Logic Gate Delay

Chip designers need to choose:

- What is the best circuit topology for a function?
- How many stages of logic produce least delay?
- How wide transistors should be?

Logical Effort

Helps make the above decisions.

Uses a simple delay model

Allows easy hand calculations

Compare alternative designs easily

Express delay in process independent terms

$$d = d_{abs}/\tau$$ *e.g.* τ *= 12 ps in 180nm, 40 ps in 0.6* μm

Delay has two components

$$d = f + p$$ *where,*

f = *Effort Delay (stage effort)=* gh

p =*Parasitic Delay*

Logic Gate Delay

g logical Effort

- Measures relative ability of gate to deliver current
- 1 for inverter

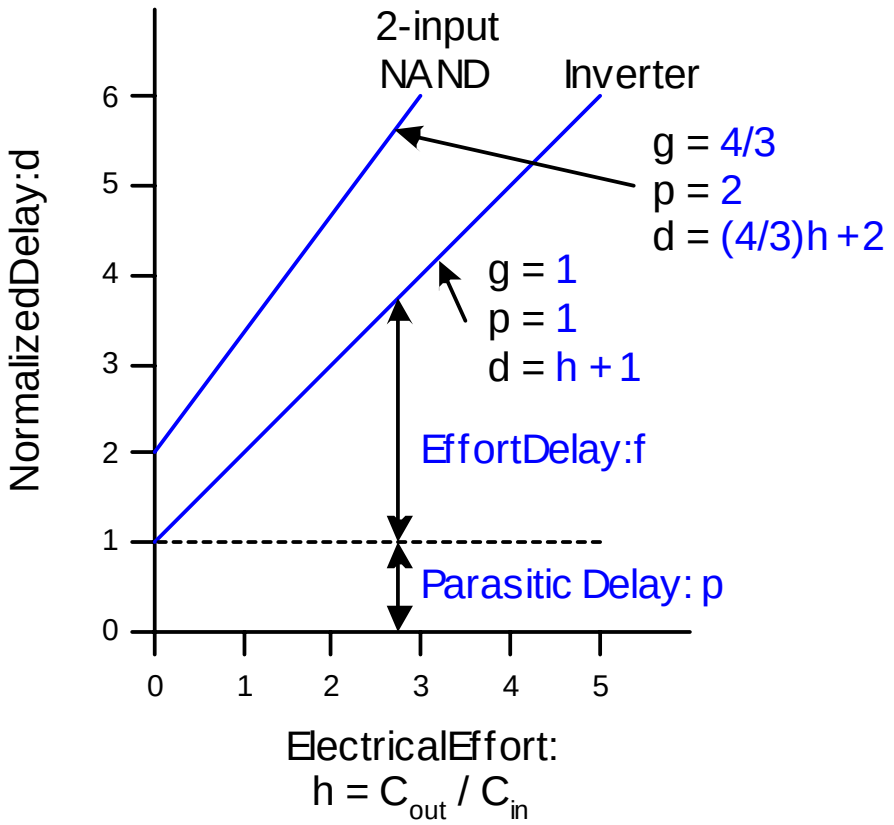
h electrical effort = C_{out}/C_{in}

- Ratio of output to input capacitance
- Sometimes called fanout

p parasitic delay

- Represents delay of gate driving no load
- Set by internal parasitic capacitance

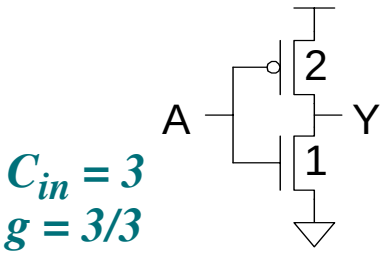
Again $d = f + p = gh + p$



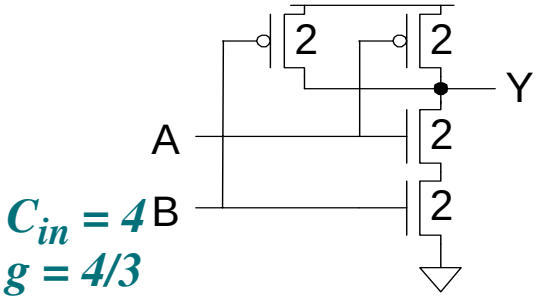
Logical Effort

Logical Effort: *It is the ratio of the input capacitance of a gate to the input capacitance of an inverter delivering the same output current.*

