inst.eecs.berkeley.edu/~eecs151

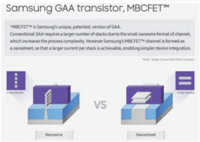
EECS151 : Introduction to Digital Design and ICs

Lecture 15 – Logical Effort

Bora Nikolić

Samsung Foundry Promises Gate All-Around in '22

October 14, 2021, EETimes - Samsung Foundry recently held its Foundry Forum where it revealed some details of its semiconductor process roadmaps and fab expansion. Samsung is being most aggressive pursuing the next generation of transistor technology, with plans to reach mass production ahead of TSMC and Intel. Samsung's 3-nanometer process will use the gate-all-around (GAA) transistor structure, which the foundry calls MBCFET (Multi-bridge channel FET) and will be in production first half of 2022. TSMC will wait another generation until its N2 process to deliver GAA some time in 2023.



Samsung GAA transistor, MBCFET™

*MBCFET™ is Samsung's unique, patented, version of GAA. Conventional GAA requires a larger number of stacks due to the small nanowire format of channel, which increases the process complexity. However Samsung's MBCFET™ channel is formed as a nanosheet, so that a larger current per stack is achievable, enabling simpler device integration.

EETimes

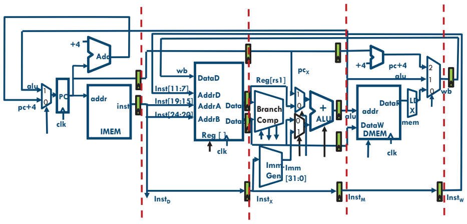
Review

- Delay is a linear function of R and C
- Delay optimization is critical to improve the frequency of the circuit.
- The dimensions of a transistor affect its capacitance and resistance.
- We use RC delay model to describe the delay of a circuit.

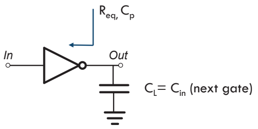
Minimizing Logic Delay

How Do We Optimize the Delay?

- How fast can a pipelined processor run?
- What is the fastest adder?



Inverter RC Delay



- $t_p = R_{eq}(C_p + C_L) = Req(Cin/\gamma + C_L)$
  - $\gamma = 1$ (closer to 1.2 in recent processes)
- $t_p = R_{eq}C_{in}(1+C_L/C_{in}) = \tau_{INV}(1+f)$
  - Propagation delay is proportional to fanout
- Normalized Delay = 1 + f

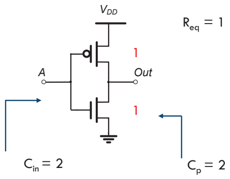
Fanout = $f = C_L/C_{in}$

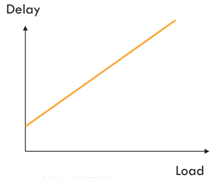
$t_p = \tau_{INV}(1+f)$

Generalizing to Arbitrary Gates

- Delay has two components: $d = f + p$
- f: *effort delay* = gh (a.k.a. stage effort)
  - Again has two components
- g: *logical effort*
  - Measures relative ability of gate to deliver current
  - $g = 1$ for inverter
- h: *electrical effort* = C_{out} / C_{in}
  - Ratio of output to input capacitance
  - Sometimes called fanout
- p: parasitic delay
  - Represents delay of gate driving no load
  - Set by internal parasitic capacitance

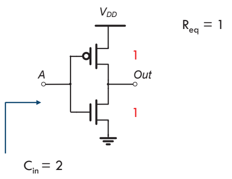
Inverter Delay

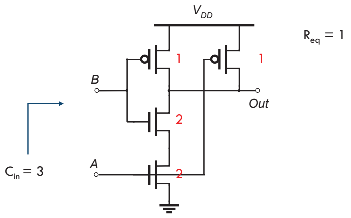




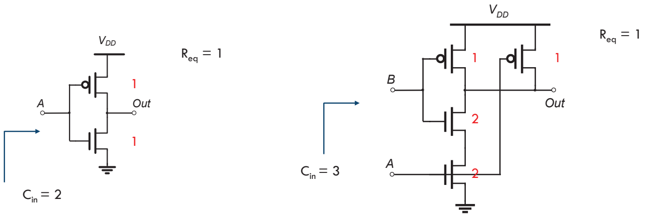
- Parasitic p is the ratio of intrinsic capacitance to an inverter
  - p(inverter) =
- Logical Effort g is the ratio of input capacitance to an inverter
  - g(inverter) =
- Electrical Effort h is the ratio of the load capacitance to the input capacitance
  - h(inverter) =
- Delay = p + f = p + g * h = 1 + f

