EECS 42 Introduction Digital Electronics
Andrew R. Neureuther

Lecture # 21 Clock Operation of Latches

Handout of This Lecture.

A) 2nd Midterm Returned
B) CMOS Propagation Delays
C) Latch circuit to hold/release signals
D) Cascade CMOS elements with latches

http://inst.EECS.Berkeley.EDU/~ee42/

Results Midterm #2

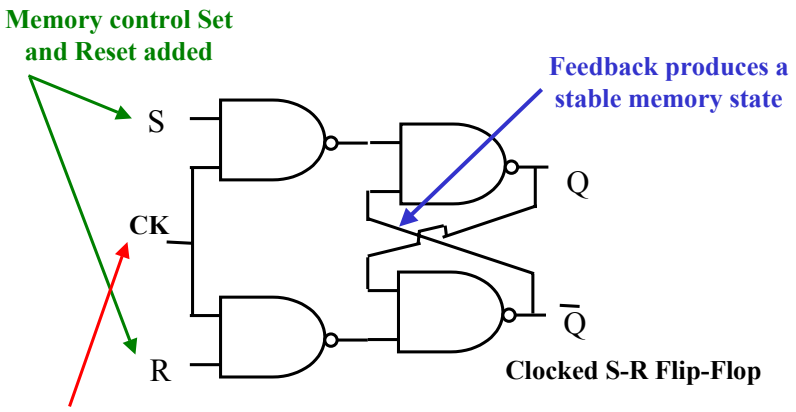
	P1	P2	P3	P4	Tot
	25	25	28	22	100
Ave	22.4	15	18.8	14.3	**70.4**
Ave/Max	0.90	0.60	0.67	0.65	0.70
StDev	4.9	8.6	7.7	6.3	**20.1**
StDev/Max	0.20	0.35	0.28	0.29	0.20

High 100 (2), Low 12, Median 74

Approximate Scale:

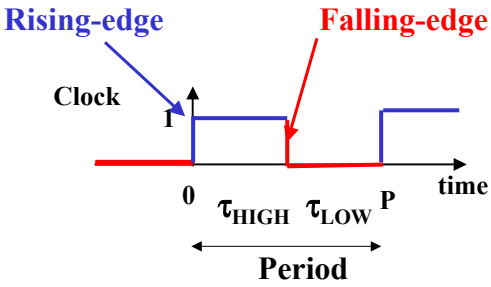
A+ = 97, A = 91, A- = 86, B+ = 81, B = 76, B- = 70,
C+ = 64, C = 57, c- = 50, D+ = 43, D = 36, D- =30

Memory Element: Cross-Coupled Gates



In HW #11 you will do a timing diagram to show that the clock prevents 1's catching

Clock Signal Definitions



Period =P = $\tau_{HIGH} + \tau_{LOW}$

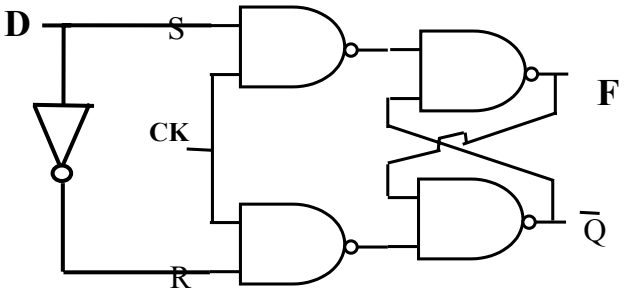
Frequency = 1/P = 1/($\tau_{HIGH} + \tau_{LOW}$)

Duty Cycle = (τ_{HIGH})/($\tau_{HIGH} + \tau_{LOW}$)

D-Type Flip Flop

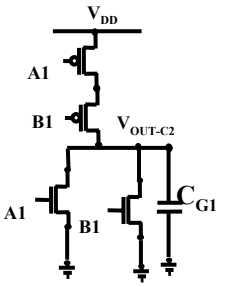
D stands for Delay

This circuit retains (delays and holds) the old input signal until the clock goes high

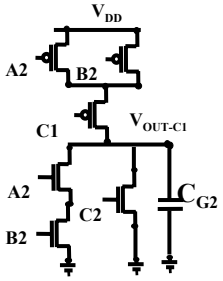


This circuit is very frequently used in applications such as latches and shift registers.

