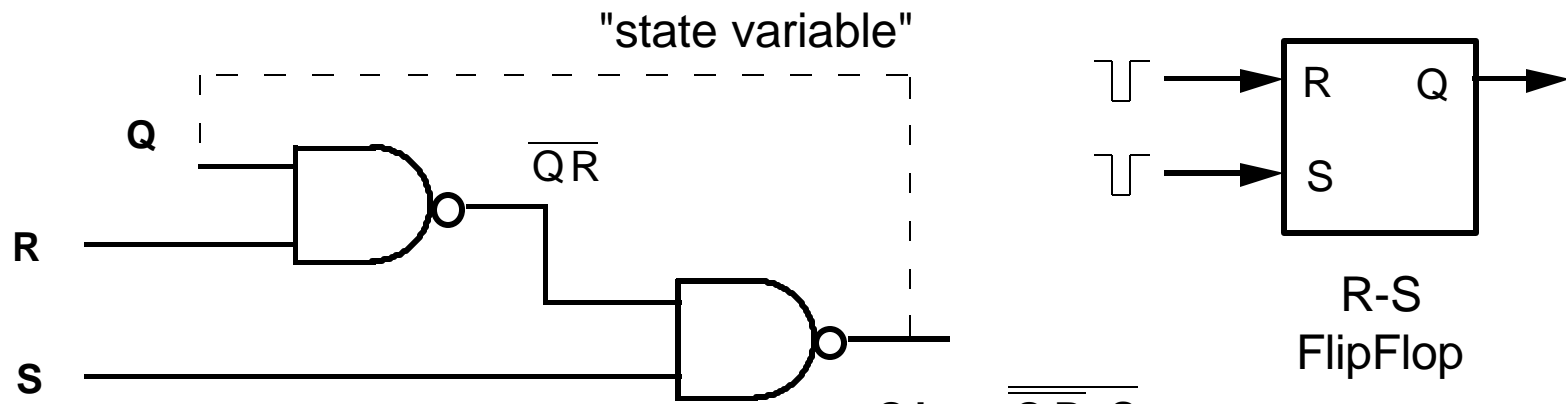
Q: When should we worry about hazards?

- A. Whenever we build combinational circuits.
- B. When driving in Cambridge
- C. On 6.004 quizzes
- D. Only in special circumstances, eg the rest of this lecture.

A. _____

Common digital engineering approaches do not depend on combinational outputs during propagation, hence tolerate combinational hazards.

Bistable (memory) Device



$$Q' = \overline{\overline{QR} \cdot S}$$

$$= QR + \overline{S}$$

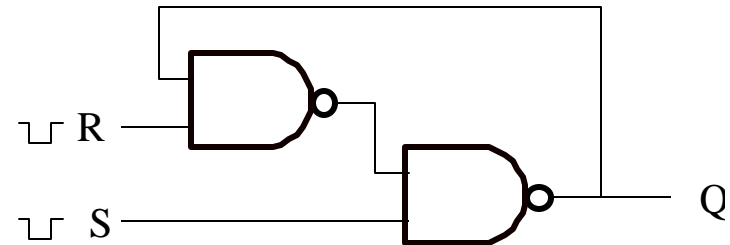
$$Q = QR + \overline{S} \dots$$

for $R=S=1$, 2 Solutions:
 $Q=0$ and $Q=1$.

R	S	Q'	
1	1	Q	holds
0	1	0	resets
1	0	1	sets
0	0	1	set overrides reset

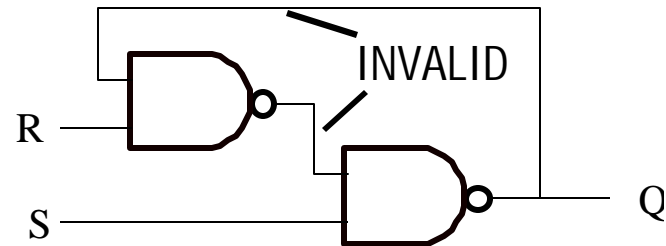
Reality Check...

Suppose these gates are NOT lenient. Do we have any guarantees on Q?



Certainly _____!

