EE434
ASIC & Digital Systems

Partha Pande
School of EECS
Washington State University
pande@eecs.wsu.edu

Spring 2015

Dae Hyun Kim
daehyun@eecs.wsu.edu

Lecture 8

Interconnect

Interconnect Layers

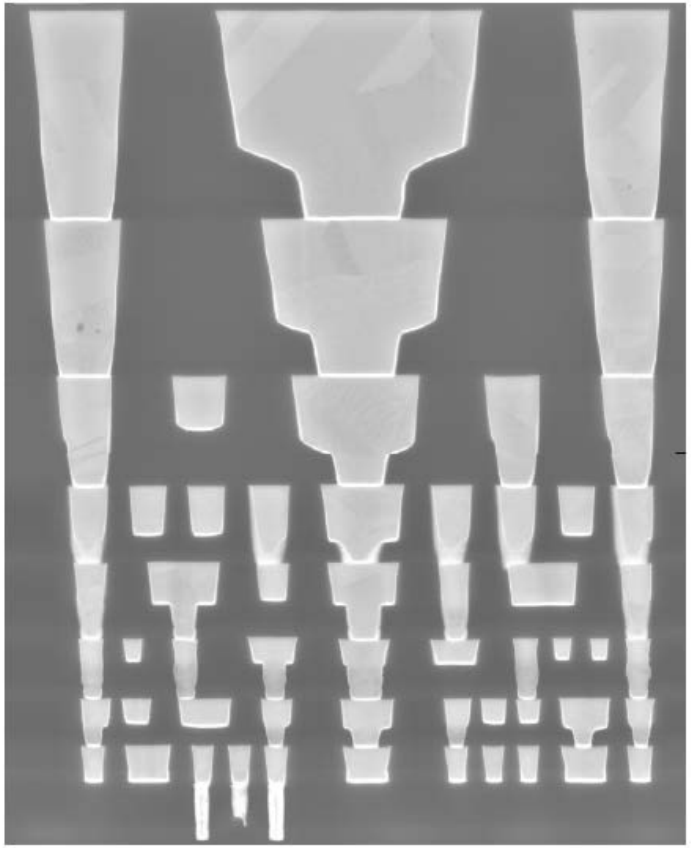


Figure 10: Cross-section: the dual damascene interconnects

Intel 65nm (IEDM'04)

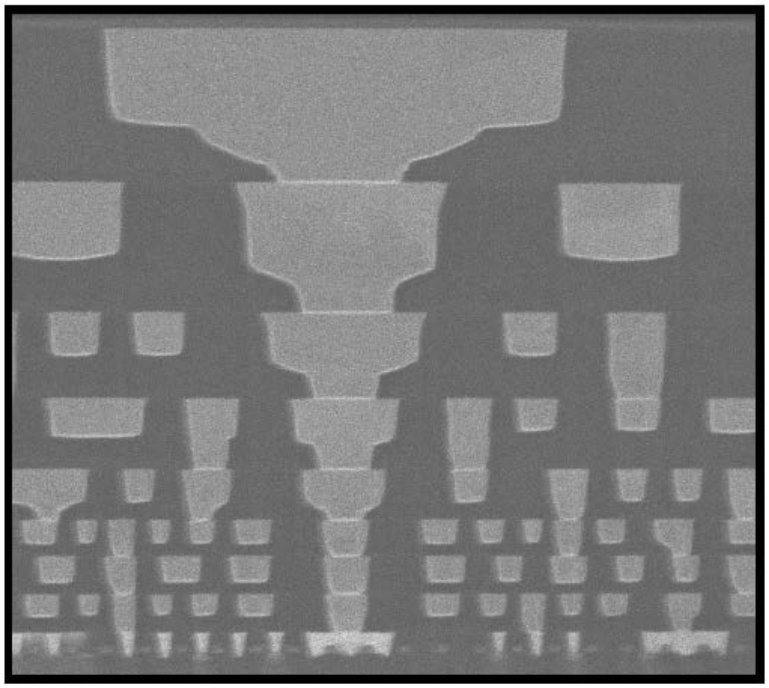


Fig.16 Cross-section of 8 of the 9 Cu interconnect layers

Intel 45nm (IEDM'07)

Interconnect Layers

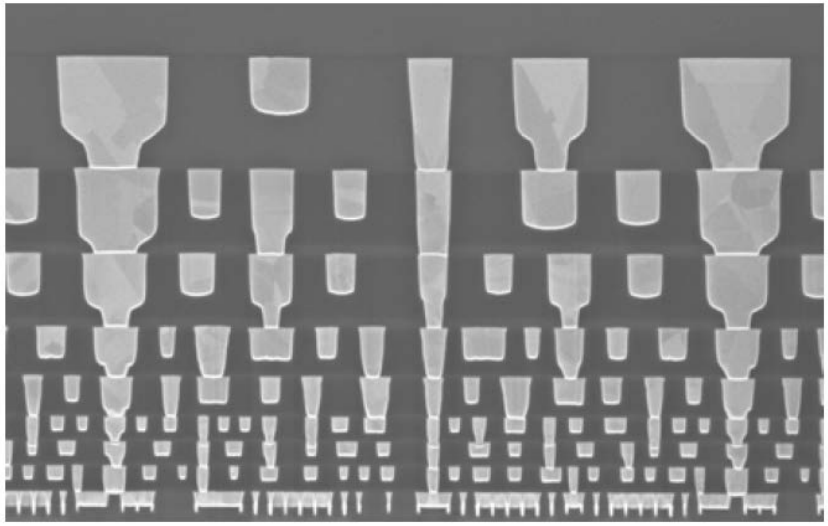


Figure 11: Cross-section of interconnect stack (8 layers

Intel 32nm (IEDM'08)

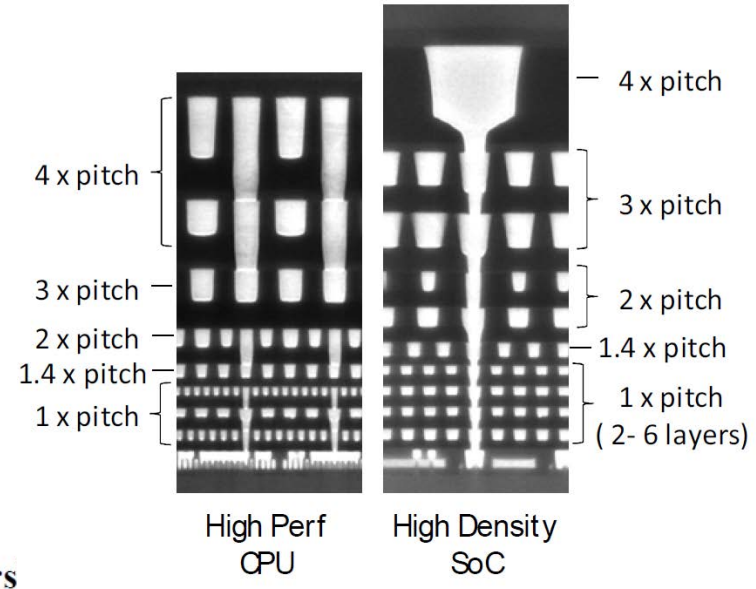


Fig.16. Interconnect architecture comparison of 22 nm CPU and SoC technologies

Intel 22nm (IEDM'12)