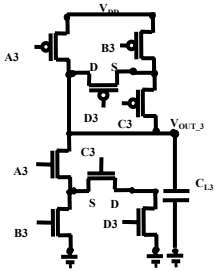
Logic Functions



Gate 1



Gate 2

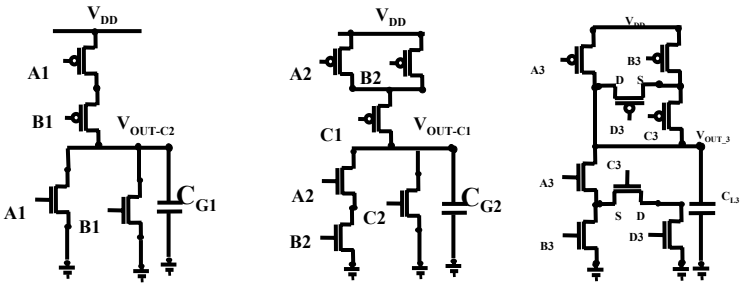


Gate 3

F1 = $\overline{(A1 + B1)}$ **F2 = $\overline{((A2 * B2) + C2)}$**

F3 = $\overline{(A3 * (B3 + C3 * D3))}$

Logic Worst Case Delays



Gate 1

$\tau_{LH} = 2\tau_{INV}$

Gate 2

$\tau_{HL} = 2\tau_{INV}$

$\tau_{LH} = 2\tau_{INV}$

Gate 3

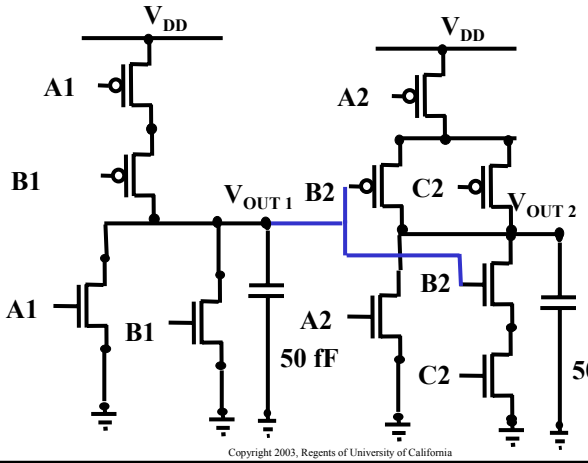
$\tau_{HL} = 3\tau_{INV}$

$\tau_{LH} = 3\tau_{INV}$

EECS 42 Intro. Digital Electronics Fall 2003 Lecture 21: 11/13/03 A.R. Neureuther
Version Date 11/12/03

Logic Gate Cascade

To avoid large resistance due to many gates in series, logic functions with 4 or more inputs are usually made from cascading two or more 2-4 input blocks.



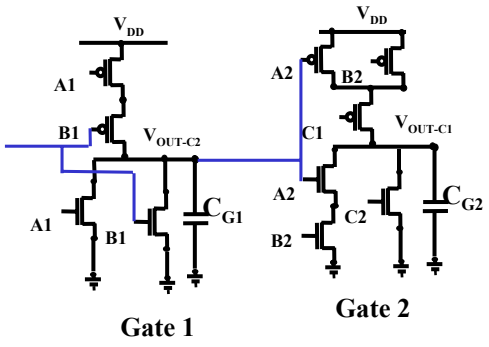
$B2 = V_{OUT\ 1}$

The four independent input are A1, B1, A2 and C2.

A2 high discharges gate 2 without even waiting for the output of gate 1.

C2 high and A2 low makes gate 2 wait for Gate 1 output

Propagation Delays Add in Cascade

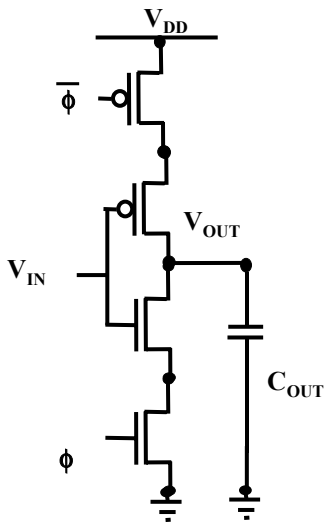


$$\tau_{PD\_CASCADE} = \tau_{PD\_1} + \tau_{PD\_2}$$

Data Synchronization problem

- Combinatorial logic gates can give incorrect answers prematurely and may take several gate propagation delays produce an answer.
- Clocks (signals as to when to proceed) and latches (which capture and hold the correct outputs) can provide synchronization.