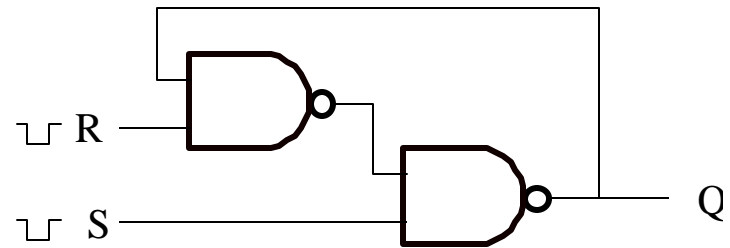
NOTE THAT we have no validity guarantees on Q unless we know history.... might have circulating invalid voltages!



Making Reliable 'Flops

So lets use LENIENT gates to make our R-S flip flop...

Can we FORCE a $Q=0$ or $Q=1$ state?



Using lenient gates:

$R=0, S=1$ forces $Q=0$ after _____ gate delays.

To force a $Q=0$ STATE CHANGE

(stable after return to $R=1, S=1$)

we must hold $R=0, S=1$ for at least _____ gate delays.

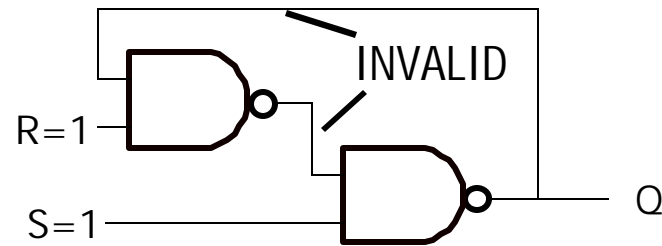
$R=1, S=0$ similarly forces $Q=1$.

Bistable Badness

Note that we can still have invalid state unless some input is asserted (low)... how to avoid?

Things that might cause problems:

“RUNT” (active low) pulses on R or S
Nearly simultaneous pulses on R and S

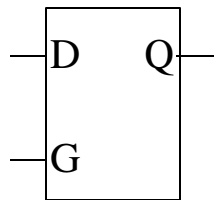


Need a DYNAMIC DISCIPLINE sufficient to guarantee validity...

... eg:

Force state changes only by non-overlapping R or S pulse at least $3 \cdot t_{pd}$ long

Bistable Latch

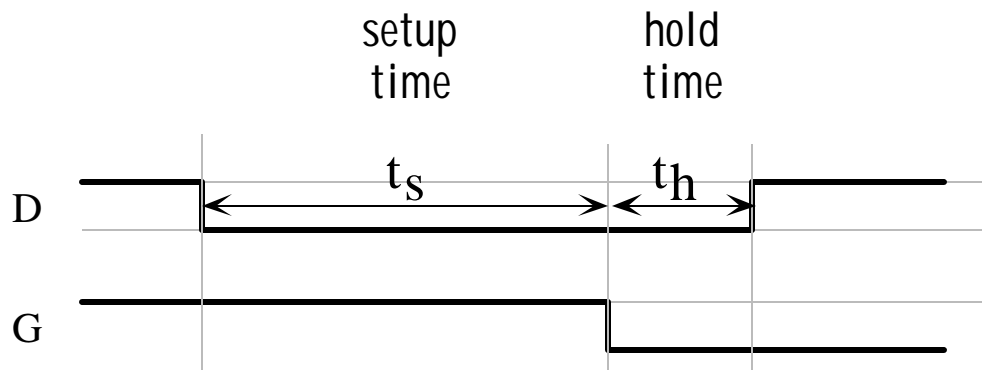
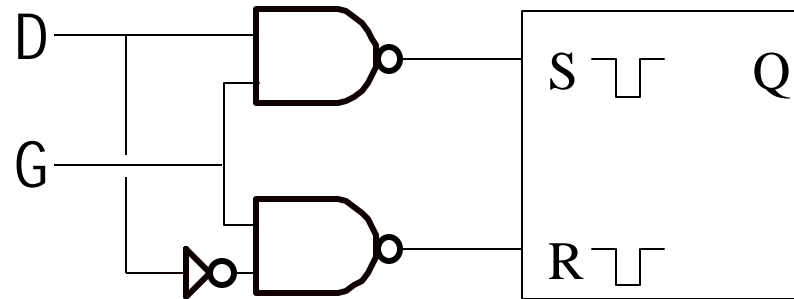