Interconnect Layers

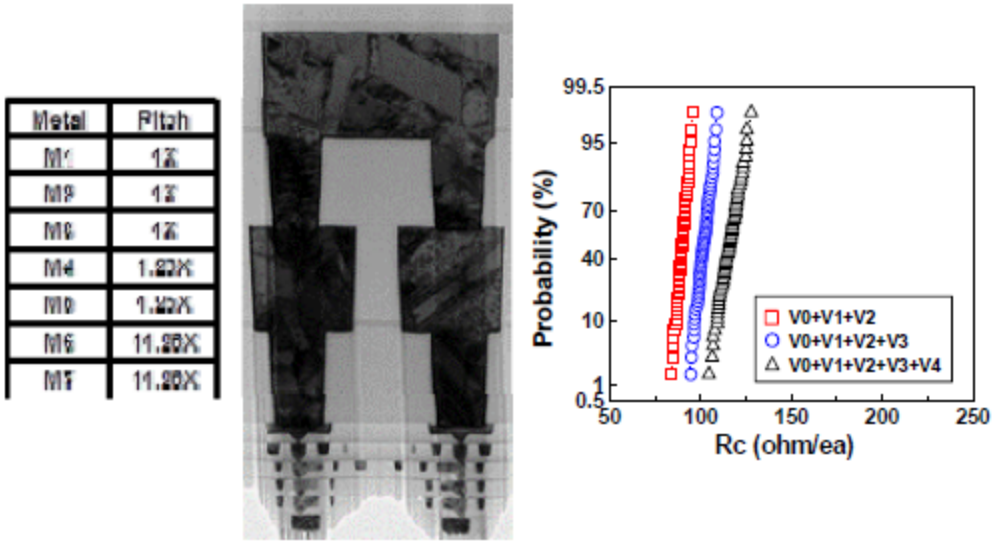


Fig. 12 (a) Cross-section view of Cu/Low for 7-level metal, (b) Cumulative distribution of stacked via R_c resistance

TSMC 16nm FinFET (IEDM'13)

Interconnect Layers

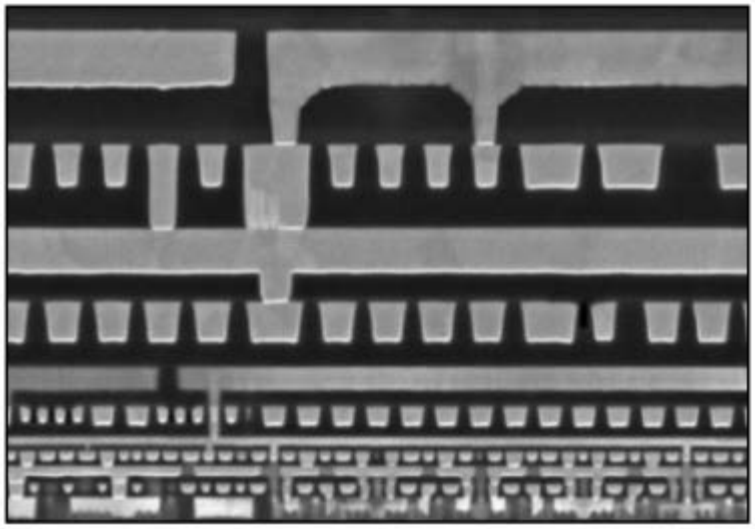


Figure 10: Interconnect Stack

Intel 14nm FinFET (IEDM'14)

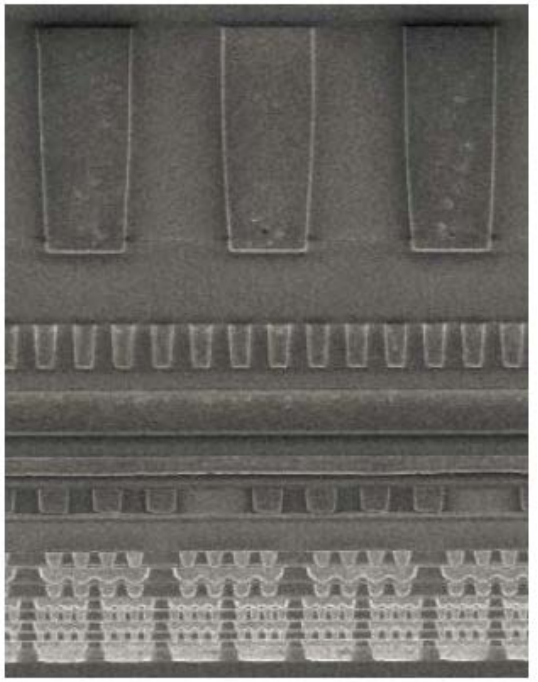


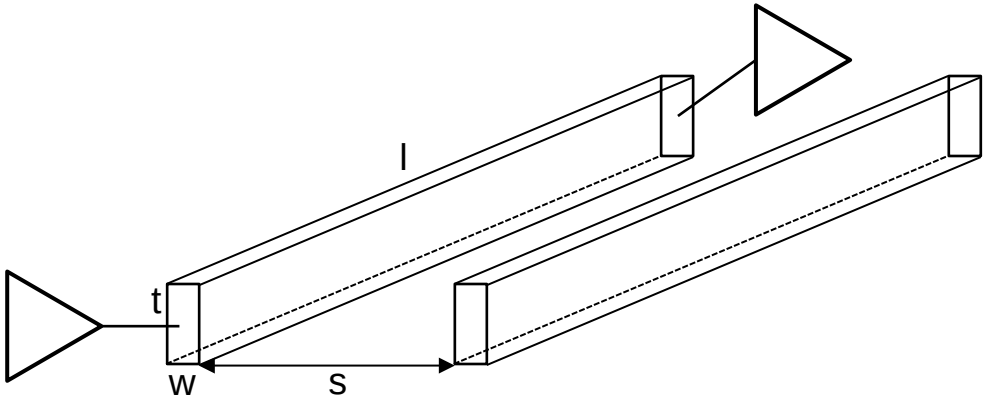
Fig. 13 Hierarchical back end reflecting 1X, 1.25X, 2X, 4X, 8X, 40X layering.

IBM 14nm FinFET (IEDM'14)

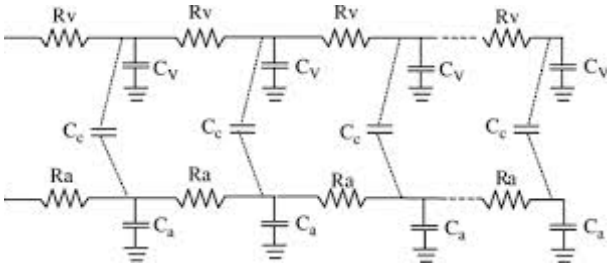
Interconnect Analysis

- Resistance
- Capacitance
- Delay calculation
- Coupling
- Buffer insertion (interconnect optimization)

Wire

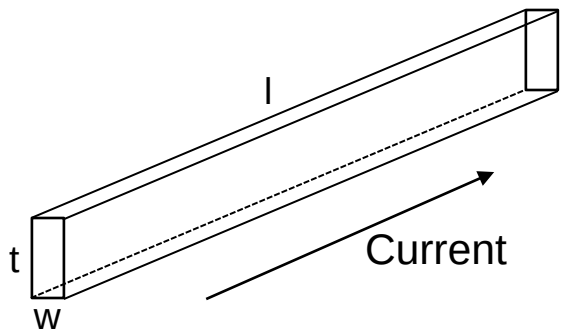


modeling



$$R = \rho \frac{l}{t \cdot w} \qquad C = \epsilon \frac{t \cdot l}{s} \qquad Delay \propto RC \propto l^2$$