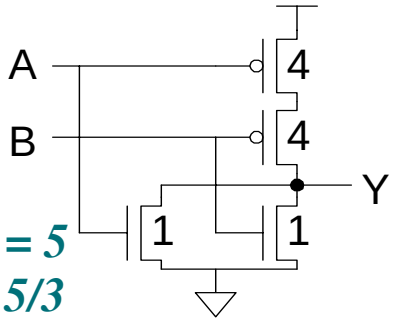
- Can be measured from delay vs. fanout plots
- Or estimate by counting transistor widths



$C_{in} = 3$
$g = 3/3$



$C_{in} = 4$
$g = 4/3$



$C_{in} = 5$
$g = 5/3$

Gate Type	Number of Inputs				
	1	2	3	4	n
Inverter	1				
NAND		4/3	5/3	6/3	(n+2)/3
NOR		5/3	7/3	9/3	(2n+1)/3
Tristate, Mux	2	2	2	2	2
XOR, XNOR		4,4	6,12,6	8,16,16,8	

Parasitic Delay

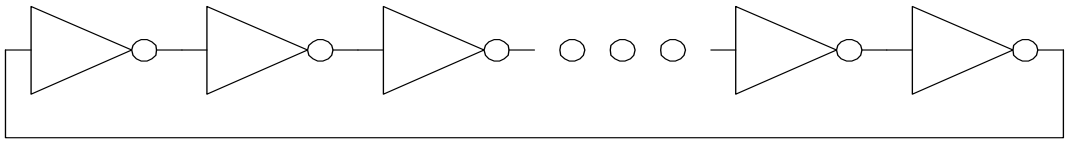
Count diffusion capacitance on the output assuming contacted diffusions.

- Inverter: 3 units of diffusion capacitance, parasitic delay is 3RC = τ.
- Normalized parasitic p_{inv} is.
- p_{inv} is the ratio of diffusion capacitance to gate capacitance for a particular process. Is considered close to 1 for simplicity
- More refined parasitic delay estimations can be performed using Elmore delay. Internal diffusion capacitance are considered, delay grows quadratically rather than linearly as estimated by the crude method.
- Parasitic delay for common gates using the crude method

Gate Type	*Number of Inputs*				
	1	*2*	*3*	*4*	*n*
Inverter	1				
NAND		2	3	4	n
NOR		2	3	4	n
Tristate, Mux	2	4	6	8	2n

Example: Ring Oscillator and FO4 inverter

- Estimate the frequency of a N-stage ring oscillator

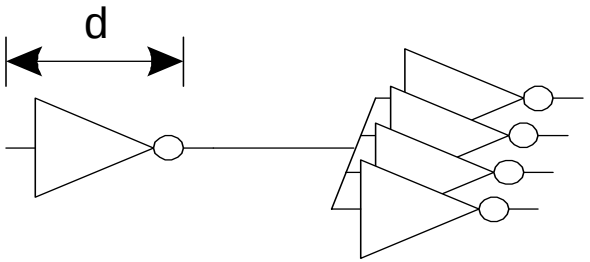


  - Logical Effort g = 1, Electrical Effort h = 1, Parasitic Delay p = 1
  - Stage Delay d = 2
  - Frequency fosc = 1 / (2 * N * d) = 1 / 4N

  Period = 2N (edge has to propagate twice through the ring to attain original polarity)

31 stage ring oscillator in 0.6 µm technology has frequency of ~ 200MHz.

