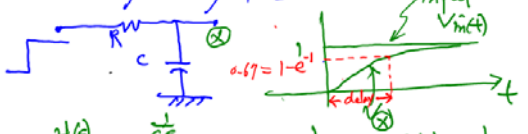
EE 222 Lecture 6 Jan 25, 2017

RC delay analysis [text pp 292-295]



$$H(s) = \frac{\frac{1}{Cs}}{R+\frac{1}{Cs}} = \frac{1}{1+RCs} = \left(\frac{1}{RC}\right)\frac{1}{s+\frac{1}{RC}}$$

zero i.c.

$$V_{in}(s) = \mathcal{L}[u(t)] = \frac{1}{s}$$

$$V_x(s) = H(s)V_{in}(s) = \frac{1}{RC}\,\frac{1}{s\left(s+\frac{1}{RC}\right)} = \frac{1}{RC}\left(\frac{RC}{s} + \frac{-RC}{s+\frac{1}{RC}}\right)$$

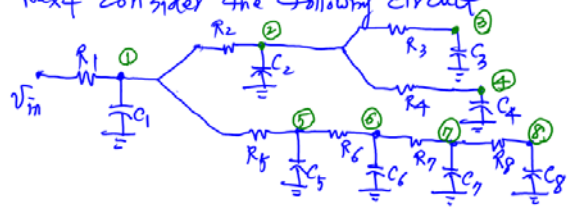
$$v_x(t) = 1 - e^{-\frac{1}{RC}t}$$

When $t = RC$, $v_x(t) = 1 - e^{-1}$

$= 1 - 0.37 = 0.67$

In this case 1 time constant corresponds to ~ 67% of the $v_x(t=\infty)$ final.

Next consider the following circuit



Let N = Total number of nodes with Capacitor

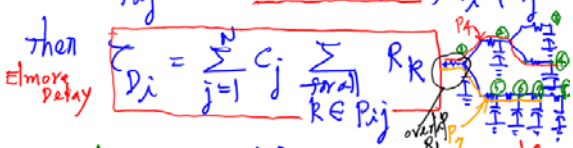
P_i = the unique path from input to node (i)

P_j = " to node (j) "

P_{ij} = the common portion of P_i & P_j

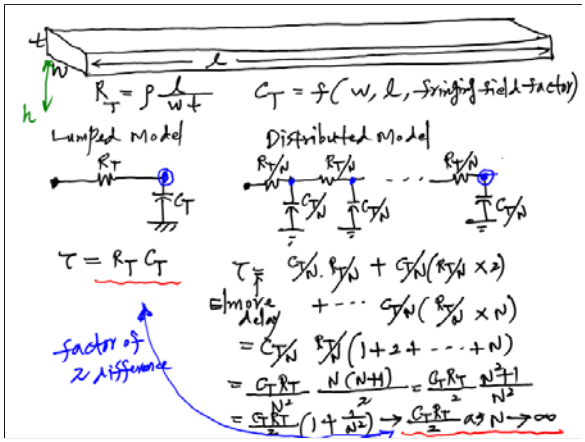
Then Elmore Delay

$$\tau_{Di} = \sum_{j=1}^{N} C_j \sum_{\text{from } k \in P_{ij}} R_k$$



Example $\tau_{D7} = \sum_{j=1}^{N=8} C_j \sum_{k\in P_{j7}} R_k$ (overlap between P_7 & P_j)

$= C_1R_1 + C_2R_1 + C_3R_1 + C_4R_1 + C_5(R_1+R_5) + C_6(R_1+R_5+R_6) + C_7(R_1+R_5+R_6+R_7) + C_8(R_1+R_5+R_6+R_7+R_8)$