Wire Resistance



$$R = \rho \frac{l}{t \cdot w}$$

ρ: Resistivity (constant)

t: Thickness (constant)

l: Wire length

w: Wire width

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

R_{sq}: sheet resistance

Example 1

$$l: 100\mu m, w: 0.065\mu m, t: 0.13\mu m, \rho = 17.1\, n\Omega \cdot m$$

$$R = (17.1) \cdot 10^{-9}(\Omega \cdot m) \cdot \frac{100 \cdot 10^{-6} m}{(0.13 \cdot 10^{-6} m) \cdot (0.065 \cdot 10^{-6} m)} = 202\Omega$$

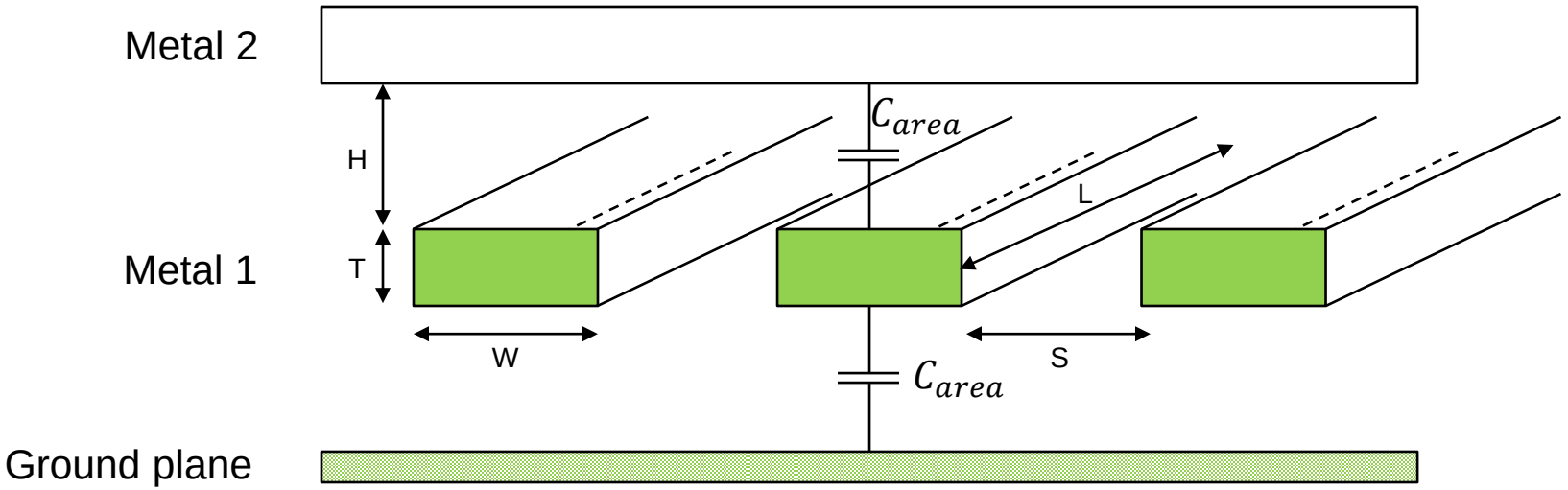
Example 2 (Nangate 45nm)

$$l: 100\mu m, w: 0.065\mu m, R_{sq} = 0.38\Omega$$

$$R = (0.38\Omega) \cdot \frac{100\mu m}{0.065\mu m} = 585\Omega$$

Wire Capacitance

- Area capacitance

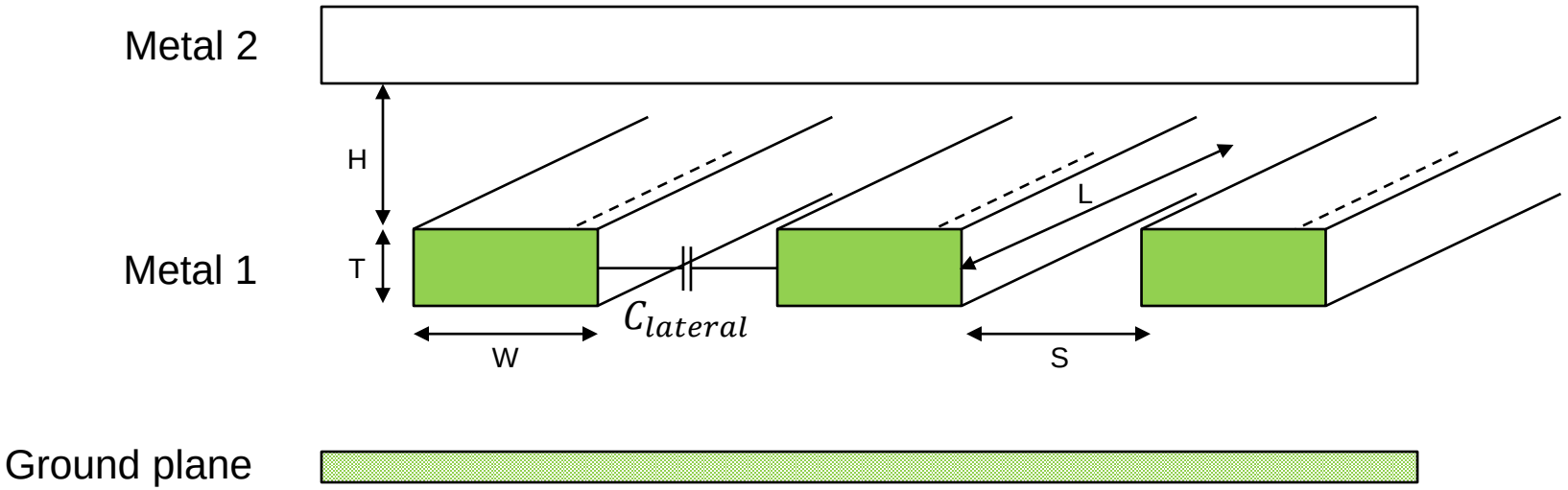


$$c_{area} = \varepsilon_{OX} \cdot \frac{W}{H}$$ ⇒ Area capacitance per unit length (F/m)

$$C_{area} = c_{area} \cdot L$$ ⇒ Total area capacitance (F)