Latch Controlled by a Clock

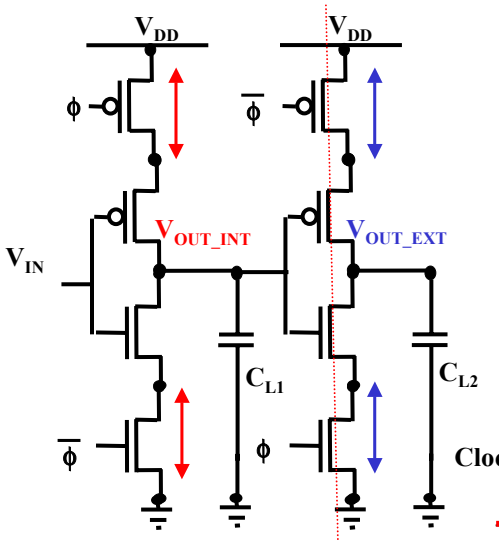


An inverter with clocked devices in series can form a latch.

When the clock ϕ is high its complement $\overline{\phi}$ is low and the inverter operates.

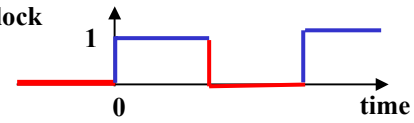
To synchronize the data the clock remains low until the data is correct at all locations on the chip. When the clock goes high the inverse of the data is passed.

Latch Work Best In Pairs

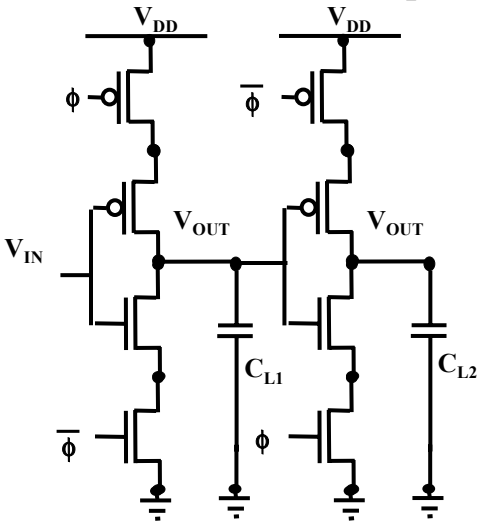


The first stage operates while the clock is low and inverts and amplifies the arriving signal and charges or discharges C_{L1}.

The second stage operates while the clock is high and inverts the signal on C_{L1} to charge or discharge C_{L2} and downstream logic gate inputs.



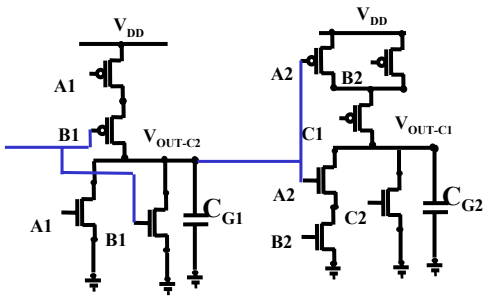
A Double Latch is an Edge-Triggered D Type Flip-Flop



During the low part of the clock cycle this circuit records the input value and when the clock goes high drives $V_{OUT\ 2}$ to the voltage level that arrived. (This is the classic function of a D flip-flop.)

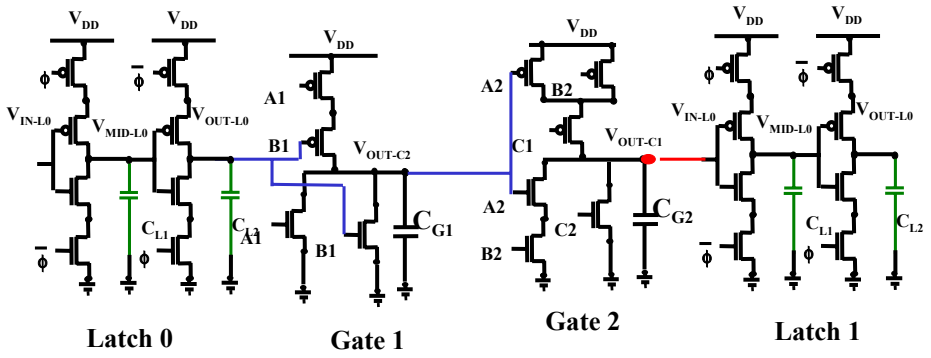
Note that this circuit is not fooled by noise on the input and makes its decision on the rising edge of the clock (edge-triggered).

Example of Circuits to Integrate with Latches



$$\tau_{PD\_CASCADE} = \tau_{PD\_1} + \tau_{PD\_2}$$

Latch Implementation: Lumped



Latch Operation: Lumped