Typical 65 nm technology case

$h = 3\,\mu m$ $t_{ox} = 2.6\,nm$ $t_{Poly} = 1\,\mu m$

Metal 1 thickness $= 1.8\,\mu m$

Minimum width $= 0.09\,\mu m$

$C_{metal\text{-}over\text{-}field\ oxide}(h) = 0.03\ fF/\mu m^2$ area, $0.044\ fF/\mu m$ perimeter

$$R_\square = \rho\frac{0.09\,\mu m}{1.8\times 0.09\,(\mu m)^2} = 1.6\times10^{-8}\,\Omega\cdot m\,\frac{1}{1.8\times10^{-6}\,m}$$

Al wire

$$= \frac{1.6}{1.8}\times10^{-8}\times10^{6}\,\Omega = 9\times10^{-3}\,\Omega/\square$$

For $l = 900\,\mu m$, $\frac{900\,\mu m}{0.09\,\mu m} = 10{,}000\,\square$

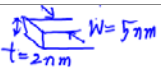
$R_{total} = 9\times10^{-3}\,\Omega/\square \times 10{,}000\,\square = 90\,\Omega$

$C_{total} = 0.03\ fF/\mu m^2 \times 0.09\times 900(\mu m)^2 + 0.044\ fF/\mu m \times 2\times 900\,\mu m$

$= 2.43\ fF + 79.2\ fF = 81.6\ fF$

$R_T C_T = 90\,\Omega \times 81.6\ fF = 7.34\ ps$ (distributed case $\approx 4\ ps$)

(Example) $l = 100\,\mu m$



$$R_T = \rho\frac{l}{A} = 1.6\times10^{-8}\,\Omega\cdot m\,\frac{10^{-4}\,m}{10\times10^{-18}\,m^2} = 1.6\times10^{5}\,\Omega = 160\,K\Omega$$

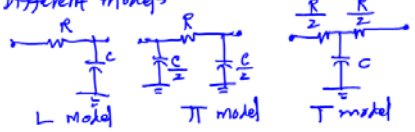
(Aluminum)

(assumed including fringe effect) $C_T = 0.05\ fF/\mu m^2 \times 100\,\mu m \times 5\times10^{-3}\,\mu m = 0.025\ fF$

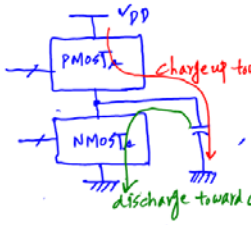
$R_T C_T = (160\times10^3)\times(0.025\times10^{-15}) = 4\ ps$ lumped

(Distributed model for large N = 2 ps)

Different models

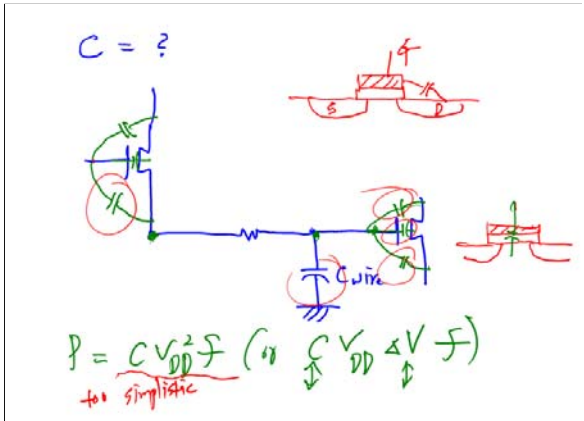


Power Meter (Ref Text section 6.7) by S. Kang



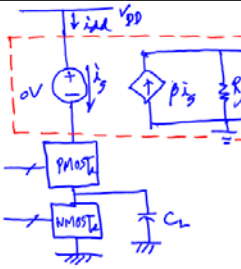
$$\text{Power} = \frac{1}{T}\int_0^T i_{dd}(t)\, V_{DD}\, dt = \left[\frac{1}{T}\int_0^T i_{dd}(t)\, dt\right] V_{DD}$$

average current drawn from the power supply



$P = C V_{DD}^2 f$ (or $C V_{DD} \Delta V f$)

too simplistic



$$C_y \frac{dV_y(t)}{dt} = \beta i_s(t) - \frac{V_y(t)}{R_y}$$

$V_y(0-) = 0$

$$V_y(t) = \frac{\beta}{C_y}\int_0^t e^{\frac{-(t-\tau)}{R_y C_y}} i_s(\tau)\, d\tau$$