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

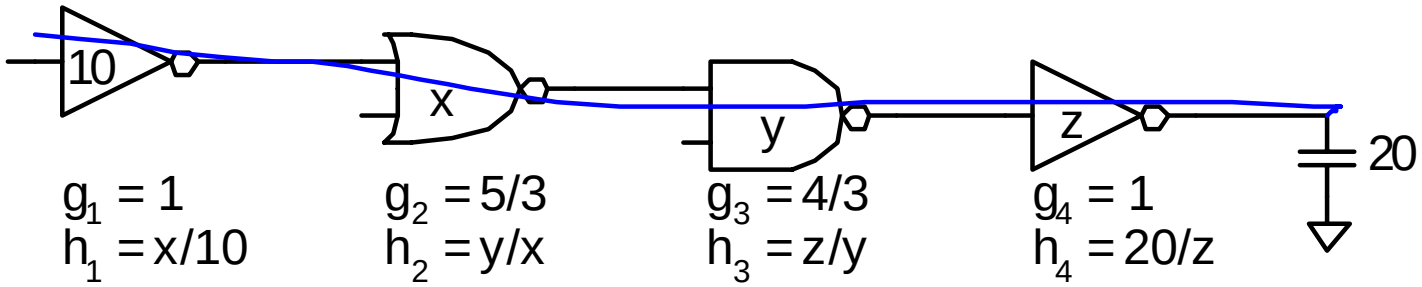


  - Logical Effort g = 1, Electrical Effort h = 4, Parasitic Delay p = 1
  - Stage Delay d = 5
- The FO4 delay is: 200ps in 0.6 µm, 60ps in 180nm, ~f/3 ns in an f µm process.

Multistage Logic Networks

Logical Effort generalizes to multistage networks

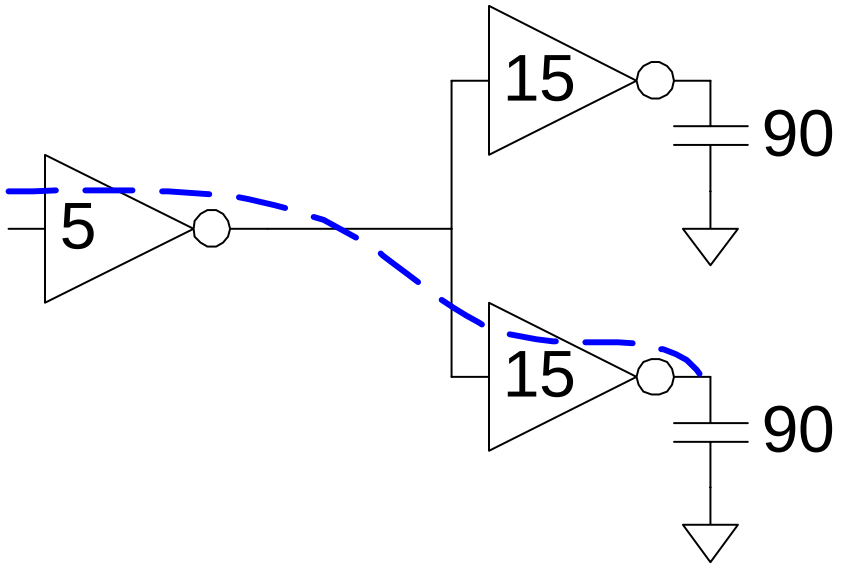
- Path Logical Effort: $G = \prod g_i$
- Path Electrical Effort: $H = \frac{C_{outpath}}{C_{inpath}}$
- Path Effort: $F = \prod f_i = \prod g_i h_i$



Thus by analogy is F = GH?

NO! Due to branching paths.

Multistage Logic Networks: Branching Effort



G = 1

H = 90/5 = 18

GH = 18

$h_1 = (15 + 15) / 5 = 6$

$h_2 = 90/15 = 6$

$F = g_1 g_2 h_1 h2 = 36 = 2GH$

Thus we need to introduce ***branching effort***.

Multistage Delay

- *Branching Effort*

  Accounts for branching between stages in path

$$b = \frac{C_{onpath} + C_{offpath}}{C_{onpath}}$$

$$B = \prod b_i$$

- *Path Effort*: $F = GBH$
- *Path Effort Delay*: $D_F = \sum f_i$
- *Path Parasitic Delay*: $P = \sum p_i$
- *Path Delay*: $D = \sum d_i = D_F + P$

Designing Fast Circuits

Delay is the smallest when each stage bears the same effort $\hat{f}$, with N stages in the path

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

Thus, minimum delay of N stage path is

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

The above equation helps to find fastest possible delay without calculating gate sizes.

Capacitance transformation used to used to calculate gate widths.

$$\hat{f} = gh = g\frac{C_{out}}{C_{in}}$$

Working backward, apply capacitance transformation to find input capacitance of each gate given load it drives

Check work by verifying input capacitance specification is met.

