EE671: VLSI Design
Spring 2024/25

Laxmeesha Somappa

Department of Electrical Engineering

IIT Bombay

laxmeesha@ee.iitb.ac.in

Lecture – 30
Logical Effort

laxmeesha@ee.iitb.ac.in

Generalizing for Multi gate paths

- Summary:

Term	Stage	Path
number of stages	1	N
logical effort	g	$G = \prod g_i$
electrical effort	$h = \frac{C_{out}}{C_{in}}$	$H = \frac{C_{out\text{-}path}}{C_{in\text{-}path}}$
branching effort	$b = \frac{C_{on\text{-}path} + C_{off\text{-}path}}{C_{on\text{-}path}}$	$B = \prod b_i$
effort	$f = gh$	$F = GBH$
delay	$d = f + p$	$D = \sum d_i = D_F + P$
effort delay	f	$D_F = \sum f_i$
parasitic delay	p	$P = \sum p_i$

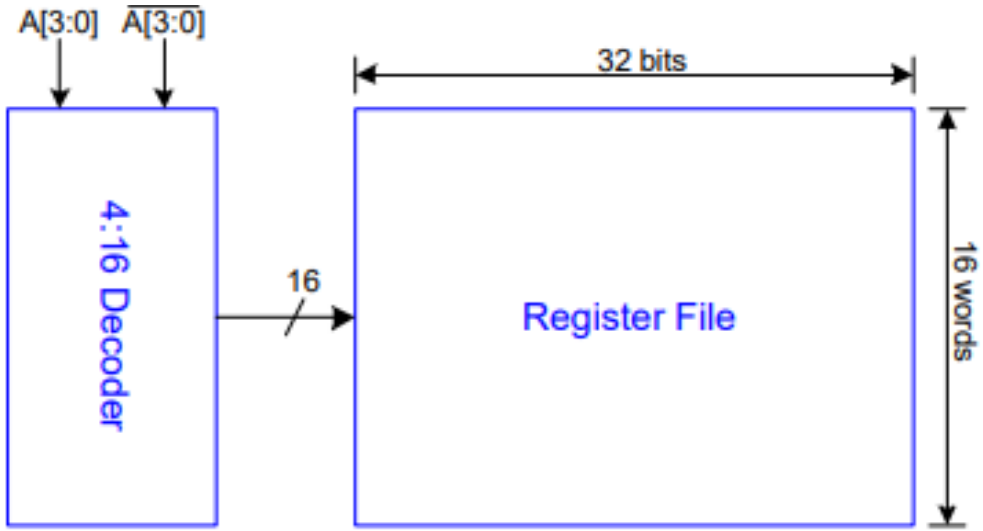
GENERALIZING FOR MULTI GATE PATHS

- Method/algorithm to size the path:

Step-1	**Compute the path effort**	**F = GBH**
Step-2	**If N is already chosen, proceed to step-3. If N is not given, the first estimate of N is calculated:** **(log base-4 since we saw that optimality is usually f_{opt} = 3.6)**	**$N = \log_4 F$**
Step-3	**Determine the optimal stage effort**	**$f_{opt} = \sqrt[N]{F}$**
Step-4	**Find individual gate caps (start from the output side)**	**$C_{in} = \frac{g\ Cout}{f_{opt}}$**
Step-5	**Based on input cap, obtain the gate strength**	
Step-6	**Estimate optimal delay – go back to Step-3 and iterate "N"**	**$D = P + NF^{\frac{1}{N}}$**

GENERALIZING FOR MULTI GATE PATHS

- Example-3: An address decoder for memory
- Need to design a 4 to 16 decoder
- Assume address (A) and the complement are available



- Decoder specifications:
  - 16 word memory
  - Each word is 32 bits (4 bytes)
  - Each bit represents a load of 3-unit sized MOS
  - Each addr input can max drive 10-unit sized MOS

- Questions:
  - How many stages of gates required?
  - What will be the size of each gate?
  - What is the optimal decoder delay?