when $R_y C_y \gg T$,

$$V_y(t) = \frac{\beta}{C_y}\int_0^t i_s(\tau)\, d\tau$$

choose $\beta = V_{DD} C_y / T$, then

$$V_y(T) = \frac{V_{DD}}{T}\int_0^T i_{dd}(t)\, dt = P_{avg}$$

$i_s = i_{dd}$

Example Page 293

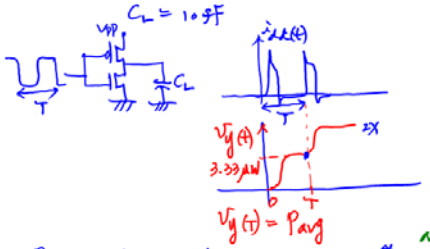
65nm technology

$V_{DD} = 1.2V$

Temp = 50°C

$W_n/L = 5/1$, $W_p/L = 2 W_n/L = 10/1$

$C_L = 10fF$



$V_y(T) = P_{avg}$

Power-Delay Product (PDP) $= C_L V_{DD}^2 \approx 2 P_{avg}^{*} \tau_p$

τ_p: average prop. delay

P_{avg}^{*}: average switching power at max. op. freq.

Energy-Delay Product (EDP)

$$EDP = PDP \times \tau_p = C_L V_{DD}^2 \tau_p \approx 2 P_{avg}^{*} \cdot \tau_p^2$$

(if leakage, short ckt powers are neglected)

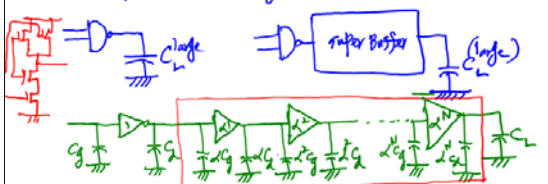
64b adder A & B

	delay (ps)	power (mW)	PDP (fJ)	EDP (10^{-24} J·s)
Add-A	180	40	7.2	1.296
Add-B	240	30	7.2	1.728

30% faster

captures FOM for Add-A

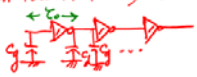
Super Buffer Design



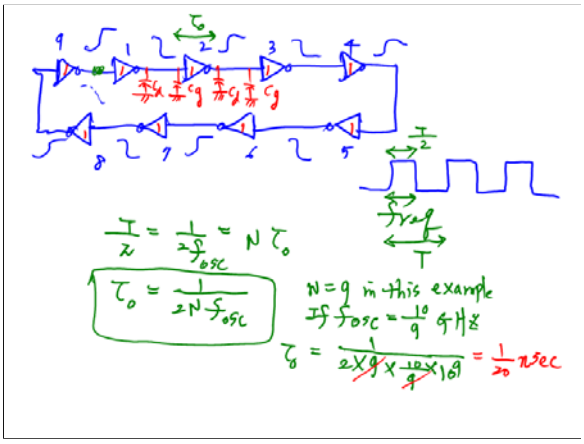
choose α, N such that $\alpha^N C_g = C_L^{large}$

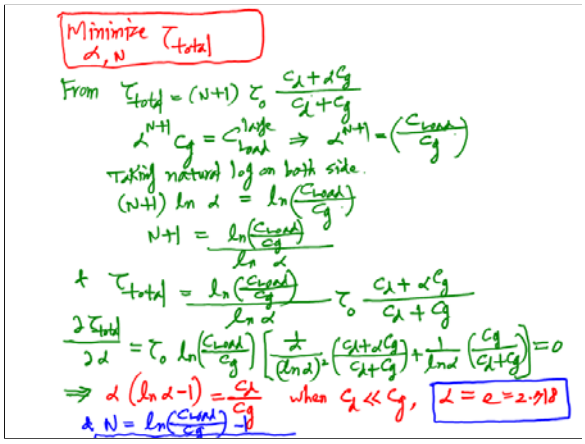
In the chain each inverter has an identical delay of