Logical Effort Example: 3-stage path

Logical Effort G=(4/3) * (5/3) * (5/3) = 100/27

Electrical Effort H= 45/8

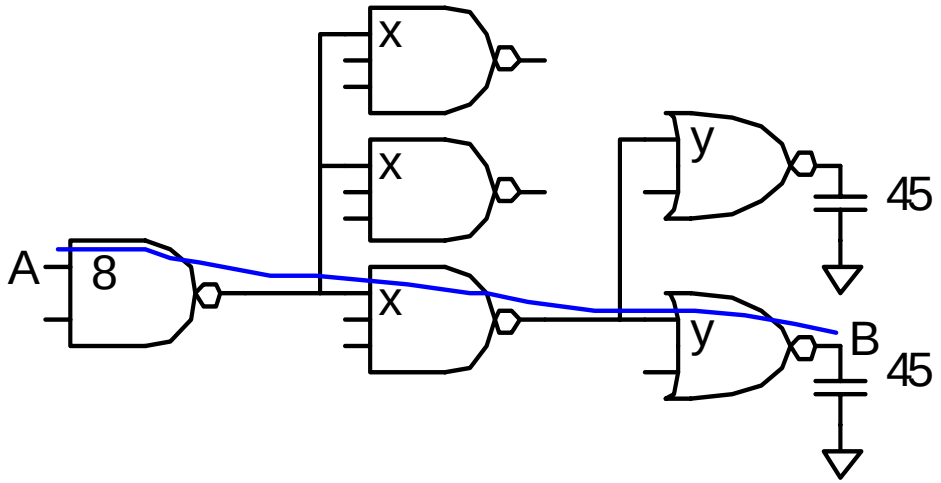
Branching Effort G= 3 *2 = 6

Path Effort F=GBH= 125

Best Stage Effort $\hat{f} = \sqrt[3]{F} = 5$

Parasitic Delay P = 2 + 3 + 2 = 7

Delay D = 3 * 5 + 7 = 22 = 4.4 FO4

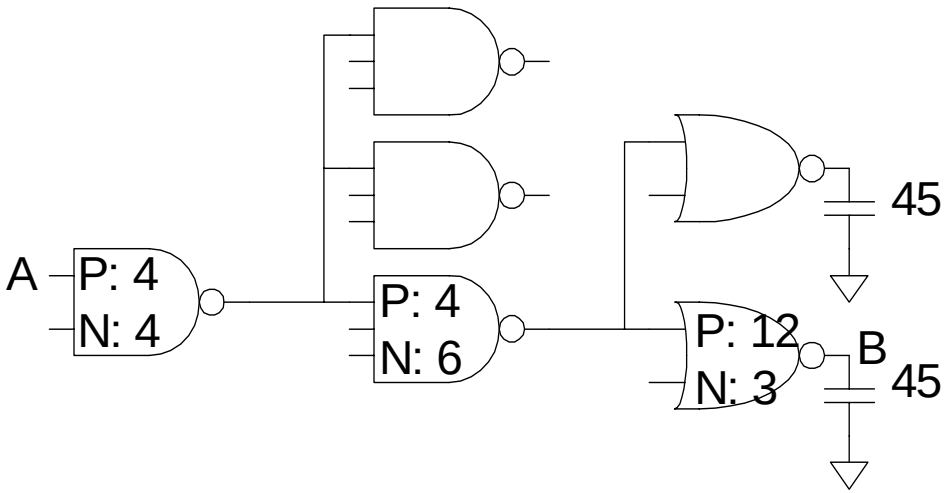


For best sizes work backward

y = 45 * (5/3) / 5 = 15
x = (15 * 2) * (5/3) / 5 = 10

Sizes chosen to get equal rise-fall times

Check path input capacitance to check values
(10 + 10 + 10) (4/3) / 5 = 8



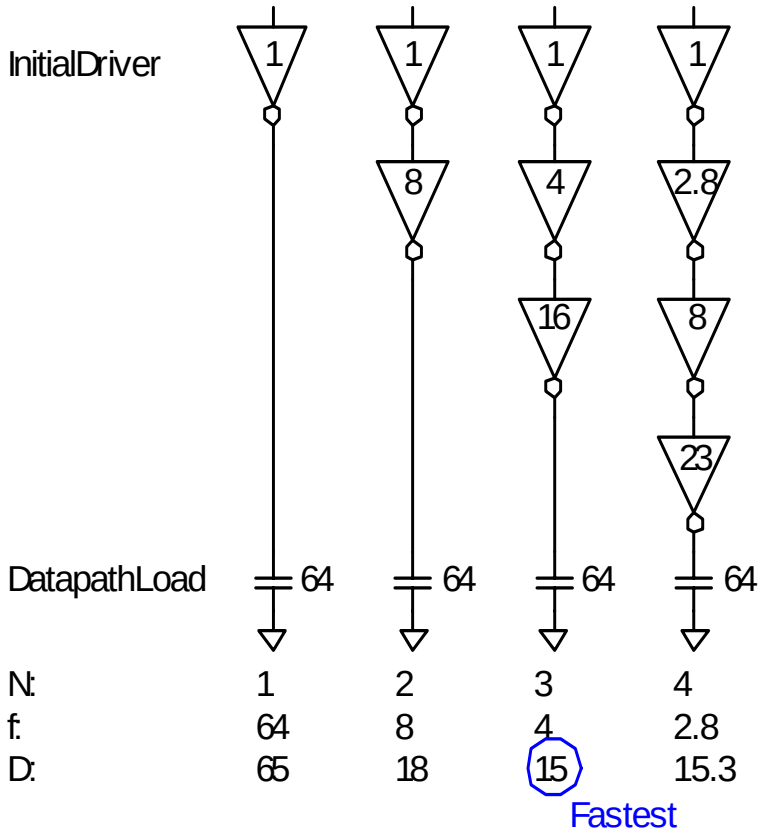
Best Number of Stages

Another important choice is the number of stages in a path

Minimum number of stages does not provide best delay in all cases