Wire Capacitance

- Lateral capacitance

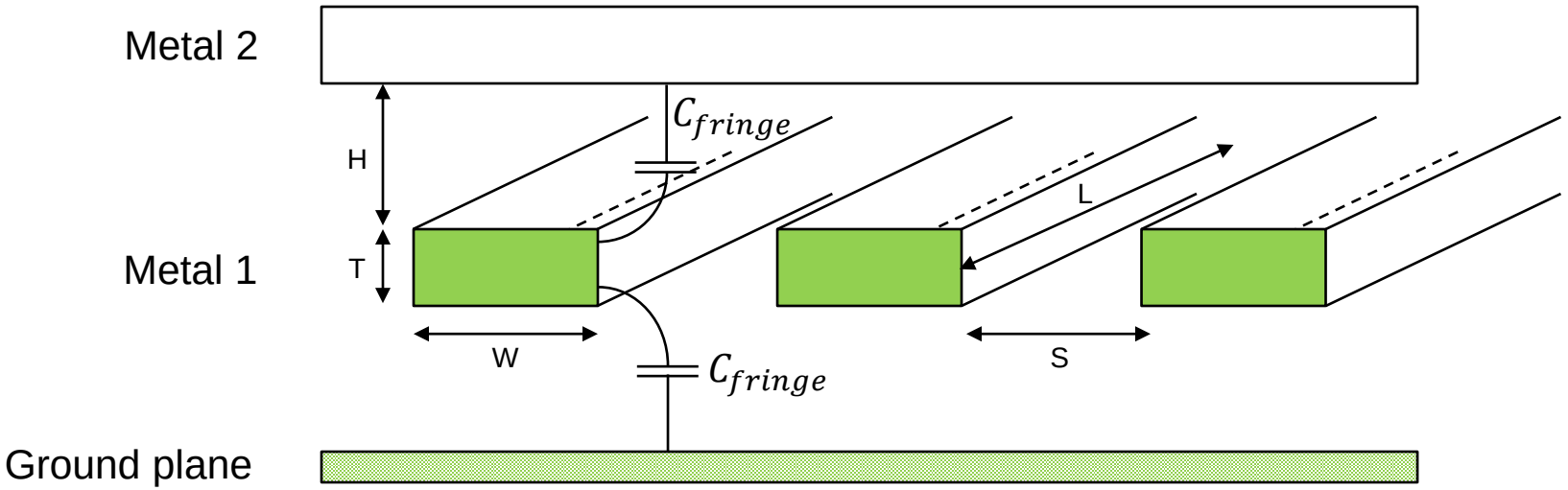


$$c_{lateral} = \varepsilon_{OX} \cdot \frac{T}{S}$$ ⇒ Lateral capacitance per unit length (F/m)

$$C_{lateral} = c_{lateral} \cdot L$$ ⇒ Total lateral capacitance (F)

Wire Capacitance

- Fringe capacitance



$$c_{fringe} = \varepsilon_{OX} \cdot \ln\left(1 + \frac{T}{H}\right)$$ ⇒ Fringe capacitance per unit length (F/m)

$$C_{fringe} = c_{fringe} \cdot L$$ ⇒ Total fringe capacitance (F)

Delay Calculation (Elmore Delay)

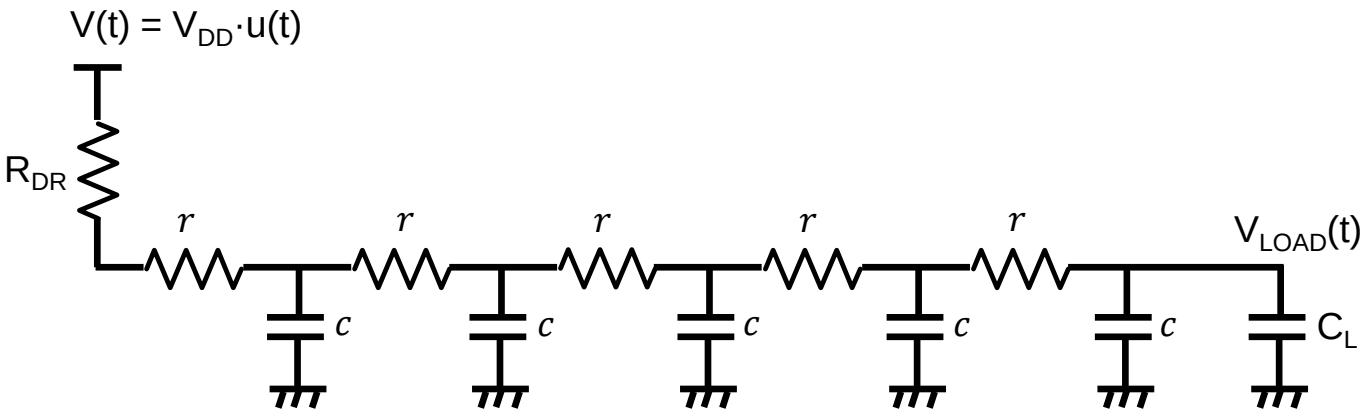
- Delay of a long wire (distributed RC network)
  - $\tau = R_{DR} * (c * n + C_L) +$

$$r(c * n + C_L) + r * (c * (n-1) + C_L) + \cdots + r * (c + C_L)$$

$$= R_{DR} * (C_{wire} + C_L) + R_{wire} \cdot C_L + rc\frac{n(n+1)}{2}$$

$$= R_{DR} * (C_{wire} + C_L) + R_{wire} \cdot C_L + \frac{R_{wire}}{n} \cdot \frac{C_{wire}}{n} \cdot \frac{n(n+1)}{2}$$

$$\approx \boxed{R_{DR} * (C_{wire} + C_L) + R_{wire} \cdot C_L + \frac{R_{wire} \cdot C_{wire}}{2}}$$



Coupling

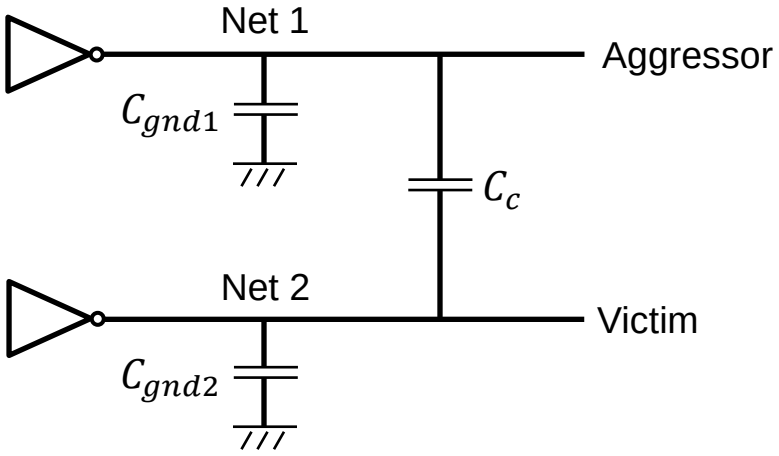
- Effects
  - Delay
  - Crosstalk

Coupling

- Delay

Victim: The net of interest.

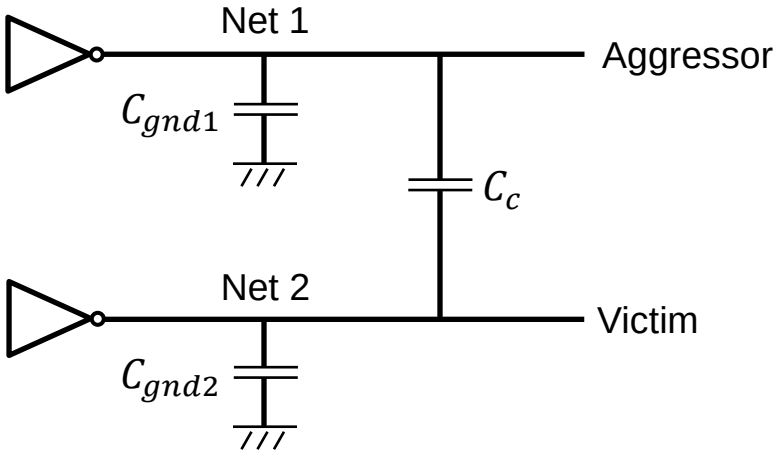
Aggressor: The neighboring nets of the victim net.