NAND2 Gate



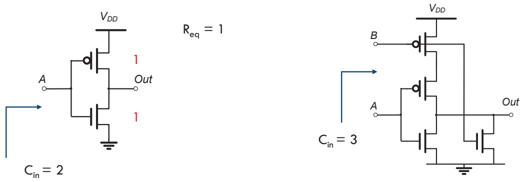


Logical Effort of NAND2 Gate

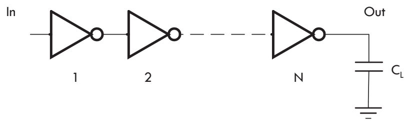


- In velocity-saturated devices Ion of a stack is 2/3 (not a half) of two devices
  - So the correct upsizing factor is 1.5 (not 2)
- We will use 2, as it makes calculations easier

NOR2 Gate



Example: Inverter Chain



Logical Effort: g =

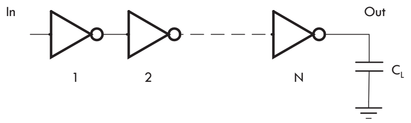
Electrical Effort: h =

Parasitic Delay: p =

Stage Delay: d =

Total Delay: d_total =

Example: Inverter Chain



Logical Effort: g = 1

Electrical Effort: h = 1

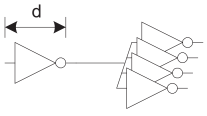
Parasitic Delay: p = 1

Stage Delay: d = 2

Total Delay: d_total = 2*N

Example: FO4 Inverter

- Estimate the delay of a fanout-of-4 (FO4) inverter



Logical Effort: g =

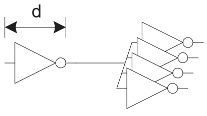
Electrical Effort: h =

Parasitic Delay: p =

Stage Delay: d =

Example: FO4 Inverter

- Estimate the delay of a fanout-of-4 (FO4) inverter



Logical Effort: g = 1

Electrical Effort: h = 4

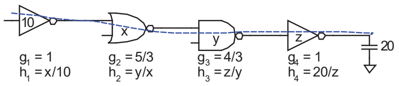
Parasitic Delay: p = 1

Stage Delay: d = 5

Fanout-of-4 is commonly used to normalize the delay across technologies

Multi-stage Logic Networks

- Logical effort generalizes to multistage networks
- *Path Logical Effort* $G = \prod g_i$
- *Path Electrical Effort* $H = \dfrac{C_{\text{out-path}}}{C_{\text{in-path}}}$
- *Path Effort* $F = \prod f_i = \prod g_i h_i$



Branching Effect

$$b = \frac{C_{\text{on path}} + C_{\text{off path}}}{C_{\text{on path}}} \qquad B = \prod b_i$$

G = 1

H = 90 / 5 = 18

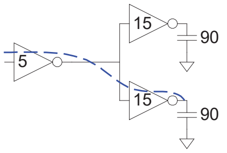
GH = 18

h_1 = (15 +15) / 5 = 6

h_2 = 90 / 15 = 6

B = 2

F = $g_1 g_2 h_1 h_2$ = 36 = BGH



Designing Fast Circuits

$$D = \sum d_i = D_F + P$$

- Delay is smallest when each stage bears same effort

$$\hat{f} = g_i h_i = F^{\frac{1}{N}}$$

- Thus minimum delay of N stage path is

$$D = NF^{\frac{1}{N}} + P$$

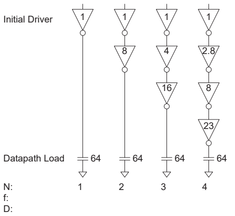
- This is a key result of logical effort
  - Find fastest possible delay
  - Doesn't require calculating gate sizes

Example: Best Number of Stages

- How many stages should a path use?
  - Minimizing number of stages is not always fastest
- Example: drive 64-bit datapath with unit inverter

D $= NF^{1/N} + P$

$= N(64)^{1/N} + N$

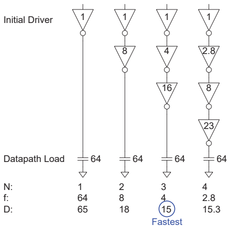


Example: Best Number of Stages

- How many stages should a path use?
  - Minimizing number of stages is not always fastest
- Example: drive 64-bit datapath with unit inverter