E.g. drive 64-bit datapath with unit inverter

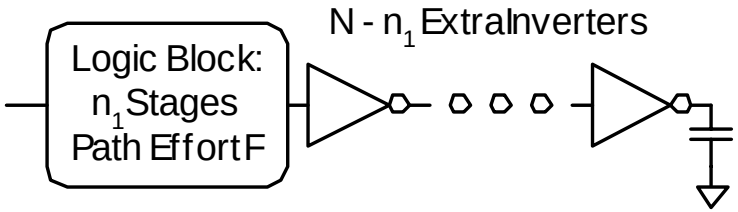
$$D = NF^{1/N} + P = N(64)^{1/N} + N$$



Best Number of Stages

Consider adding inverters at the end of the path?

How many produce the best delay?



$$D = NF^{1/N} + \sum_{i=1}^{n_1} p_i + (N - n_1)\rho_{inv}$$

$$\frac{\partial D}{\partial N} = -F^{1/N}\ln F^{1/N} + F^{1/N} + \rho_{inv} = 0$$

Define best stage effort $\rho = F^{1/N}$

$$\rho_{inv} + \rho(1 - \ln\rho) = 0$$

Best Stage Effort

$\rho_{inv} + \rho(1 - \ln\rho) = 0$ has no closed form solution

Neglecting parasitics ($\rho_{inv} = 0$), $\rho = 2.718$ (e)

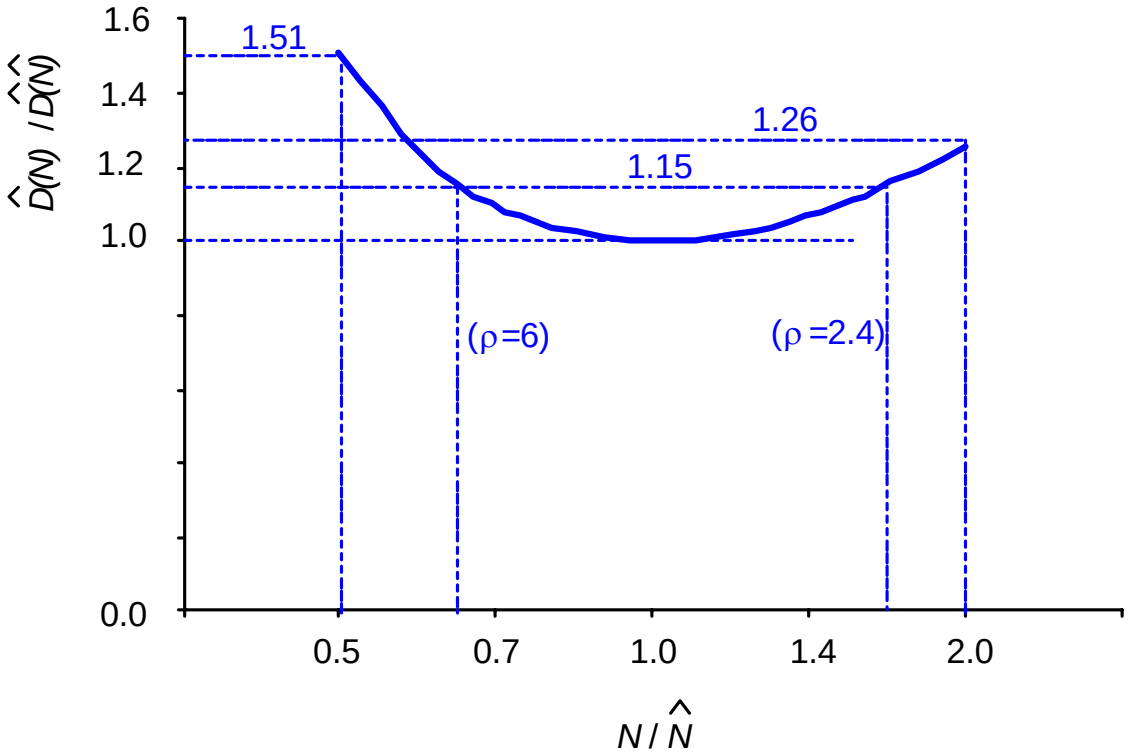
For $\rho_{inv} = 1$, solve numerically for $\rho = 3.59$

