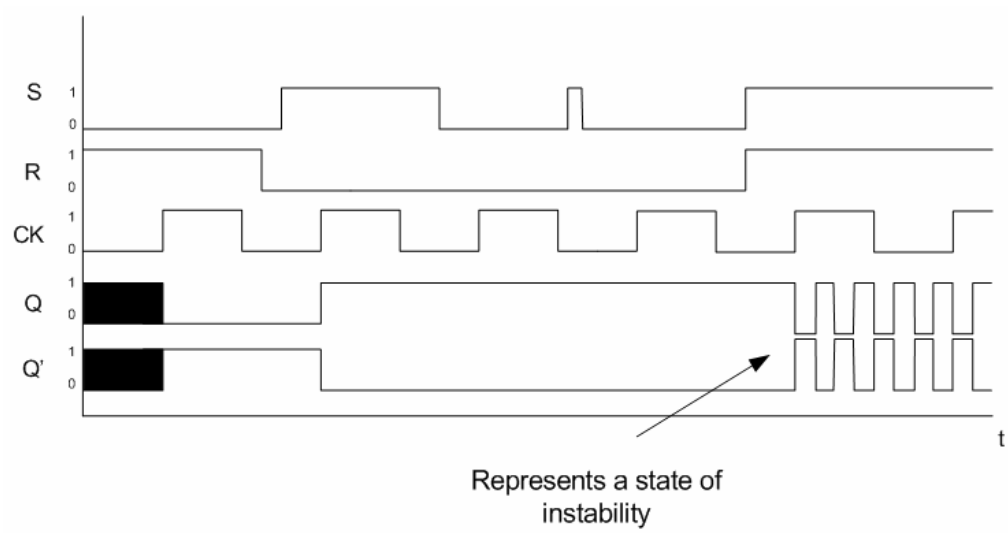
UNIVERSITY OF CALIFORNIA, BERKELEY
College of Engineering
Department of Electrical Engineering and Computer Sciences
Last modified on November 26, 2003 by Eric Chung (e_chung@uclink.berkeley.edu)

Prof. Neureuther

EECS 42
Fall 2003

Homework 11 Solutions

Problem 11.1. S-R Flip-Flop



Problem 11.2. Circuit extraction and delay
(a)

$$F1 = \overline{(A1 + B1) \bullet C1 \bullet D1}$$

$$F2 = \overline{A2 \bullet B2 + C2}$$

(b)
Gate 1

Maximum high-to-low delay: $t_{HL} = 0.69(3R_N C) = 0.69(3)(10k)(50\,fF) = 1.035ns$

Minimum high-to-low delay: $t_{HL} = 0.69(R_N + R_N + R_N \,||\, R_N)C = 0.69(10k + 10k + 5k)50\,fF = 0.865nS$

Data dependent range: $2.5t_{INV} \le t \le 3t_{INV}$

Gate 2

Maximum high-to-low delay: $t_{HL} = 0.69(2R_N C) = 0.69(2)(10k)(50\,fF) = 0.69ns$

Minimum high-to-low delay: $t_{HL} = 0.69(2R_N \,||\, R_N)C = 0.69(20k \,||\, 10k)50\,fF = 0.23ns$

Data dependent range: $\frac{2}{3}t_{INV} \le t \le 2t_{INV}$

(c)
Gate 1

Maximum low-to-high delay: $t_{LH} = 0.69(2R_P C) = 0.69(2)(10k)(50\,fF) = 0.69ns$

Minimum low-to-high delay: $t_{LH} = 0.69(2R_P \,||\, R_P \,||\, R_P)C = 0.69(20k \,||\, 10k \,||\, 10k)50\,fF = 0.138n$

Data dependent range: $0.4t_{INV} \le t \le 2t_{INV}$

Gate 2

Maximum low-to-high delay: $t_{LH} = 0.69(2R_P C) = 0.69(2)(10k)(50\,fF) = 0.69ns$

Minimum low-to-high delay: $t_{LH} = 0.69(R_P \,||\, R_P + R_p)C = 0.69(5k + 10k)50\,fF = 0.518ns$

Data dependent range: $1.5t_{INV} \le t \le 2t_{INV}$

(d)

The overall worst-case delay for Gate 1 is max(t_{HL}, t_{LH}) = $3t_{INV}$

The overall worst-case delay for Gate 2 is ax(t_{HL}, t_{LH}) = $2t_{INV}$

(e)

The fastest possible way for a signal to propagate is if no changes occur at the output when the input changes. For example, A2 = 0, B2 = 1, C2 = 0 initially. Then A2 = 0, B2 = 0, C2 = 0. The output remains at 1.