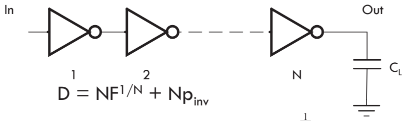
D $= NF^{1/N} + P$

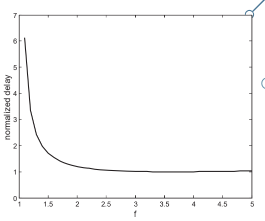
$= N(64)^{1/N} + N$



Best Stage Effort

- How many stages should a path use?
  - To drive given capacitance





$D = NF^{1/N} + Np_{inv}$

- Define best stage effort $\rho = F^{\frac{1}{N}}$
- Neglecting parasitics ($p_{inv} = 0$), we find $\rho = e = 2.718$
- For $p_{inv} = 1$, solve numerically for $\rho = 3.59$
- Choose 4 – less stages, less energy

Logical Efforts Method

1) Compute path effort $F = GBH$

2) Estimate best number of stages $N = \log_4 F$

3) Sketch path with N stages $D = NF^{\frac{1}{N}} + P$

4) Estimate least delay $\hat{f} = F^{\frac{1}{N}}$

5) Determine best stage effort $C_{in_i} = \dfrac{g_i C_{out_i}}{\hat{f}}$

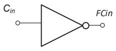
6) Find gate sizes

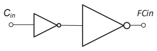
Administrivia

- Homework 6 due this week
- Projects (ASIC and FPGA) start this week

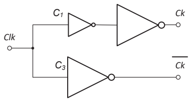
Logical Effort Design Examples

- For which F should we buffer?





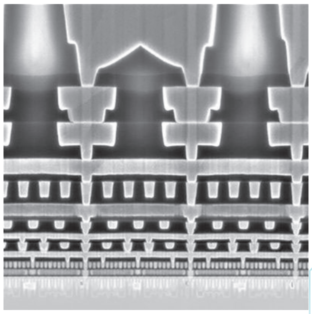
- Sizing inverter fork



Wires

A modern technology is mostly wires

- Transistors are little things under the wires
- Many layers of wires
- Wires are as important as transistors
  - Speed and power

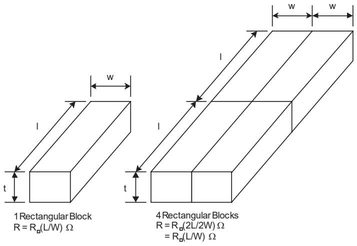


Wire Resistance

- ρ = *resistivity* (Ω*m)

$$R = \frac{\rho}{t}\frac{l}{w} = R_\square \frac{l}{w}$$

- R□ = *sheet resistance* (Ω/□)
  - □ is a dimensionless unit(!)
- Count number of squares
  - R = R□ * (# of squares)

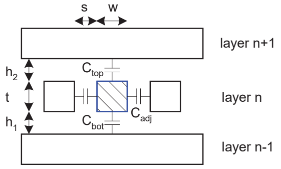


1 Rectangular Block
$R = R_\square(L/W)\ \Omega$

4 Rectangular Blocks
$R = R_\square(2L/2W)\ \Omega$
$= R_\square(L/W)\ \Omega$

Wire Capacitance

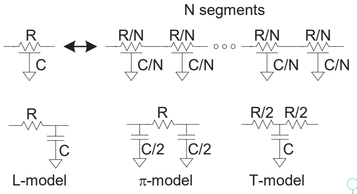
- Wire has capacitance per unit length
  - To neighbors
  - To layers above and below
- $C_{total} = C_{top} + C_{bot} + 2C_{adj}$



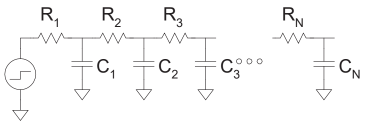
Wire Delay

Wire RC Model

- Wires are a distributed system
  - Approximate with lumped element models
- 3-segment pi-model is accurate to 3% in simulation

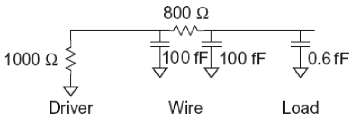


Elmore Delay for RC Tree

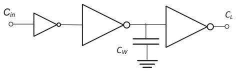


$$t_{pd} \approx \sum_{\text{nodes } i} R_{i-to-source} C_i$$
$$= R_1C_1 + (R_1 + R_2)C_2 + \ldots + (R_1 + R_2 + \ldots + R_N)C_N$$

Example: RC Delay with Wire and Gate



Logical Effort with Wires



Summary

- Two delay components in logical effort:
    - Parasitic delay (p)
    - Effort delay (F)
        - Logical effort (g): intrinsic complexity of the gate
        - Electrical effort (h): load capacitance dependent
- Wires are modelled as RC
    - Most commonly just C for hand analysis