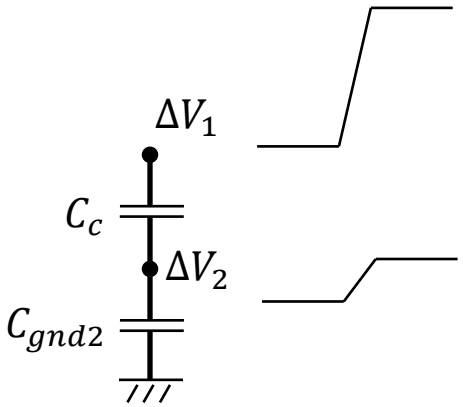


1. When the neighboring net is stationary: $C_L = C_{gnd} + C_c$
2. When the two nets are switching in the same direction: $C_L = C_{gnd}$
3. When the two nets are switching in the opposite direction: $C_L = C_{gnd} + 2C_c$

Coupling

- Crosstalk



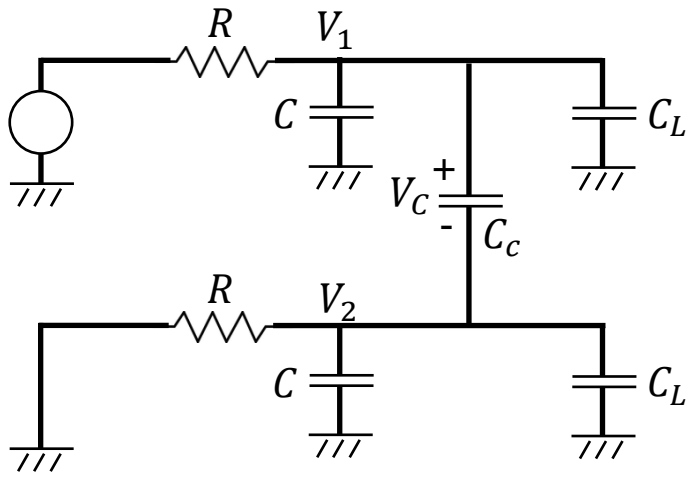


$$\Delta V_2 = \frac{C_c \cdot \Delta V_1}{(C_{gnd2} + C_c)}$$

Coupling

- Crosstalk

$V_s(t) = V_0 \cdot u(t)$



$$\tau_1 = R(C + C_L + 2C_C)$$
$$\tau_2 = R(C + C_L)$$
$$V_2(t) = \frac{V_0}{2}\left[e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}}\right]u(t)$$

Coupling

- How to reduce the coupling effect
  - Spacing

  - Shielding

Coupling

- How to reduce the coupling effect
  - Coding/Decoding

- Duan, TVLSI'09

Coding/Decoding for Coupling Minimization

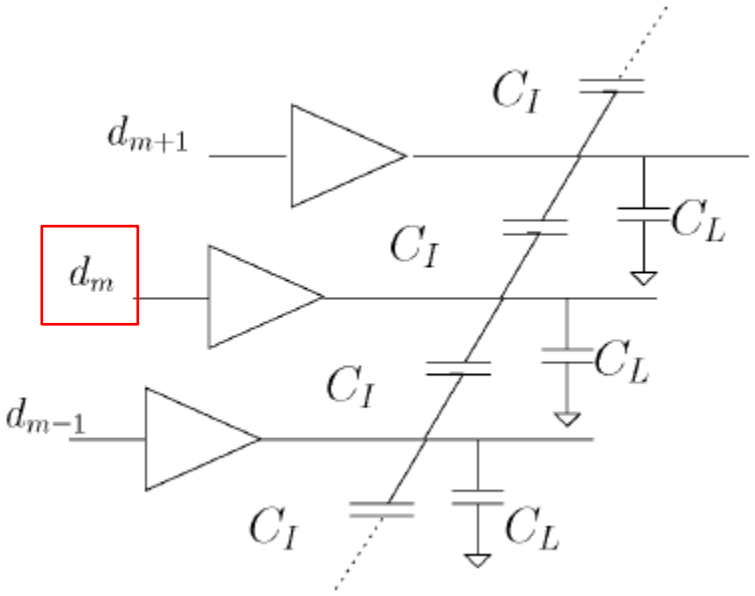


TABLE I
CLASSES OF CROSSTALK

Class	C_{eff}	Transition patterns
0C	C_L	$000 \rightarrow 111$
1C	$C_L(1+\lambda)$	$011 \rightarrow 000$
2C	$C_L(1+2\lambda)$	$010 \rightarrow 000$
3C	$C_L(1+3\lambda)$	$010 \rightarrow 100$
4C	$C_L(1+4\lambda)$	$010 \rightarrow 101$

Eliminate these patterns.

$$\lambda = \frac{C_I}{C_L}$$

Coding/Decoding for Coupling Minimization

- Forbidden Pattern Based Crosstalk Avoidance
  - Forbidden patterns
    - “101”
    - “010”
  - Forbidden pattern free (FPF) code
    - 1101110: not FPF
    - 1100110: FPF

- “If a bus contains FPF codes only, the bus will experience maximum crosstalk of no greater than 2C”.

Coding/Decoding for Coupling Minimization

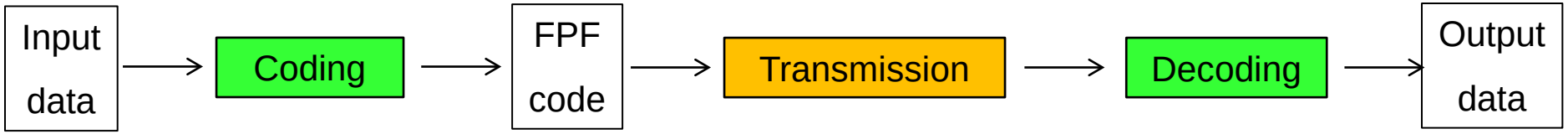
- Forbidden Pattern Free (FPF)-Crosstalk Avoidance Code (CAC) FPF-CAC

TABLE II
FPF-CAC CODEWORDS FOR 2-, 3-, 4-, AND 5 BIT BUSSES

2-bit	3-bits	4-bits	5 bit	
00	000	0000	00000	10000
01	001	0001	00001	10001
10	011	0011	00011	10011
11	100	0110	00110	11000
	110	0111	00111	11001
	111	1000	01100	11100
		1001	01110	11110
		1100	01111	11111
		1110		
		1111		

Coding/Decoding for Coupling Minimization

- Coding/Transmission/Decoding



Coding/Decoding for Coupling Minimization

Algorithm 2 Near-Optimal FPF-CAC Encoder

FPF-CAC(v)
\\ MSB stage:
if $v \geq f_{m+1}$ **then**
  $d_m = 1$;
  $r_m = v - f_m$;
else
  $d_m = 0$;
  $r_m = v$;
end if
\\ other stages:
for k = m − 1 to 2 **do**
  if $r_{k+1} \geq f_{k+1}$ **then**
    $d_k = 1$;
  else if $r_{k+1} < f_k$ **then**
    $d_k = 0$;
  else
    $d_k = d_{k+1}$;
  end if
  $r_k = r_{k+1} - f_k \cdot d_k$;
end for
LSB
$d_1 = r_2$;
return $(d_m d_{m-1} \ldots d_1)$;