$G=0$: Holds Q constant

$G=1$: Q follows D

DYNAMIC DISCIPLINE (to guarantee R-S Flop specs are obeyed):

1. MINIMUM $G=1$ Pulse Width
2. Can't change D near $1 \rightarrow 0$ G transition



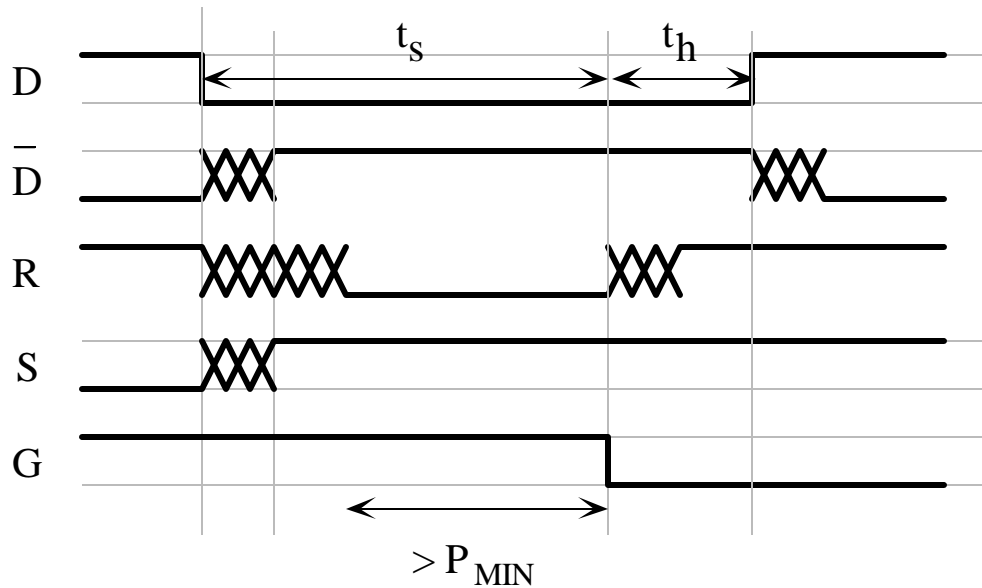
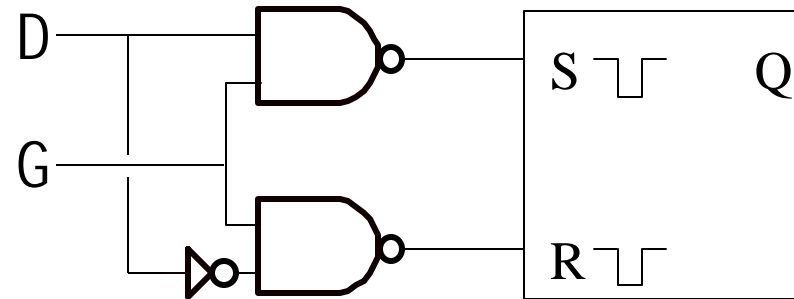
Setup and Hold time specs:

Dictate a "window" during which data must be stable surrounding a transition.

Latch Setup/Hold times

Constraint:

Setup time spec must be sufficient to GUARANTEE reset pulse width $> P_{MIN}$



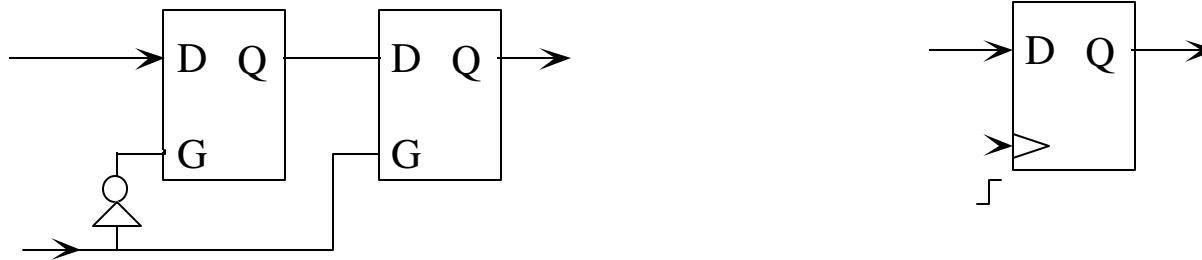
If gate $t_{pd}=1$, $t_{cd}=0$:

