EECS 42 Introduction to Electronics for Computer Science

Andrew R. Neureuther

Lecture # 21 Clock Operation of Latches

Handout of Monday Lecture.

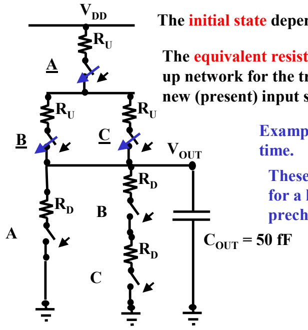
A) 2^{nd} Midterm Returned

B) Latch circuit to hold/release signals

C) Cascade CMOS elements with latches

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

Logic Gate Propagation Delay: Initial State



The initial state depends on the old (previous) inputs.

The equivalent resistance of the pull-down or pull-up network for the transient phase depends on the new (present) input state.

Example: A=0, B=0, C=0 for a long time.

These inputs provided a path to V_{DD} for a long time and the capacitor has precharged up to V_{DD} = 5V.

Game Plan 04/21/03

Last Week: Logic Delay; resistor model, CMOS operation and delay

Monday 4/14/03:

- ❑ 2^{nd} Midterm returned
- ❑ CMOS Latch and use in logic cascade

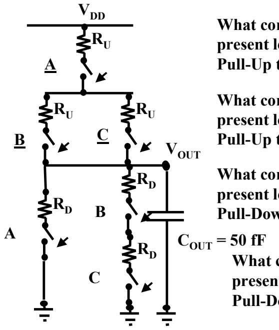
Wednesday 04/09/03:

- ❑ Clocked Latch and Timing Diagram
- ❑ Latency and Throughput
- ❑ Feedback in logic to produce memory

Next (13th) Week: Diodes and MOS Devices

Problem set #10 for 4/303: Logic Delay; Cascade; Latches and Clock frequency

Logic Gate: Worst Case Scenarios



What combination of previous and present logic inputs will make the Pull-Up the fastest?

What combination of previous and present logic inputs will make the Pull-Up the slowest?

What combination of previous and present logic inputs will make the Pull-Down the fastest?

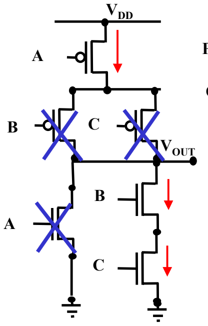
What combination of previous and present logic inputs will make the Pull-Down the slowest?

Fastest overall?

Slowest overall?

CMOS Logic Gate: Example Inputs

A = 0
B = 1
C = 1



PMOS A conducts; B and C Open

Output is High

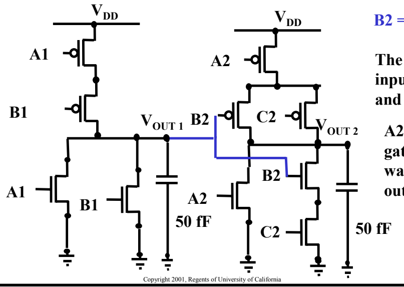
= 0

NMOS B and C conduct; A open

Logic is Complementary and produces F = 0

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.

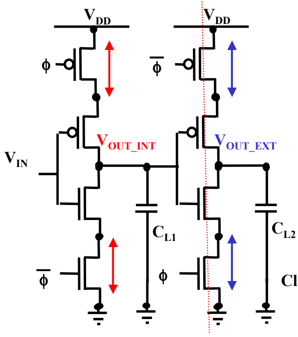
A2 high discharges gate 2 without even waiting for the output of gate 1.

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

Data Synchronization problem

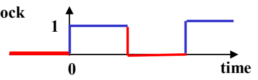
- Combinatorial logic gates can give incorrect answers prematurely and may take several gate propagation delays produce an answer.
- Clocks (signals as to when to proceed) and latches (which capture and hold the correct outputs) can provide synchronization.