Input	**CODE-1**					
Decimal value	f_6 8	f_5 5	f_4 3	f_3 2	f_2 1	f_1 1
20*	1	1	1	1	1	1
19*	1	1	1	1	1	0
18*	1	1	1	1	0	0
17*	1	1	1	0	0	1
16*	1	1	1	0	0	0
15	1	1	0	0	1	1
14	1	1	0	0	0	1
13	1	1	0	0	0	0
12	0	1	1	1	1	1
11	0	1	1	1	1	0
10	0	1	1	1	0	0
9	0	1	1	0	0	1
8	0	1	1	0	0	0
7	0	0	1	1	1	1
6	0	0	1	1	1	0
5	0	0	1	1	0	0
4	0	0	0	1	1	1
3	0	0	0	1	1	0
2	0	0	0	0	1	1
1	0	0	0	0	0	1
0	0	0	0	0	0	0

f_k: Fibonacci number

Coding/Decoding for Coupling Minimization

Algorithm 2 Near-Optimal FPF-CAC Encoder

FPF-CAC(v)
\\ MSB stage:
if $v \geq f_{m+1}$ **then**
  $d_m = 1$;
  $r_m = v - f_m$;
else
  $d_m = 0$;
  $r_m = v$;
end if
\\ other stages:
for k = m − 1 to 2 **do**
  if $r_{k+1} \geq f_{k+1}$ **then**
    $d_k = 1$;
  else if $r_{k+1} < f_k$ **then**
    $d_k = 0$;
  else
    $d_k = d_{k+1}$;
  end if
  $r_k = r_{k+1} - f_k \cdot d_k$;
end for
LSB
$d_1 = r_2$;
return $(d_m d_{m-1} \ldots d_1)$;

Input	CODE-1					
Decimal value	f_6 8	f_5 5	f_4 3	f_3 2	f_2 1	f_1 1
20*	1	1	1	1	1	1
19*	1	1	1	1	1	0
18*	1	1	1	1	0	0
17*	1	1	1	0	0	1
16*	1	1	1	0	0	0
15	1	1	0	0	1	1
14	1	1	0	0	0	1
13	1	1	0	0	0	0
12	0	1	1	1	1	1
11	0	1	1	1	1	0
10	0	1	1	1	0	0
9	0	1	1	0	0	1
8	0	1	1	0	0	0
7	0	0	1	1	1	1
6	0	0	1	1	1	0
5	0	0	1	1	0	0
4	0	0	0	1	1	1
3	0	0	0	1	1	0
2	0	0	0	0	1	1
1	0	0	0	0	0	1
0	0	0	0	0	0	0

Example (m=6)

1) v=0

$v < f_7$ (=13)

=> d_6=0, r_6=0

2) k=5 (for)

r_6 (=0) ≥ f_6 (=8) => false

r_6 (=0) < f_5 (=5) => true

=> d_5=0

$r_5 = r_6 - f_5 * d_5 = 0 - 5*0 = 0$

3) k=4

r_5 (=0) < f_4 (=3)

=> d_4=0

$r_4 = r_5 - f_4 * d_4 = 0$

4) k=3: d_3=0, r_3=0

5) k=2: d_2=0, r_2=0

6) d_1=r_2=0

0 => (000000)

Coding/Decoding for Coupling Minimization

Algorithm 2 Near-Optimal FPF-CAC Encoder

FPF-CAC(v)
\\ MSB stage:
if $v \geq f_{m+1}$ **then**
  $d_m = 1$;
  $r_m = v - f_m$;
else
  $d_m = 0$;
  $r_m = v$;
end if
\\ other stages:
for k = m − 1 to 2 **do**
  if $r_{k+1} \geq f_{k+1}$ **then**
    $d_k = 1$;
  else if $r_{k+1} < f_k$ **then**
    $d_k = 0$;
  else
    $d_k = d_{k+1}$;
  end if
  $r_k = r_{k+1} - f_k \cdot d_k$;
end for
LSB
$d_1 = r_2$;
return $(d_m d_{m-1} \ldots d_1)$;

Input	CODE-1					
Decimal value	f_6 8	f_5 5	f_4 3	f_3 2	f_2 1	f_1 1
20*	1	1	1	1	1	1
19*	1	1	1	1	1	0
18*	1	1	1	1	0	0
17*	1	1	1	0	0	1
16*	1	1	1	0	0	0
15	1	1	0	0	1	1
14	1	1	0	0	0	1
13	1	1	0	0	0	0
12	0	1	1	1	1	1
11	0	1	1	1	1	0
10	0	1	1	1	0	0
9	0	1	1	0	0	1
8	0	1	1	0	0	0
7	0	0	1	1	1	1
6	0	0	1	1	1	0
5	0	0	1	1	0	0
4	0	0	0	1	1	1
3	0	0	0	1	1	0
2	0	0	0	0	1	1
1	0	0	0	0	0	1
0	0	0	0	0	0	0

Example (m=6)

1) v=15

$v > f_7$ (=13)

=> $d_6 = 1$, $r_6 = 7$

2) k=5 (for)

$d_5 = d_6 = 1$

$r_5 = r_6 - f_5 * d_5 = 7 - 5*1 = 2$

3) k=4

$d_4 = 0$

$r_4 = r_5 - f_4 * d_4 = 2$

4) k=3

$d_3 = 0$

$r_3 = r_4 - f_3 * d_3 = 2$

5) k=2

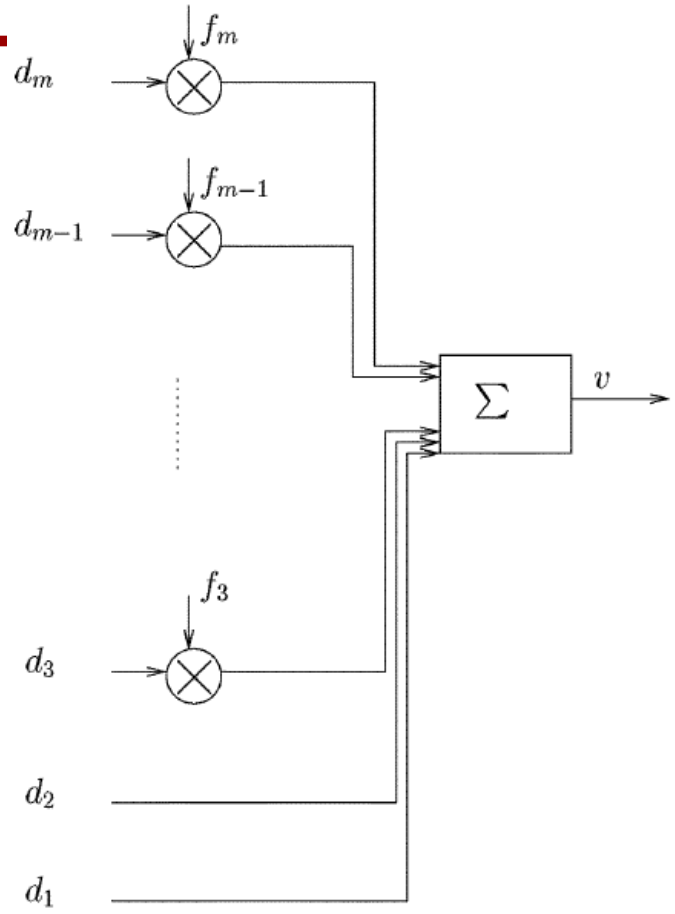
$d_2 = 1$

$r_2 = r_3 - f_2 * d_2 = 2-1 = 1$

6) $d_1 = r_2 = 1$

15 => (110011)