$$t_s = \underline{\hspace{2cm}}$$

$$t_h = \underline{\hspace{2cm}}$$

(to avoid S glitch)

Edge-triggered Flip Flops

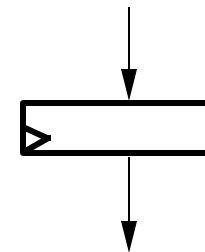


Positive edge triggered D flip flop

IDEA:

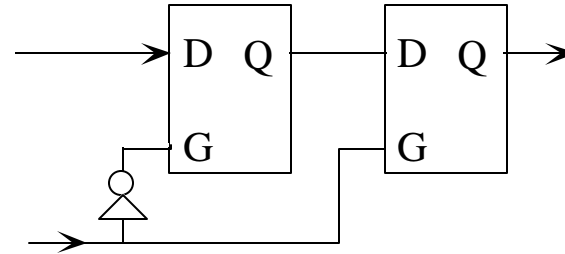
- Only one latch "transparent" at any time -- no combinational path through flip flop.
- Seems to "capture" value on D input at time of (positive) transition of clock input.

Clocked registers: FF's sharing clock input, hold multi-bit data.



Um, about that hold time...

Does this
implementation
actually work???

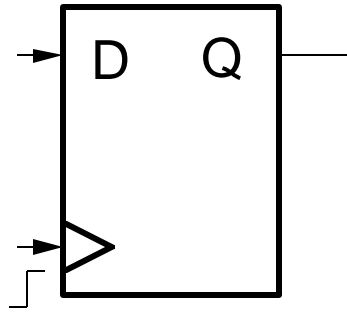


Consider HOLD TIME requirement for right-hand latch:

- Negative clock transition -> RH latch freezes data:
 - SHOULD be no output glitch, since LH latch held constant data; BUT
 - LH latch output contaminated by change in G input!
- HOLD TIME of RH latch not met, UNLESS we assume sufficient contamination delay in the path to its D input!

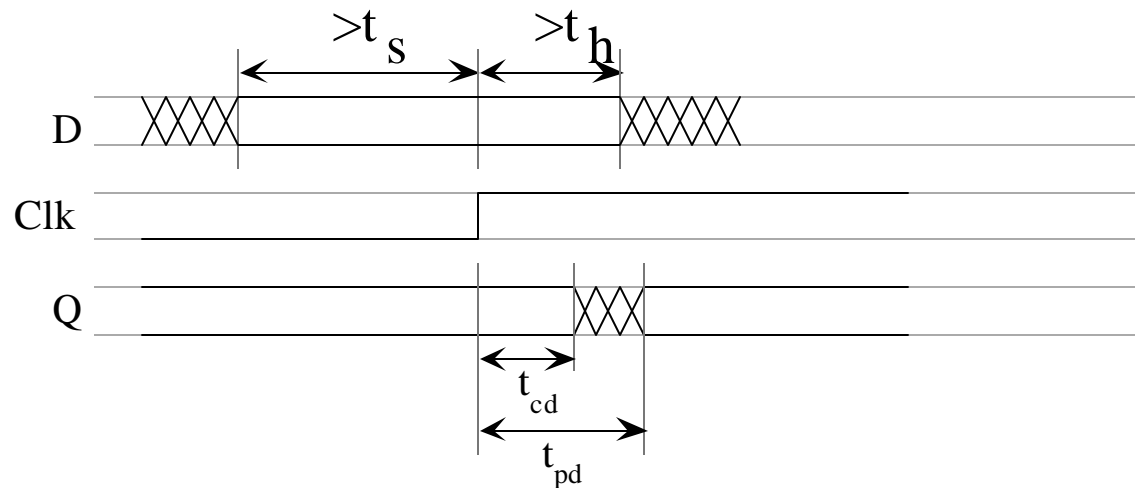
Accumulated t_{cd} thru inverter, G->Q path of latch must cover latch t_h for this design to work!