However, if one wants to consider the second-fastest propagation delay, a possibility is if A2 = 1, B2 = 0, C2 = 0, then A2 = 1, B2 = 1, C2 = 1.

Both answers are valid.

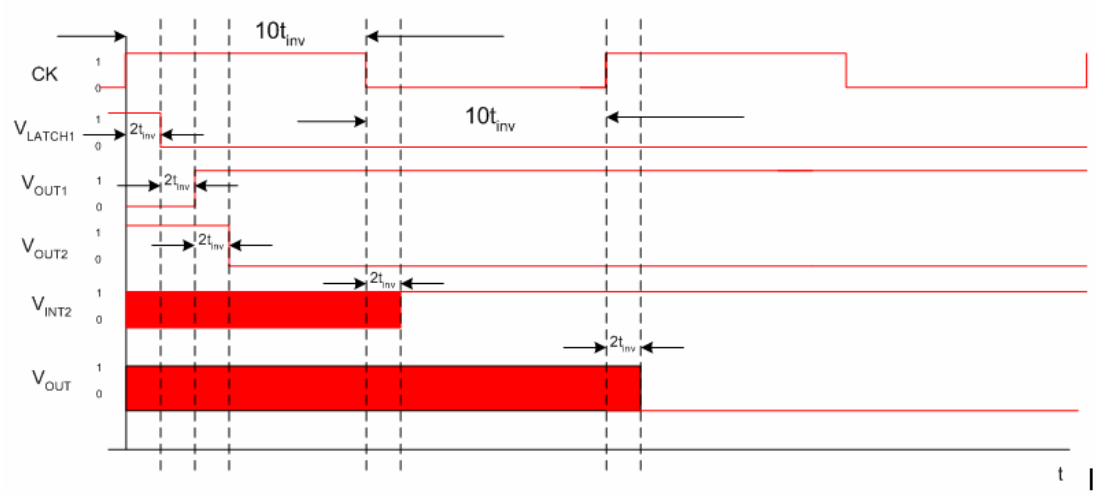
(f)

The slowest possible way for a signal to propagate is if the maximum resistance is present at the output. For example, A2 = 1, B2 = 1, C2 = 1 initially. Then A2 = 0, B2 = 1, C2 = 0.

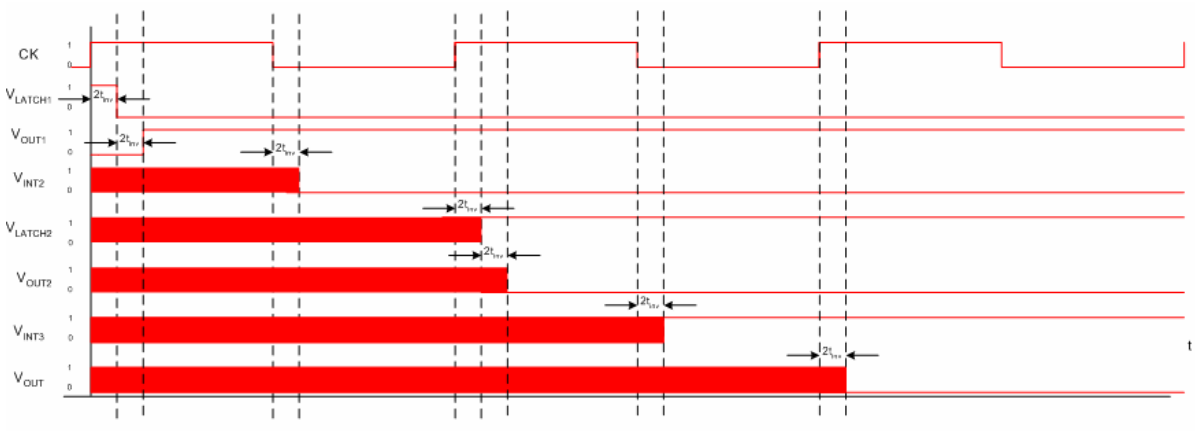
11.3 Lumped Logic. Vs. Pipelined Logic.