Coding/Decoding for Coupling Minimization



Decoder

Input	CODE-1						
Decimal value	f_6 8	f_5 5	f_4 3	f_3 2	f_2 1	f_1 1	
20*	1	1	1	1	1	1	8+5+3+2+1+1 = (10100)
19*	1	1	1	1	1	0	8+5+3+2+1+0 = (10011)
18*	1	1	1	1	0	0	8+5+3+2+0+0 = (10010)
17*	1	1	1	0	0	1	8+5+3+0+0+1 = (10001)
16*	1	1	1	0	0	0	8+5+3+0+0+0 = (10000)
15	1	1	0	0	1	1	8+5+0+0+1+1 = (01111)
14	1	1	0	0	0	1	8+5+0+0+0+1 = (01110)
13	1	1	0	0	0	0	8+5+0+0+0+0 = (01101)
12	0	1	1	1	1	1	0+5+3+2+1+1 = (01100)
11	0	1	1	1	1	0	0+5+3+2+1+0 = (01011)
10	0	1	1	1	0	0	0+5+3+2+0+0 = (01010)
9	0	1	1	0	0	1	0+5+3+0+0+1 = (01001)
8	0	1	1	0	0	0	0+5+3+0+0+0 = (01000)
7	0	0	1	1	1	1	0+0+3+2+1+1 = (00111)
6	0	0	1	1	1	0	0+0+3+2+1+0 = (00110)
5	0	0	1	1	0	0	0+0+3+2+0+0 = (00101)
4	0	0	0	1	1	1	0+0+0+2+1+1 = (00100)
3	0	0	0	1	1	0	0+0+0+2+1+0 = (00011)
2	0	0	0	0	1	1	0+0+0+0+1+1 = (00010)
1	0	0	0	0	0	1	0+0+0+0+0+1 = (00001)
0	0	0	0	0	0	0	0+0+0+0+0+0 = (00000)

Coding/Decoding for Coupling Minimization

TABLE VI
POWER CONSUMPTION COMPARISON BETWEEN CODED AND UNCODED BUSSES

bus size	E_{norm} **uncoded**	E_{norm} **coded**	**saving**
8	35075	27825	20.7 %
12	55004	44222	19.6 %
16	74999	61456	18.6 %
32	154191	124148	19.5 %

Coding/Decoding for Coupling Minimization

- Comparison

Input	CODE-1						
Decimal value	f_6 8	f_5 5	f_4 3	f_3 2	f_2 1	f_1 1	
20*	1	1	1	1	1	1	(10100)
19*	1	1	1	1	1	0	(10011)
18*	1	1	1	1	0	0	(10010)
17*	1	1	1	0	0	1	(10001)
16*	1	1	1	0	0	0	(10000)
15	1	1	0	0	1	1	(01111)
14	1	1	0	0	0	1	(01110)
13	1	1	0	0	0	0	(01101)
12	0	1	1	1	1	1	(01100)
11	0	1	1	1	1	0	(01011)
10	0	1	1	1	0	0	(01010)
9	0	1	1	0	0	1	(01001)
8	0	1	1	0	0	0	(01000)
7	0	0	1	1	1	1	(00111)
6	0	0	1	1	1	0	(00110)
5	0	0	1	1	0	0	(00101)
4	0	0	0	1	1	1	(00100)
3	0	0	0	1	1	0	(00011)
2	0	0	0	0	1	1	(00010)
1	0	0	0	0	0	1	(00001)
0	0	0	0	0	0	0	(00000)

Example

1) 20 → 8

Non-FPF: (10100) → (01000) : $(2+4+3+0+0)C_l = 9C_l$

FPF: (111111) → (011000) : $(1+0+0+1+0+0)C_l = 2C_l$

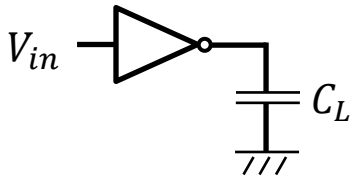
2) 21 → 10

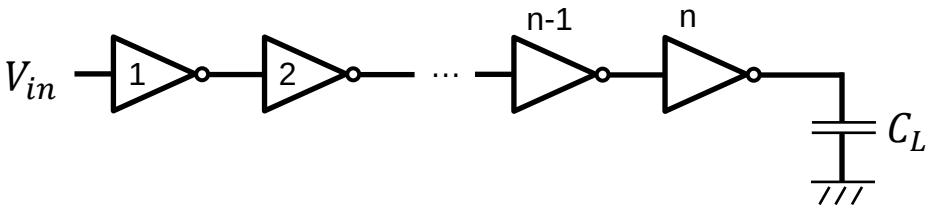
Non-FPF: (10101) → (01010) : $(2+4+4+4+2)C_l = 16C_l$

FPF: (1100000) → (0011100) : $(0+2+2+0+1+0+0)C_l = 5C_l$

Buffer Insertion (1)

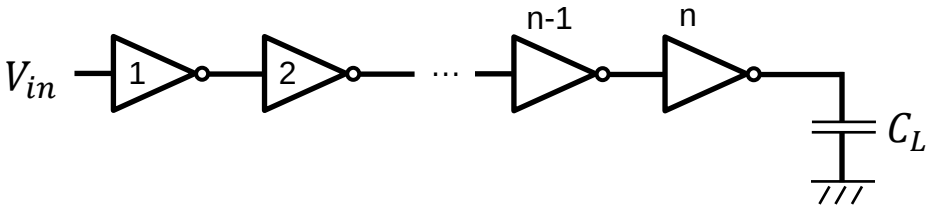
- Delay minimization for driving a large load





$$\left(\frac{W}{L}\right)_{\alpha} = S_{\alpha}\left(\frac{W}{L}\right)_{1}$$

Buffer Insertion (1)



$$C_{in,1} = C_{OX}\left[(WL)_{n1} + (WL)_{p1}\right]$$
$$C_{out,1} = C_{GD,n1} + C_{GD,p1} + C_{SB,n1} + C_{SB,p1}$$
$$R_1 \approx \frac{1}{k\left(\frac{W}{L}\right)_1 (V_{DD} - V_T)}$$
$$R_\alpha = \frac{R_1}{S_\alpha}\,, C_{in,\alpha} = S_\alpha C_{in,1}\,, C_{out,\alpha} \approx S_\alpha C_{out,1}$$

$$\tau_\alpha = R_\alpha\left[C_{out,\alpha} + C_{in,\alpha+1}\right] = \frac{R_1}{S_\alpha}\left[S_\alpha C_{out,1} + S_{\alpha+1} C_{in,1}\right]$$
$$\tau_{all} = \sum_{\alpha=1}^{n} R_1\left[C_{out,1} + \frac{S_{\alpha+1}}{S_\alpha} C_{in,1}\right]$$