GENERALIZING FOR MULTI GATE PATHS

- Decoder truth table:

A[3]	A[2]	A[1]	A[0]	WL[15]	WL[14]	WL[13]	WL[12]	WL[11]	WL[10]	WL[9]	WL[8]	WL[7]	WL[6]	WL[5]	WL[4]	WL[3]	WL[2]	WL[1]	WL[0]
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Generalizing for Multi gate paths

$$WL[0] = \overline{A[3] + A[2] + A[1] + A[0]}$$

$$WL[1] = \overline{A[3] + A[2] + A[1] + \overline{A[0]}}$$

$$WL[2] = \overline{A[3] + A[2] + \overline{A[1]} + A[0]}$$

$$WL[3] = \overline{A[3] + A[2] + \overline{A[1]} + \overline{A[0]}}$$

$$WL[4] = \overline{A[3] + \overline{A[2]} + A[1] + A[0]}$$

$$WL[5] = \overline{A[3] + \overline{A[2]} + A[1] + \overline{A[0]}}$$

$$WL[6] = \overline{A[3] + \overline{A[2]} + \overline{A[1]} + A[0]}$$

$$WL[7] = \overline{A[3] + \overline{A[2]} + \overline{A[1]} + \overline{A[0]}}$$

$$WL[8] = \overline{\overline{A[3]} + A[2] + A[1] + A[0]}$$

$$WL[9] = \overline{\overline{A[3]} + A[2] + A[1] + \overline{A[0]}}$$

$$WL[10] = \overline{\overline{A[3]} + A[2] + \overline{A[1]} + A[0]}$$

$$WL[11] = \overline{\overline{A[3]} + A[2] + \overline{A[1]} + \overline{A[0]}}$$

$$WL[12] = \overline{\overline{A[3]} + \overline{A[2]} + A[1] + A[0]}$$

$$WL[13] = \overline{\overline{A[3]} + \overline{A[2]} + A[1] + \overline{A[0]}}$$

$$WL[14] = \overline{\overline{A[3]} + \overline{A[2]} + \overline{A[1]} + A[0]}$$

$$WL[15] = \overline{\overline{A[3]} + \overline{A[2]} + \overline{A[1]} + \overline{A[0]}}$$

Generalizing for Multi gate paths

- Assuming $A[i]$ and $\overline{A[i]}$ are available, observe the following:
  - $A[3]$ and $\overline{A[3]}$ have 8 connections
  - $A[2]$ and $\overline{A[2]}$ have 8 connections
  - $A[1]$ and $\overline{A[1]}$ have 8 connections
  - $A[0]$ and $\overline{A[0]}$ have 8 connections

- Next question:
  - What are the different ways of implementing the logic? (using only NAND, NOR and INV)

GENERALIZING FOR MULTI GATE PATHS

- Decoder truth table (with $\overline{A[i]}$):

$\overline{A[3]}$	$\overline{A[2]}$	$\overline{A[1]}$	$\overline{A[0]}$	WL[15]	WL[14]	WL[13]	WL[12]	WL[11]	WL[10]	WL[9]	WL[8]	WL[7]	WL[6]	WL[5]	WL[4]	WL[3]	WL[2]	WL[1]	WL[0]
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Generalizing for Multi gate paths

$$WL[0] = \overline{A[3]} \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot \overline{A[0]}$$

$$WL[1] = \overline{A[3]} \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot A[0]$$

$$WL[2] = \overline{A[3]} \cdot \overline{A[2]} \cdot A[1] \cdot \overline{A[0]}$$

$$WL[3] = \overline{A[3]} \cdot \overline{A[2]} \cdot A[1] \cdot A[0]$$

$$WL[4] = \overline{A[3]} \cdot A[2] \cdot \overline{A[1]} \cdot \overline{A[0]}$$

