11/18/03

EE 42 – Introduction to Electronics for Computer Science

Fall 2003, **Prof. A. R. Neureuther**
Dept. EECS, 510 Cory **neureuth@eecs.berkeley.edu** **642-4590**
UC Berkeley **Office Hours M1, Tu, Th 10:30-11:30, F 11**
Course Web Site **http://www-inst.eecs.berkeley.edu/~ee42/**

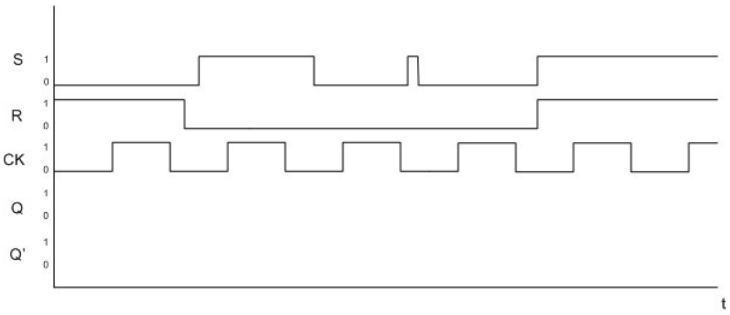
Problem Set # 11

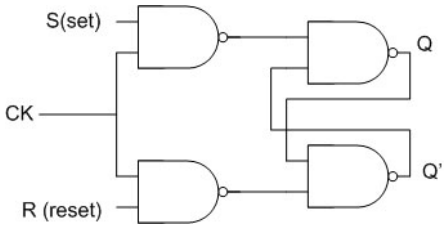
Due:1 PM Nov 26, 2003 in box outside 240 Cory

Throughout this problem, make the simplifying assumption that each gate or latch stage has a load of 50fF and that each 'on' transistor has a resistance of 10k. That is $R_N = R_P = 10k$. The inverter delay is 0.69(10k)50fF = 345pS.

11.1 Transient

Fill out the timing diagram for the given clocked S-R Flip Flop:

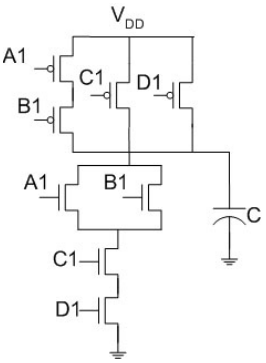




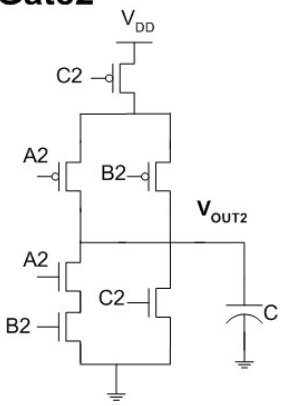
11.2 Circuit Extraction and Delay

(a) Determine the logic function of Gate1 and Gate2.

Gate1



Gate2



(b) Find the minimum and maximum propagation delay t_{HL} for the output going from high to low for each of the gates. Give the data dependent range of delays compared to a simple inverter (such as ½ to 4/3 the inverter delay)

(c) Find the minimum and maximum propagation delay t_{LH} for the output going from low to high for each of the gates. Give the data dependent range of delays compared to a simple inverter (such as ½ to 4/3 the inverter delay)

(d) Give the worst case overall range of data dependent delays.

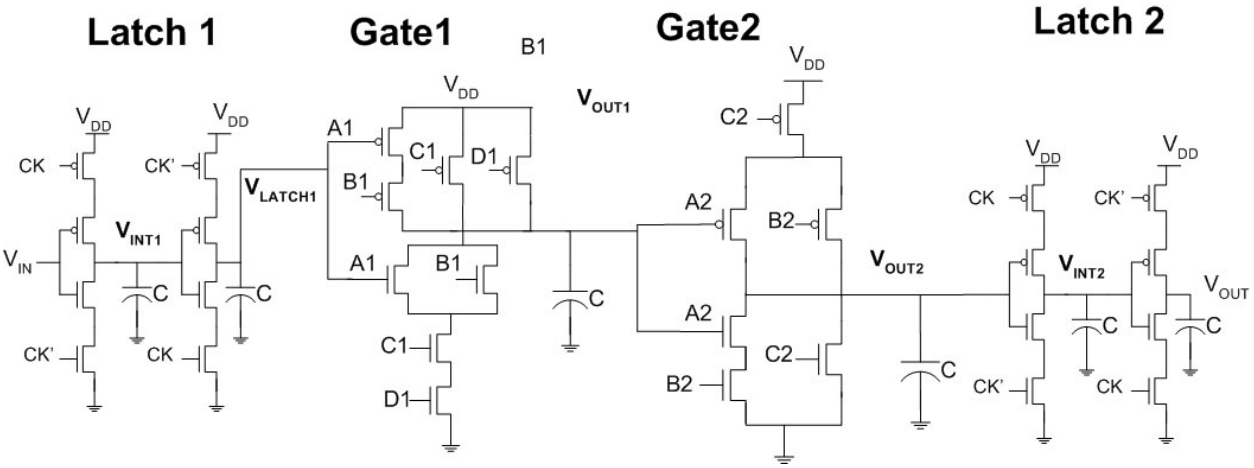
(e) Determine the shortest propagation delay for the output of Gate2 to change compared to an inverter and give a set of initial inputs and input changes that produce it.