Latch Work Best In Pairs

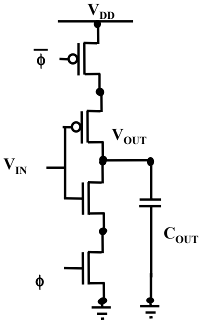


The first stage operates while the clock is low and inverts and amplifies the arriving signal and charges or discharges C_{L1}.

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.



Latch Controlled by a Clock

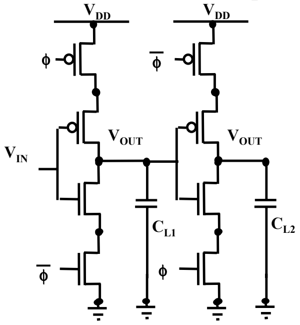


An inverter with clocked devices in series can form a latch.

When the clock ϕ is high its complement $\bar{\phi}$ 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.

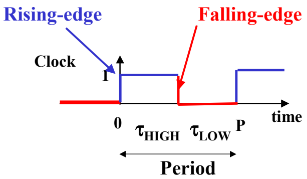
A Double Latch is an Edge-Triggered D Type Flip-Flop



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).

Clock Signal Definitions

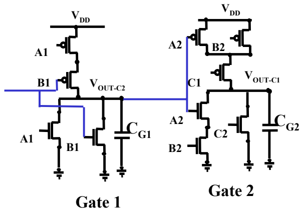


Period =P = $\tau_{HIGH} + \tau_{LOW}$

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

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

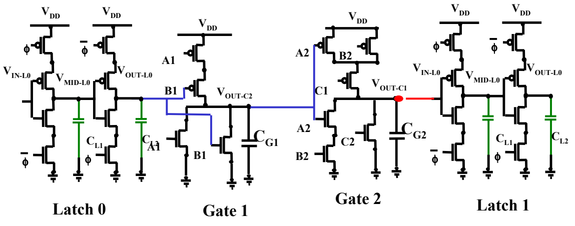
Example of Circuits to Integrate with Latches



EECS 42 Intro. electronics for CS Spring 2003 Lecture 21: 04/21/03 A.R. Neureuther

Version Date 04/19/03

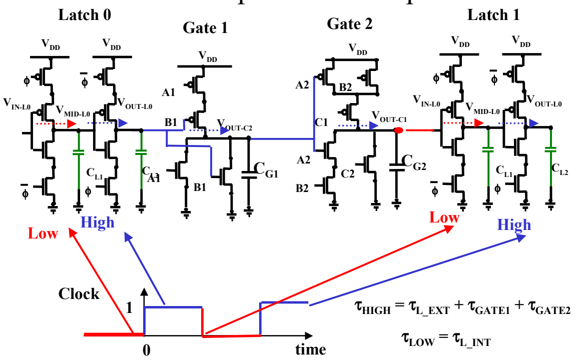
Latch Implementation: Lumped



EECS 42 Intro. electronics for CS Spring 2003 Lecture 21: 04/21/03 A.R. Neureuther

Version Date 04/19/03

Latch Operation: Lumped



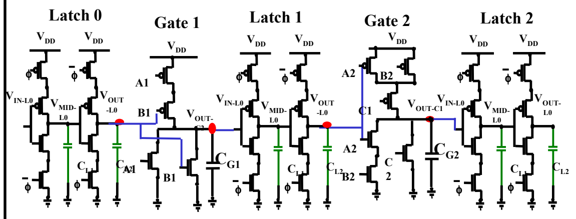
$\tau_{HIGH} = \tau_{L\_EXT} + \tau_{GATE1} + \tau_{GATE2}$

$\tau_{LOW} = \tau_{L\_INT}$

EECS 42 Intro. electronics for CS Spring 2003 Lecture 21: 04/21/03 A.R. Neureuther

Version Date 04/19/03

Latch Implementation: Pipelined


Latch 0
Gate 1
Latch 1
Gate 2
Latch 2
V_DD
φ
V_IN-L0
V_MID-L0
V_OUT-L0
C_L1
A1
B1
V_OUT-C2
C_G1
C_L2
A2
B2
C1
C2
V_OUT-C1
C_G2