$$WL[5] = \overline{A[3]} \cdot A[2] \cdot \overline{A[1]} \cdot A[0]$$

$$WL[6] = \overline{A[3]} \cdot A[2] \cdot A[1] \cdot \overline{A[0]}$$

$$WL[7] = \overline{A[3]} \cdot A[2] \cdot A[1] \cdot A[0]$$

$$WL[8] = A[3] \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot \overline{A[0]}$$

$$WL[9] = A[3] \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot A[0]$$

$$WL[10] = A[3] \cdot \overline{A[2]} \cdot A[1] \cdot \overline{A[0]}$$

$$WL[11] = A[3] \cdot \overline{A[2]} \cdot A[1] \cdot A[0]$$

$$WL[12] = A[3] \cdot A[2] \cdot \overline{A[1]} \cdot \overline{A[0]}$$

$$WL[13] = A[3] \cdot A[2] \cdot \overline{A[1]} \cdot A[0]$$

$$WL[14] = A[3] \cdot A[2] \cdot A[1] \cdot \overline{A[0]}$$

$$WL[15] = A[3] \cdot A[2] \cdot A[1] \cdot A[0]$$

GENERALIZING FOR MULTI GATE PATHS

$$\boldsymbol{WL[0] = \overline{A[3] + A[2] + A[1] + A[0]}} \qquad \boldsymbol{WL[0] = \overline{A[3]} \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot \overline{A[0]}}$$

- With N = 2 (two stages of gates)
  1. $\boldsymbol{WL[0] = \overline{A[3]} \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot \overline{A[0]}}$ ≡ 4AND ≡ 4NAND → INV
  2. $\boldsymbol{WL[0] = \overline{A[3]} \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot \overline{A[0]}}$ ≡ INV → 4NOR
  3. $\boldsymbol{WL[0] = \overline{A[3]} \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot \overline{A[0]}}$ ≡ 2NAND → 2NOR

GENERALIZING FOR MULTI GATE PATHS

$$\boldsymbol{WL[0] = \overline{A[3] + A[2] + A[1] + A[0]}} \qquad \boldsymbol{WL[0] = \overline{A[3]} \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot \overline{A[0]}}$$

- With N = 3 (three stages of gates)
  1. $\boldsymbol{WL[0] = \overline{A[3]} \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot \overline{A[0]}}$ ≡ INV → 4NAND → INV

GENERALIZING FOR MULTI GATE PATHS

$$WL[0] = \overline{A[3] \cdot A[2] \cdot A[1] \cdot A[0]}$$

- With N = 4
  1. $WL[0] = \overline{A[3]} \cdot \overline{A[2]} \cdot \overline{A[1]} \cdot \overline{A[0]}$ ≡ 4NAND → INV → INV → INV
  2. $WL[0] = \overline{(A[3] \cdot A[2])} + \overline{(A[1] \cdot A[0])}$ ≡ 2NAND → 2NOR → INV → INV
  3. $WL[0] = \overline{(A[3] \cdot A[2])} + \overline{(A[1] \cdot A[0])}$ ≡ 2NAND → INV → 2NAND → INV