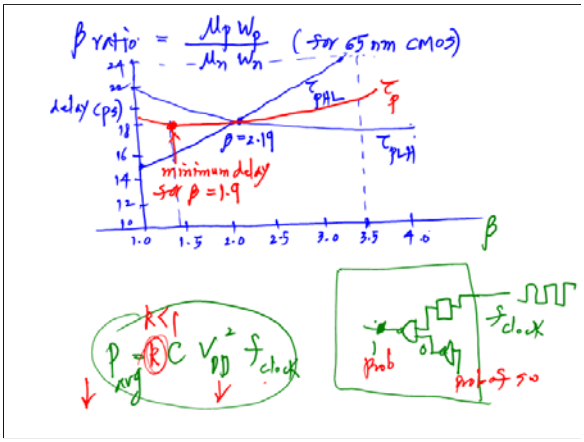
$$\tau_0 \left(\frac{C_d + \alpha C_g}{C_d + C_g}\right)$$

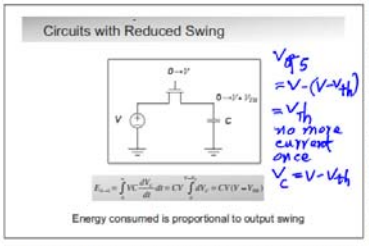


thus $\tau_{total} = (N+1)\,\tau_0 \left(\frac{C_d + \alpha C_g}{C_d + C_g}\right)$



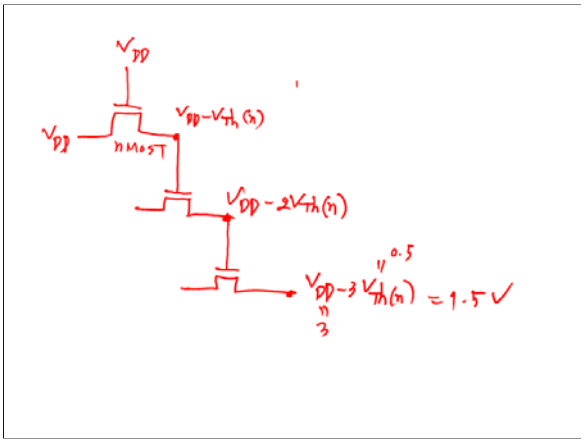


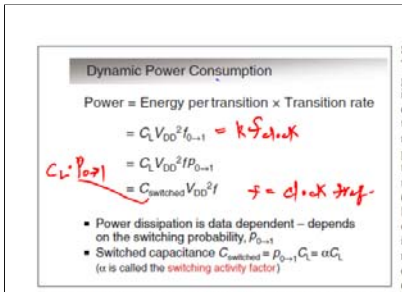




Slide 3.8
More generically, we can compute the energy it takes to charge a capacitance from a voltage V_1 to a voltage V_2. Using similar math, we derive that this requires from the supply an amount of energy equal to $CV_2(V_2 - V_1)$. This equation will come in handy for a number of special circuits. One example is the NMOS pass-transistor chain. It is well-known that the maximum voltage at the end of such as chain is one threshold voltage below the supply [Rabaey03]. Using the afore-derived equation, we find that the energy dissipation in this case equals $CV_{DD}(V_{DD} - V_{TH})$, and is proportional to the swing at the output. In general, reducing the swing in a digital network results in a linear reduction in energy consumption.

[R1]

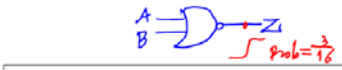




Slide 3.12
This brings us back to the generic case of the CMOS inverter. To translate the derived energy per operation into power, it must be multiplied with the rate of power-consuming transitions $f_{0\to1}$. The unit of the resulting metric is Watt (= Joules/sec). This translation leads right away to one of the hardest problems in power analysis and optimization: it requires knowledge of the "activity" of the circuit. Consider a circuit with a clock frequency f. The probability that a node will make a 0-to-1 transition at a given clock tick is given by γf, where $0 \le \gamma \le 1$ is the activity factor at that node. As we discuss in the following slides, γ is a function of the circuit topology and the activity of the input signals. The accuracy of power estimation depends largely upon how well the activity is known – which is most often not very much.