(a) Each of the latches, consist of two stages. The first stage of the latch is transparent when the clock is low (Ö=0) and has the delay of (0.69×2×RNC) or (0.69×2×RPC) depending on if it is going from H to L or vice versa. The second stage of the latch is transparent when clock goes high, and has the delay of (0.69×2×RPC) or (0.69×2×RNC) depending on the transition. Overall, since the outputs of the two stages are compliment, the overall delay of a latch is: (0.69×2×RUC)+(0.69×2×RDC) = 0.69×2×C(RU+RD), which is 4 inverter delays.

(b)



(c)



10.4 Latency and Throughput of Lumped and Pipelined Logic
(a)

High Delay = Second-half Delay from first latch + Delay from Gate 1 (worst-case) + Delay from Gate 2 (worst-case) = 2 + 3 + 2 = **7 inverter delays for the minimum amount of the time the clock must stay high**

Low Delay = First-half Delay from Latch 2 = **2 inverter delays for the minimum amount of the time the clock must stay low**

(b) From latch 1 to latch 2 high delay = Second-half Delay from first latch + Delay from Gate 1 (worst-case) = 2 + 3 = **5 inverter delays for the minimum amount of the time the clock must stay high**

Low delay = Delay from first-half of latch 2 = **2 inverter delays for the minimum amount of the time the clock must stay low**

From latch 2 to latch 3 delay = Delay from second-half of second latch + Delay from Gate 3 (worst-case) = 2 + 2 = **4 inverter delays for the minimum amount of the time the clock must stay high**

Low delay = Delay from first-half of latch 3 = **2 inverter delays for the minimum amount of the time the clock must stay low**

Note: The clock must stay low for 5 inverter delays since the delay from latch 1 to latch 2 is the limiting case for the entire system. If the clock was high only for 4 inverter delays, the operation from latch 1 to latch 2 would fail.

(c) Latency (lumped) = Second-Half Delay from first latch + Delay from Gate 1 (worst-case) + Delay from Gate 2 (worst-case) + First-half Delay from Latch 2 + Second-half Delay from Latch 2= 2 + 2 + 3 + 2 + 2 = **11 inverter delays (9 inverter delays if we drop the second-half delay from latch 2 by convention shown in the lecture)**

The pipelined latency is a bit tricky because it's not a simple matter of adding all the delays leading to the output. The clock is assumed to be clocked as determined in part b. Thus, T_{HIGH} = 5 and T_{LOW} = 2.

Latency (pipelined; including the last latch) = Second-Half Delay from first latch + Max(Gate 1 worst delay, Gate 2 worst delay) + First-half delay from latch 2 + Second-half delay from latch 2 + Max(Gate 1 worst delay, Gate 2 worst delay) + First-half delay from latch 3 + Second-half delay from latch 3 = 2 + 3 + 2 + 2 + 3 + 2 + 2 = 2 * (T_{LOW} + T_{HI}) = **16 inverter delays (14 inverter delays if we drop the second-half delay from latch 3 by convention shown in lecture)**

* Latency is usually defined going all the way to the output, however in this class we will drop the second-half delay from the output latch

(d) Minimum clock period (lumped) = T = T_{HIGH} + T_{LOW} = 7 + 2 = 9 inverter delays
frequency = 1 / T = 1 / (9 * 345pS) = 0.32 Ghz

Minimum clock period (pipelined) = T = T_{HIGH} + T_{LOW} = 5 + 2 = 7 inverter delays
Frequency = 1 / T = 1 / (7 * 345pS) = 0.41 Ghz

Pipelining increases throughput of data in digital circuits!