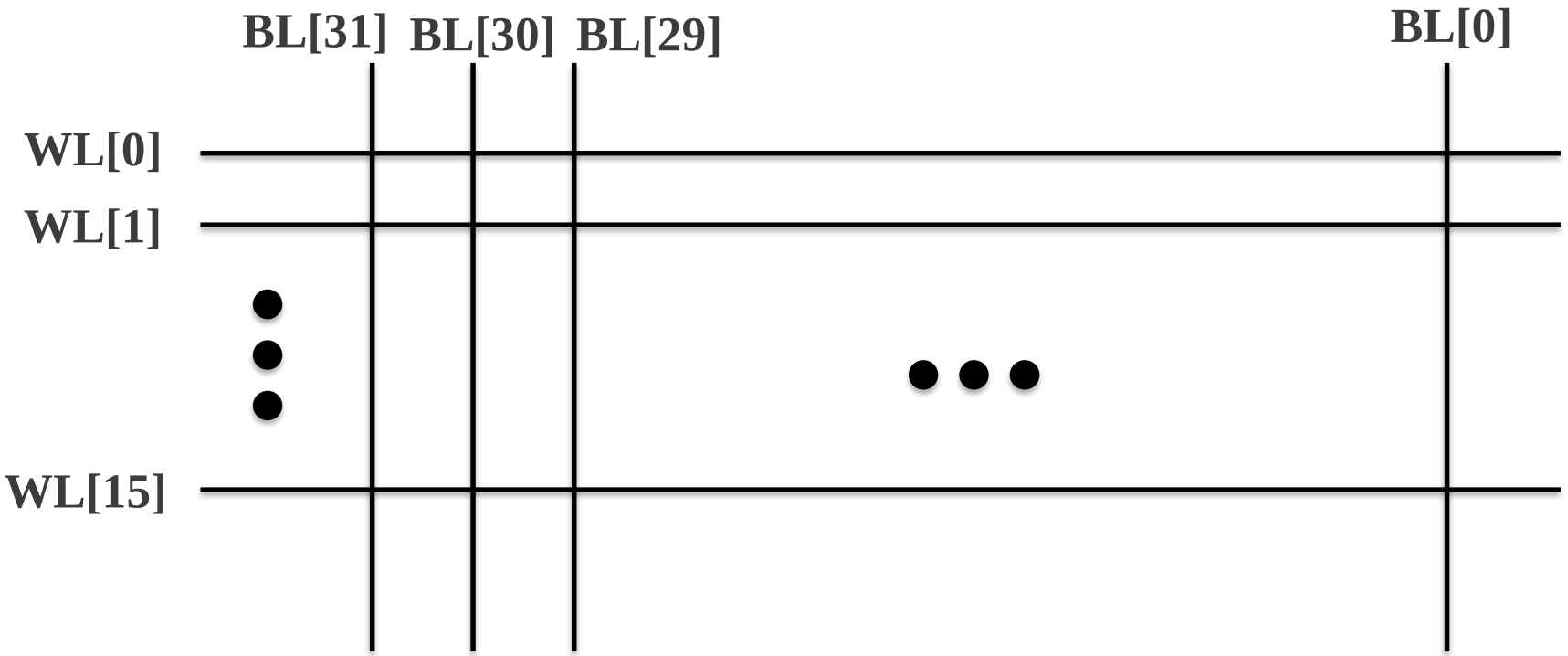
Generalizing for Multi gate paths

- Let us find the delay for these cases:
  - N = 2
    - A. 4NAND → INV
    - B. INV → 4NOR
    - C. 2NAND → 2NOR
  - N = 3
    - A. INV → 4NAND → INV
  - N = 4
    - A. 4NAND → INV → INV → INV
    - B. 2NAND → 2NOR → INV → INV
    - C. 2NAND → INV → 2NAND → INV

- Decoder specifications:
  - 16 word memory
  - Each word is 32 bits (4 bytes)
  - Each bit represents a load of 3-unit sized MOS
  - Each A input can max drive 10-unit sized MOS