[R1]

Also, F. Najm, R. Burch, P. Yang and I. Hajj, "Probabilistic simulation for Reliability Analysis of CMOS VLSI circuits", IEEE Trans. on CAD, April 1990 (Best Paper Award)



Impact of Logic Function

Example: Static two-input NOR gate

A	B	Out
0	0	1
0	1	0
1	0	0
1	1	0

Assume signal probabilities
$p_{A=1} = 1/2$
$p_{B=1} = 1/2$

Then transition probability
$p_{0\to1} = p_{out=0} \times p_{out=1}$
$= 3/4 \times 1/4 = 3/16$

If inputs switch every cycle
$\alpha_{NOR} = 3/16$

NAND gate yields similar result

Slide 3.13
Let us, for instance, derive the activity of a two-input NOR gate (which defines the topology of the circuit). Assume that each input has an equal probability of being a 1 or a 0, and that the probability of a transition at a clock tick is 50–50 as well, ensuring an even distribution between states. With the aid of the truth table we derive that the probability of a 0→1 transition (or the activity) equals 3/16. More generally, the activity at the output node can be expressed as a function of the 1-probabilities of the inputs A and B: $\alpha_{NOR} = p_A p_B (1 - p_A p_B)$.

[R1]

Impact of Logic Function

Example: Static two-input XOR gate

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

Assume signal probabilities
$p_{A=1} = 1/2$
$p_{B=1} = 1/2$

Then transition probability
$p_{0\to1} = p_{out=0} \times p_{out=1}$
$= 1/2 \times 1/2 = 1/4$

If inputs switch every cycle
$p_{0\to1} = 1/4$

Slide 3.14
A similar analysis can be performed for an XOR gate. The observed activity is a bit higher (1/4).

[R1]

Transition Probabilities for Basic Gates

As a function of the input probabilities

	$p_{0\to1}$
AND	$(1-p_A p_B)p_A p_B$
OR	$(1-p_A)(1-p_B)(1-(1-p_A)(1-p_B))$
XOR	$(1-(p_A+p_B-2p_A p_B))(p_A+p_B-2p_A p_B)$

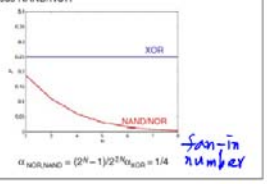
Activity for static CMOS gates
$\alpha = p_{0\to1}$

Slide 3.15
These results can be generalized for all basic gates.

[R1]

Activity as a Function of Topology

XOR versus NAND/NOR

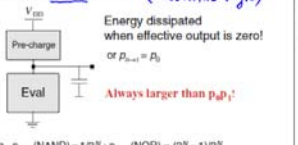


$\alpha_{NOR,NAND} = (2^N - 1)/2^{2N}\alpha_{XOR} = 1/4$

Slide 3.16
The topology of the logic network has a major impact on the activity. This is nicely illustrated by comparing the activity of NAND (NOR) and XOR gates as a function of fan-in. The output-transition probability of a NAND gate goes asymptotically to zero. The probability of the output being a 0 is indeed becoming smaller with increasing fan-in. An example of such a network is a memory-address decoder. On the other hand, the activity of an XOR network is independent of fan-in. This does not bode well for the power dissipation of modules such as large en(de)cryption and coding functions, which primarily consist of XORs.

[R1]

How About Dynamic Logic? (domino logic)



Energy dissipated when effective output is zero!

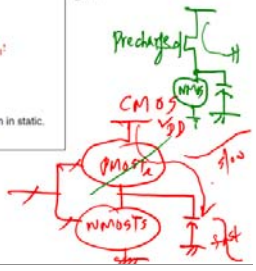
or $p_{0\to1} = p_0$

Always larger than $p_0 p_1$!