Timing of Clocked Devices



Timing specifications: setup, hold, propagation times relative to clock edge

MAY specify a clock-to-Q contamination delay.



NB: if timing doesn't conform to specs, ALL guarantees are void!

USUALLY $t_{cd} > t_h$ so that flops may be _____ .

The Dynamic Discipline

Engineering goal: Build systems GUARANTEED to work (meet spec) assuming each component works.

Behavior of clocked devices (e.g. Flops, Registers) guaranteed ONLY so long as setup, hold times are met...

The Need for a Dynamic Discipline:

Design of sequential circuits MUST guarantee that inputs to sequential devices are valid and stable during periods when they may influence state changes.

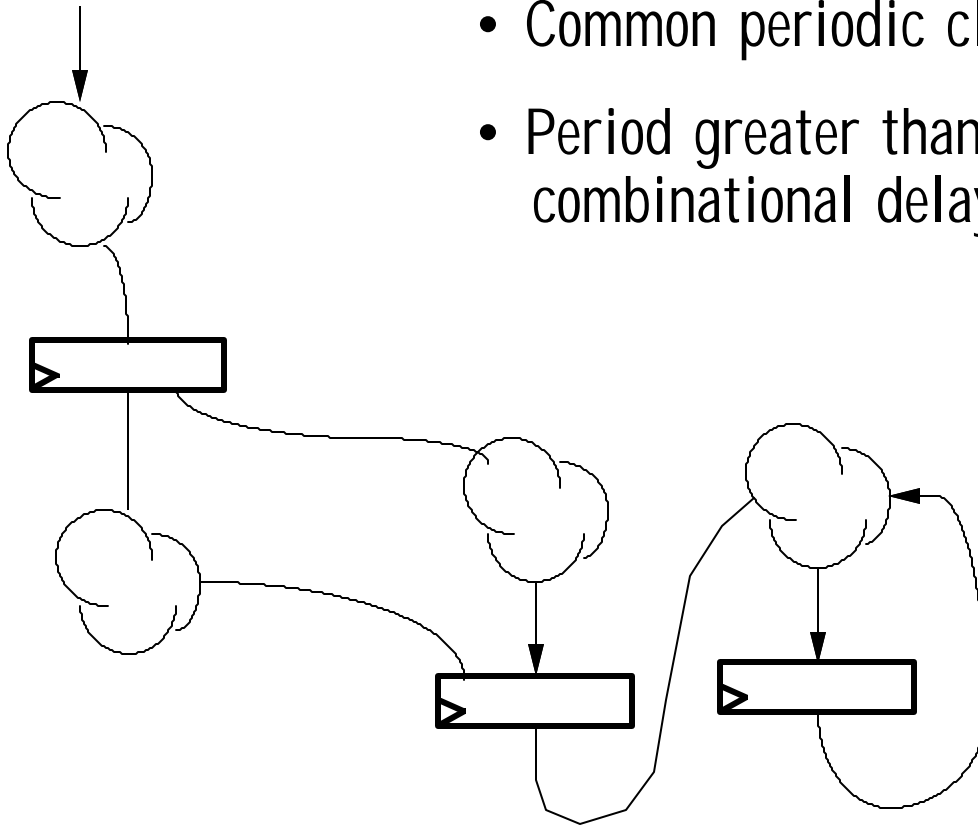
Asynchronous sequential devices: minimum pulse widths, complex rules for dealing with overlapping control signals & edge signalling.

Clocked devices: obey setup, hold times, max clock frequency.

NEED: A simple engineering discipline to follow...

Synchronous, 1-clock Discipline

- NO combinational cycles
- Common periodic clock
- Period greater than every combinational delay



Thursday: Finite State Machines

Finite State?
Where'd I read that?

Must be a
Rhode Island
license plate.