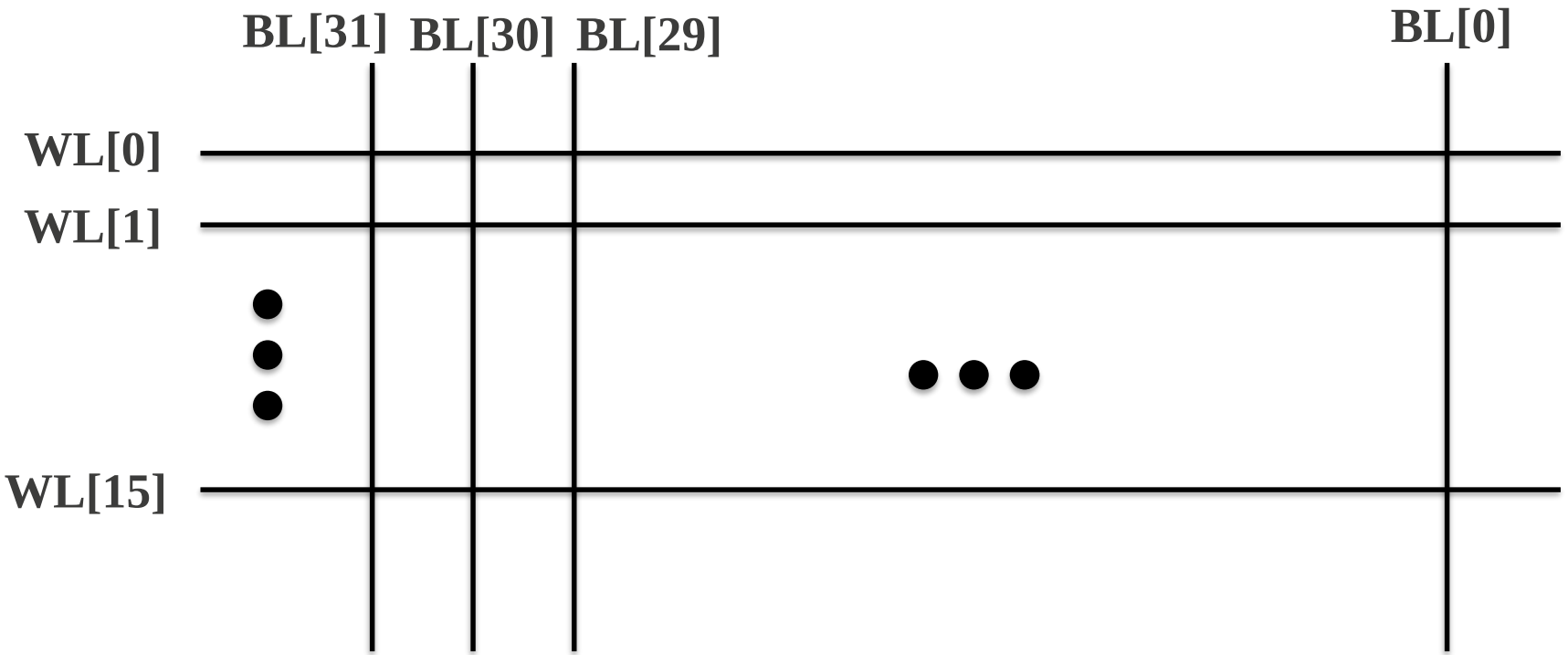
Generalizing for Multi gate paths

- Decoder specifications:
  - 16 word memory
  - Each word is 32 bits (4 bytes)
  - Each bit represents a load of 3-unit sized MOS
  - Each A input can max drive 10-unit sized MOS



Generalizing for Multi gate paths

- Load on each wordline = 3*32 = 96
- Input cap = 10
- H = 96/10
- B = 8



Generalizing for Multi gate paths

- 4NAND → INV
- H = 9.6, B = 8
- G = (6/3) * 1 = 2
- F = GBH = 153.6
- $f_{opt} = \sqrt[2]{153.6} = 12.36$
- P = 4 + 1 = 5
- D = 5 + 2*(12.36) = 29.73

- INV → 4NOR
- H = 9.6, B = 8
- G = (9/3) * 1 = 3
- F = GBH = 230.4
- $f_{opt} = \sqrt[2]{153.6} = 15.17$
- P = 1 + 4 = 5
- D = 5 + 2*(15.17) = 35.35

Generalizing for Multi gate paths

- 4NAND → INV

- H = 9.6, B = 8
- G = (6/3) * 1 = 2
- F = GBH = 153.6
- $f_{opt} = \sqrt[2]{153.6} = 12.36$
- P = 4 + 1 = 5
- D = 5 + 2*(12.36) = 29.73

- 2NAND → 2NOR

- H = 9.6, B = 8
- G = (4/3) * (5/3) = 20/9
- F = GBH = 170.7
- $f_{opt} = \sqrt[2]{153.6} = 13.06$
- P = 2 + 2 = 4
- D = 4 + 2*(13.06) = 30.12