(f) Determine the longest propagation delay for the output of Gate2 to change compared to an inverter and give a set of initial inputs and input changes that produce it.

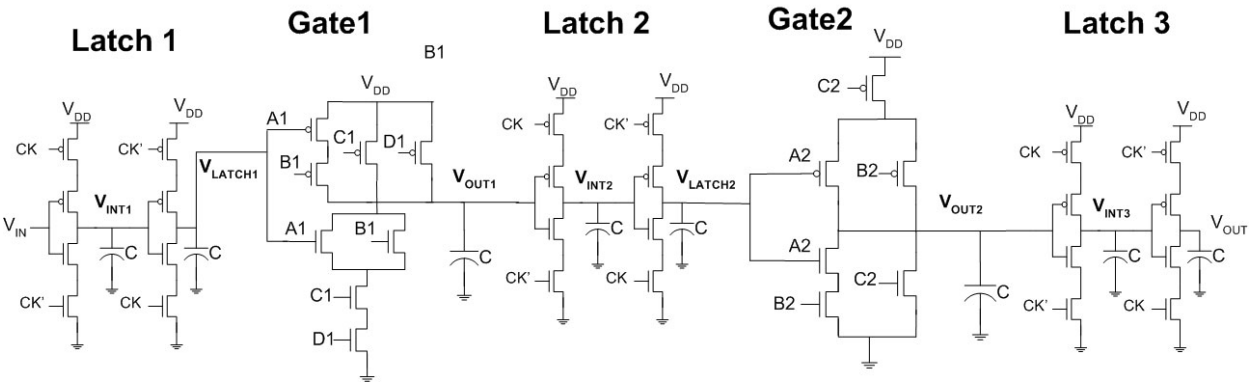
11.3 Lumped Logic vs. Pipelined Logic. Latches are now introduced to synchronize transfer of logic outputs among circuits as shown below. There are two implementations. In the lumped implementation, the cascade of Gate1/Gate2 form the combination logic that sit between two latches, L1 and L2. In the pipelined implementation, a latch L1 sits behind Gate1, followed by another latch L2, followed by Gate2, then followed by latch L3.

Assume the signals have been $V_{INT1} = 1$, $V_{LATCH1} = A1 = 1$, $B1 = 1$, $C1 = 1$, $D1 = 1$, $A2 = V_{OUT1}$, $B2 = 0$, $C2 = 0$ for a long time. However, at time $t = 0$, the following signals instantaneously switch: $B1 = 0$, $B2 = 1$. Assume that the clock goes from negative edge to positive edge at $t = 0$.
(a) Determine the propagation delay of the latch circuit for both rising edge and falling edge of the clock in units of inverter delay.
(b) Sketch V_{LATCH1}, V_{OUT1}, V_{INT2}, V_{OUT2}, and V_{OUT} for **two** complete clock cycles (assume clock goes up for 10 inverter delays and goes down for 10 inverter delays). Mark any delays on the timing diagram in terms of inverter delays.
(c) Sketch V_{LATCH1}, V_{OUT1}, V_{INT2}, V_{OUT2}, V_{LATCH2}, V_{OUT2}, V_{INT3}, and V_{OUT} and for **three** complete clock cycles (assume clock goes up for 10 inverter delays and goes down for 10 inverter delays). Mark any delays on the timing diagram in terms of inverter delays.

Lumped Logic



Pipelined Logic



10.4 Latency and Throughput of Lumped and Pipelined Logic.
(a) Determine the minimum amount of time the clock must stay high right after a positive edge in order for the data to propagate from latch 1 to the input of latch 2. Determine the minimum amount of time the clock must stay low right after a negative edge in order for the data transfer to be complete to Latch 2.
(b) Repeat part a for data propagating from Latch 1 to Latch 2 and from Latch 2 to Latch 3.
(c) The latency of the overall logic evaluation is the time for the data to appear on the final output latch. Find the latency for the lumped and pipeline implementations in terms of a number of inverter delays. (Be sure to use the output of Latch 3 for the pipeline).
(d) The minimum clock period must be larger than the latch-to-latch delay. The maximum clock frequency is the inverse of the minimum clock period. Find the minimum clock period for the lumped and pipelined implementations. The clock need not go low for as long as it goes high. Assuming one computed result comes out per clock cycle, how much faster is the pipeline architecture than the lumped architecture?