E.g., $p_{0\to1}(\text{NAND}) = 1/2^N$; $p_{0\to1}(\text{NOR}) = (2^N - 1)/2^N$

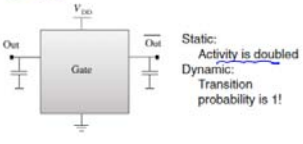
Activity in dynamic circuits hence always higher than in static. But ... capacitance most often smaller.

activity and capacitance, the latter being smaller in dynamic logic. In general though, the higher activity outweighs the capacitance gain.



domino CMOS invented by AT&T Bell Labs (R. Krambeck, C. Lee, S. Law)

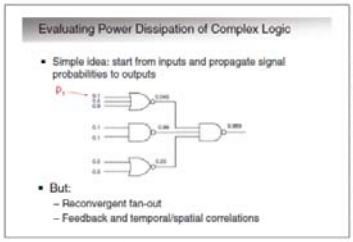
Differential Logic?



Static:
Activity is doubled
Dynamic:
Transition probability is 1!

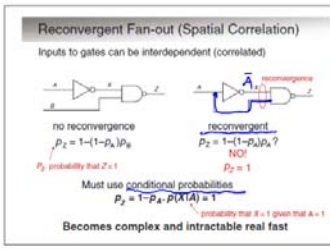
Hence power always increases.

Slide 3.18
Another interesting logic family is *differential logic*, which may seem attractive for very low-voltage designs due to its increased signal-to-noise ratio. Differential implementations come unfortunately with an inherent disadvantage from a power perspective: not only is the overall capacitance higher, the activity is higher as well (for both static and dynamic implementations). The only positive argument is that differential implementation reduces the number of gates needed for a given function, and thus reduces the length of the critical path.



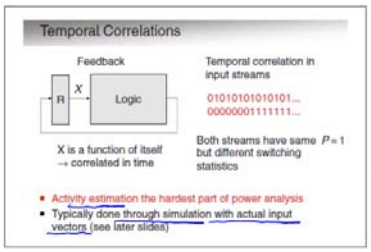
static power analysis seem favorable at a first glance. Consider, for instance, the network shown on the slide, and assume that the 1- and 0-probabilities of the primary input signals are known. Using the basic gate expressions presented earlier, the output signal probabilities can be computed for the first layer of gates starting from the primary inputs. This process is then repeated until the primary outputs are reached.

This process seems fairly straightforward indeed. However, there is a catch. For the basic gate equations to be valid, the inputs must be statistically independent. In probability theory, to say that two events are *independent* intuitively means that the occurrence of one event makes it neither more nor less probable that the other occurs. While this assumption is in general true for the network of the slide (assuming obviously that all the primary input signals are independent), it unfortunately rarely holds in actual circuits.



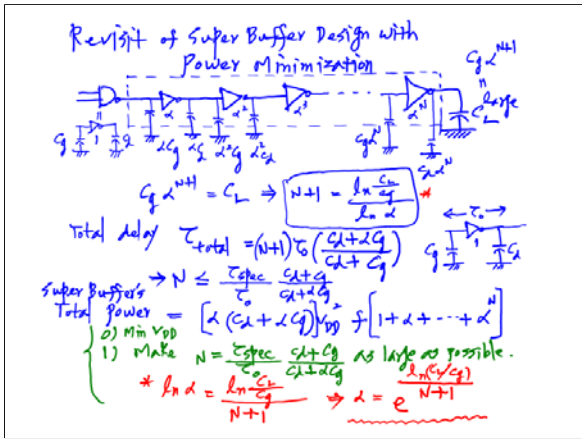
Slide 3.20
Even if the primary inputs to a logic network are independent, the signals may become correlated or "colored", while they propagate through the logic network. This is best illustrated with a simple example, which showcases the impact of a network property called *reconvergent fan-out*. In the rightmost circuit, the inputs to the NAND gate Z are not independent, but are both functions of the same input signal A. To compute the output probabilities of Z, the expression derived earlier for a NAND gate is no longer applicable, and conditional probabilities need to be used. *Conditional probability* is the probability of some event A, given the occurrence of some other event B. Conditional probability is expressed as $p(A|B)$, and is read as "the probability of A, given B". More specifically, one can derive that $p(A|B) = p(A \cap B)/p(B)$, assuming that $p(B) \neq 0$.