Sensitivity analysis: *How sensitive is the delay to using exactly the best number of stages?*

$2.4 < \rho < 6.0$ gives delay within 15% of optimal

$\rho = 4$ is a convenient choice

Due to the above simplification: FO4 inverter used as 'representative' logic gate delay in a particular process



Larger Example: Register File Decoder

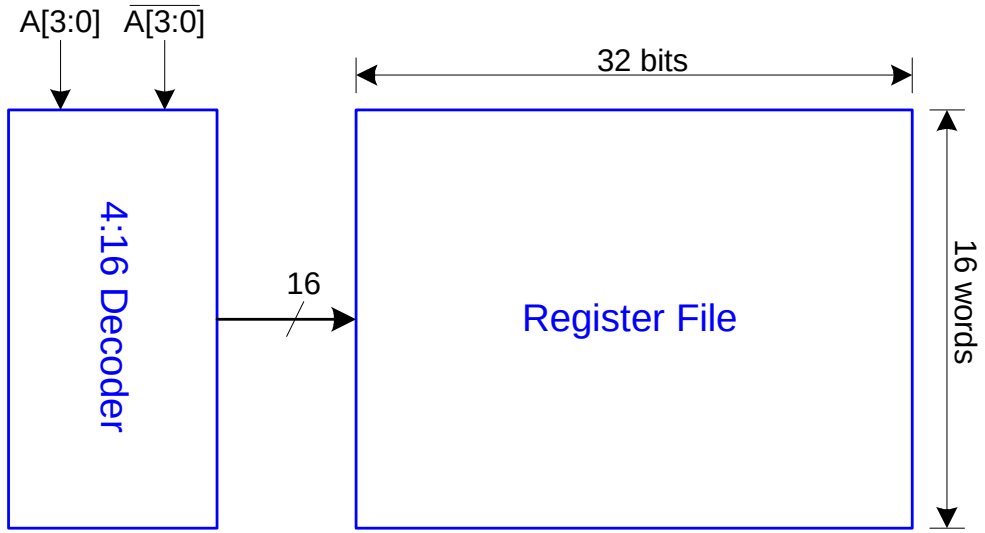
Decoder specifications

- 16 word register file
- Each word is 32 bits wide
- Each bit presents a load of 3 unit-sized transistors on the word line
- True and complimentary versions of address bits A[3:0] are available
- Each address input can drive 10 unit-sized transistors.

How many stages to use?

How large should each gate be?

How fast can the decoder operate?