Generalizing for Multi gate paths

- 4NAND → INV
- H = 9.6, B = 8
- G = (6/3) * 1 = 2
- F = GBH = 153.6
- $f_{opt} = \sqrt[2]{153.6} = 12.36$
- P = 4 + 1 = 5
- D = 5 + 2*(12.36) = 29.73

- INV → 4NAND → INV
- H = 9.6, B = 8
- G = 1* (6/3) * 1 = 2
- F = GBH = 153.6
- $f_{opt} = \sqrt[3]{153.6} = 5.35$
- P = 1 + 4 + 1 = 6
- D = 6 + 3*(5.35) = 22.06

Generalizing for Multi gate paths

- INV → 4NAND → INV
- H = 9.6, B = 8
- G = 1* (6/3) * 1 = 2
- F = GBH = 153.6
- $f_{opt} = \sqrt[3]{153.6} = 5.35$
- P = 1 + 4 + 1 = 6
- D = 6 + 3*(5.35) = 22.06

- 4NAND → INV → INV → INV
- H = 9.6, B = 8
- G = (6/3) * 1 * 1 * 1 = 2
- F = GBH = 153.6
- $f_{opt} = \sqrt[4]{153.6} = 3.52$
- P = 4 + 1 + 1 + 1 = 7
- D = 7 + 4*(3.52) = 21.08

Generalizing for Multi gate paths

- 4NAND → INV → INV → INV
  - H = 9.6, B = 8
  - G = (6/3) * 1 * 1 * 1 = 2
  - F = GBH = 153.6
  - $f_{opt} = \sqrt[4]{153.6} = 3.52$
  - P = 4 + 1 + 1 + 1 = 7
  - D = 7 + 4*(3.52) = 21.08

- 2NAND → 2NOR → INV → INV
  - H = 9.6, B = 8
  - G = (4/3) * (5/3) * 1 * 1 = 20/9
  - F = GBH = 170.7
  - $f_{opt} = \sqrt[4]{170.7} = 3.61$
  - P = 2 + 2 + 1 + 1 = 6
  - D = 6 + 4*(3.61) = 20.45

GENERALIZING FOR MULTI GATE PATHS

- 2NAND → 2NOR → INV → INV
- H = 9.6, B = 8
- G = (4/3) * (5/3) * 1 * 1 = 20/9
- F = GBH = 170.7
- $f_{opt} = \sqrt[4]{170.7} = 3.61$
- P = 2 + 2 + 1 + 1 = 6
- D = 6 + 4*(3.61) = 20.45

- 2NAND → INV → 2NAND → INV
- H = 9.6, B = 8
- G = (4/3) * 1 * (4/3) * 1 = 16/9
- F = GBH = 136.5
- $f_{opt} = \sqrt[4]{136.5} = 3.41$
- P = 2 + 1 + 2 + 1 = 6
- D = 6 + 4*(3.41) = 19.67

GENERALIZING FOR MULTI GATE PATHS

- To find the actual optima, you can write a script to iterate "N" and the associated logic (try the below table on pen and paper)

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

GENERALIZING FOR MULTI GATE PATHS

- 2NAND → INV → 2NAND → INV

- $f_{opt} = \sqrt[4]{136.5} = 3.41$
- Let W, X, Y, Z be caps of each stage. Recall $C_{in} = g\, C_{out}/f_{opt}$

- *Z = 1* 96/3.41 = 28.15*
- *Y = (4/3) * 28.15/3.41 = 11*
- *X = 1 * 11/3.41 = 3.22*
- *W = (4/3) * 3.22 /3.41 = 1.25 → shouldn't this be 10?*

- Decoder specifications:
  - Each A input can max drive 10-unit sized MOS
  - Logic driving the decoder can only drive 10-unit MOS
  - How many 2NAND gates connected to each address line?
    - 8 gates!! → total capacitance on each address line = 1.25 * 8 = 10