While propagating these conditional probabilities through the network is theoretically possible, you may guess that the complexity of doing so for complex networks rapidly becomes unmanageable – and that indeed is the case.



Slide 3.21
The story gets complicated even further by the occurrence of *temporal correlations*. A signal shows temporal correlation if a data value in the signal stream is dependent upon previous values in the stream. Temporal correlations are the norm in sequential networks, as any signal in the network is typically a function of its previous values owing to the existence of feedback network. In addition, primary input signals as well may show temporal dependence. For example, in a digitized speech signal any sample value is dependent upon the previous values.

All these arguments help to illustrate that static activity analysis is a very hard problem indeed, and actually all but impossible. Hence, power analysis tools either rely on simulations of actual signal traces to derive the signal probabilities or make simplifying assumptions – for instance, it is assumed that the input signals are independent and purely random. This is discussed in more detail in Chapter 12. In the following chapters, we will most often assume that activity of a module in its typical operation mode can be characterized by an independent parameter α.



(Example) For a CMOS technology with $C_{ox} = 22\ fF/m^2$

$L = 60\ nm$, $W_n + W_p = 600\ nm$

$C_g = C_{ox} \cdot (W_n + W_p) L$

$= 22\ fF/m^2 (600 \times 10^{-9}\ m)(60 \times 10^{-9}\ m)$

$= 22 \times 36 \times 10^{-15}\ F \approx 0.79\ pF$

Suppose $C_{load} = C_g e^7 = 0.79\ pF (e^7) \approx 1117.2\ pF$

Then $N+1 = \ln \frac{C_g e^7}{C_g} = 7 \Rightarrow N = 6$

$\tau_{total} \cong 6\tau_0$ ← $\tau_0 \left(\frac{C_d + \alpha C_g}{C_d + C_g}\right)$

$P_{total} = \alpha (C_d + \alpha C_g) V_{DD}^2 f (1 + \alpha + \alpha^2 + \cdots + \alpha^N)$, $N = 6$

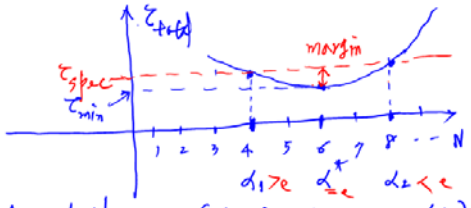
$\approx e (C_d + e C_g) V_{DD}^2 f (1 + e + e^2 + \cdots + e^6)$

For min. delay

For min. power $N+1 = \frac{\tau_{spec}}{\tau_0 \left(\frac{C_d + \alpha C_g}{C_d + C_g}\right)}$ Assume τ_0 be, say 8

$P_{total} = \alpha (C_d + \alpha C_g) V_{DD}^2 f (1 + \alpha + \alpha^2 + \cdots + \alpha^8)$

$\alpha < e = 2.718$



$N=4 \quad P = \alpha_1 \left(C_d + \alpha_1 C_g\right) V_{DD}^2 f \left(1 + \alpha_1 + \alpha_1^2 + \alpha_1^3 + \alpha_1^{N=4}\right)$

Active Area $= \sum_{k=0}^{N=4} \left(W_k + W_p\right) \alpha_1^k$

$N=4 \quad P = \alpha_2 \left(C_d + \alpha_2 C_g\right) V_{DD}^2 f \left(1 + \alpha_2 + \alpha_2^2 + \cdots + \alpha_2^{N=8}\right)$

Active Area $= \sum_{k=0}^{N=8} \left(W_n + W_p\right) \alpha_2^k$