Larger Example: Register File Decoder

Decoder effort is mainly electrical and branching

Electrical Effort H = (32 * 3) / 10 = 9.6

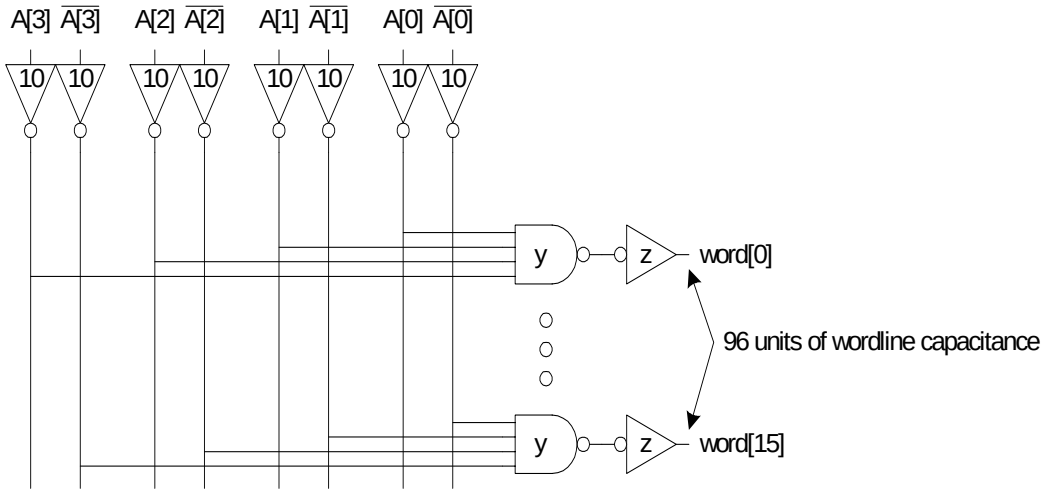
Branching Effort B = 8

If we neglect logical effort G (assume G = 1)

Path Effort F = GBH = 76.8

Number of stages = N = $\log_4 F$ = 3.1

Three stage design:



Logical Effort G = 1 * 6/3 * 1 = 2

Path Effort F= GBH = 154

Stage Effort $\hat{f} = F^{1/3} = 5.36$

Path Delay D = 3 $\hat{f}$ + 1 + 4 + 1 = 22.1

Gate sizes z = 96 * 1/ 5.36 = 18

y = 18 * 2/ 5.36 = 6.7

Larger Example: Register File Decoder

Compare alternatives with a spreadsheet

Design	N	G	P	D
NAND4 - INV	2	2	5	29.8
NAND2 - NOR2	2	20/9	4	30.1
INV - NAND4 - INV	**3**	**2**	**6**	**22.1**
NAND4 - INV - INV - INV	4	2	7	21.1
NAND2 - NOR2 - INV - INV	4	20/9	6	20.5
NAND2 - INV - NAND2 - INV	**4**	**16/9**	**6**	**19.7**
INV - NAND2 - INV - NAND2 - INV	5	16/9	7	20.4
NAND2 - INV - NAND2 - INV - INV - INV	6	16/9	8	21.6

Logical Effort: Recap of Definitions

TERM	STAGE	PATH
number of stages	1	N
logical effort	g	$G = \prod g_i$
electrical effort	C_{out} / C_{in}	$H = \frac{C_{outpath}}{C_{inpath}}$
branching effort	$b = \frac{C_{onpath} + C_{offpath}}{C_{onpath}}$	$B = \prod b_i$
effort	$f = gh$	$F = GBH$
effort delay	f	$D_F = \sum f_i$
parasitic delay	p	$P = \sum p_i$
delay	$d = f + p$	$D = \sum d_i = D_F + P$

Logical Effort: Method Recap

- Compute path effort $F = GBH$
- Estimate best number of stages $N=\log_4 F$
- Sketch path with N stages
- Estimate least delay $D = NF^{1/N} + P$
- Determine best stage effort $\hat{f} = F^{1/N}$
- Find gate sizes using capacitance transformation $\hat{f} = g\frac{C_{out}}{C_{in}}$

Logical Effort Summary

- Provides a mechanism for designing and discussing fast circuits
- NANDs are faster than NORs in CMOS
- Paths are fastest when effort delay is ~4
- Path delay is weakly sensitive to stages, sizes
- Using fewer stages doesn't mean faster circuit
- Inverters and NAND2 best for driving large loads (caps)
- BUT REQUIRES ***PRACTICE*** TO MASTER !!!

Limitations of Logical Effort

- Chicken and Egg problem
  - Need path to compute G
  - But don't know number of stages without G
- Simplistic Delay Model
  - Neglects input rise time effects and input arrival times
  - Gate-source capacitance approximation
  - Bootstrapping due to gate to drain capacitance coupling
- Ignores secondary effects: velocity saturation, body effect etc
- Does not account for interconnect
  - More applicable to datapath circuits with regular layout structure e.g. adders, mults etc
  - Iterations required in designs with significant interconnect delay
- Design for maximum speed only, no information about minimum area/power
- Paths with complex branching are difficult to analyze by hand