Buffer Insertion (1)

$$\tau_{all} = \sum_{\alpha=1}^{n} R_1 \left[C_{out,1} + \frac{S_{\alpha+1}}{S_\alpha} C_{in,1} \right]$$

$$\frac{\partial \tau_{all}}{\partial S_1} = 0, \ldots, \frac{\partial \tau_{all}}{\partial S_n} = 0$$

$$\frac{S_\alpha}{S_{\alpha-1}} = \frac{S_{\alpha+1}}{S_\alpha} = K$$

$$S_1 = 1 \, , S_{n+1} = \frac{C_L}{C_1}$$

$$\frac{S_2}{S_1} \cdot \frac{S_3}{S_2} \cdot \ldots \cdot \frac{S_{n+1}}{S_n} = \frac{S_{n+1}}{S_1} = K^n$$

$$K = \frac{S_{\alpha+1}}{S_\alpha} = \left(\frac{C_L}{C_1} \right)^{1/n}$$

$$S_1 = 1 \, , S_2 = K \, , S_3 = K^2 \, , \ldots , S_n = K^{n-1}$$

$$\tau_{all} = \sum_{\alpha=1}^{n} R_1 \left[C_{out,1} + K C_{in,1} \right] = n R_1 \left[C_{out,1} + K C_{in,1} \right] = n R_1 \left[C_{out,1} + \left(\frac{C_L}{C_1} \right)^{1/n} C_{in,1} \right]$$

Buffer Insertion (1)

$$\tau_{all} = nR_1\left[C_{out,1} + \left(\frac{C_L}{C_1}\right)^{1/n} C_{in,1}\right]$$

$$\frac{d\tau_{all}}{dn} = R_1\left[C_{out,1} + \left(\frac{C_L}{C_1}\right)^{1/n} C_{in,1}\right] + nR_1 C_{in,1}\left(\frac{C_L}{C_1}\right)^{1/n}\left(-\frac{1}{n^2}\right) ln\left(\frac{C_L}{C_1}\right) = 0$$

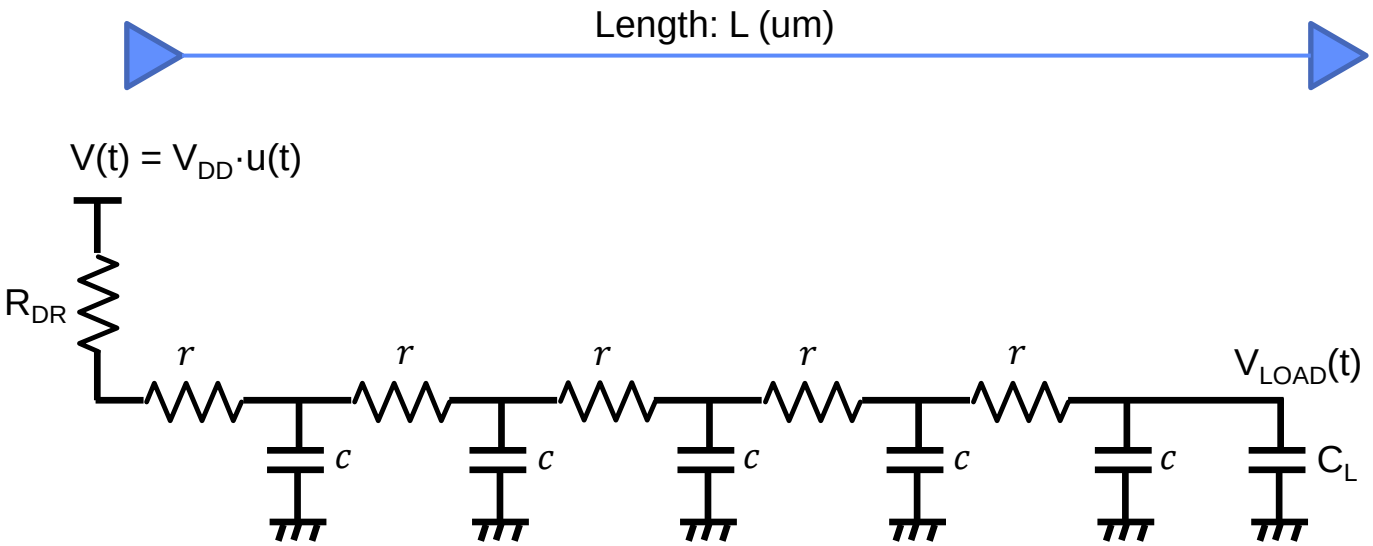
$$C_{out,1} + \left(\frac{C_L}{C_1}\right)^{1/n} C_{in,1} = C_{in,1}\left(\frac{C_L}{C_1}\right)^{1/n}\left(\frac{1}{n}\right) ln\left(\frac{C_L}{C_1}\right)$$

If $C_{out,1}$ is small => $n \approx ln\left(\frac{C_L}{C_1}\right)$

$$K = \left(\frac{C_L}{C_1}\right)^{1/n} = e$$

Buffer Insertion (2)

- Delay minimization



$$\tau \approx R_{DR} * (C_{wire} + C_L) + R_{wire} \cdot C_L + \frac{R_{wire} \cdot C_{wire}}{2}$$

Buffer Insertion (2)

- Insert a single type of buffers.

N-1 buffers

$$\tau_k = R_{DR} * \left(\frac{C_{wire}}{\frac{L}{s_k}} + C_{in}\right) + \left(\frac{R_{wire}}{\frac{L}{s_k}}\right) \cdot C_{in} + \frac{1}{2}\left(\frac{R_{wire}}{\frac{L}{s_k}} \cdot \frac{C_{wire}}{\frac{L}{s_k}}\right)$$

$$= R_{DR} * \left(\frac{C_{wire}}{L} s_k + C_{in}\right) + \left(\frac{R_{wire}}{L} s_k\right) \cdot C_{in} + \frac{1}{2}\left(\frac{R_{wire}}{L} s_k \cdot \frac{C_{wire}}{L} s_k\right)$$

$$\tau_{all} = \sum_{n=1}^{N} \tau_n = R_{DR} \cdot C_{wire} \cdot \frac{s_1 + \cdots + s_N}{L} + N \cdot R_{DR} \cdot C_{in}$$

$$+ R_{wire} \cdot C_{in} \cdot \frac{s_1 + \cdots + s_N}{L}$$

$$+ \frac{R_{wire} \cdot C_{wire}}{2L^2}\left({s_1}^2 + \cdots + {s_N}^2\right)$$

$$= R_{DR} \cdot (C_{wire} + N \cdot C_{in}) + R_{wire} \cdot C_{in} + \frac{R_{wire} \cdot C_{wire}}{2L^2}\left({s_1}^2 + \cdots + {s_N}^2\right)$$

Buffer Insertion (2)

$$\tau_{all} = \underbrace{R_{DR} \cdot (C_{wire} + N \cdot C_{in}) + R_{wire} \cdot C_{in}}_{\text{Constant}} + \frac{R_{wire} \cdot C_{wire}}{2L^2}(s_1{}^2 + \cdots + s_N{}^2)$$

Minimize $T(s_1, \ldots, s_N) = s_1{}^2 + \cdots + s_N{}^2$

subject to $s_1 + \cdots + s_N = L$

$$\frac{\partial T}{\partial s_k} = 2 \cdot s_k + 2 \cdot s_N \cdot (-1) = 0$$

$$s_k = s_N$$

$$\therefore s_1 = s_2 = \cdots = s_N$$

$$\tau_{all} = R_{DR} \cdot (C_{wire} + N \cdot C_{in}) + R_{wire} \cdot C_{in} + \frac{R_{wire} \cdot C_{wire}}{2N}$$

Buffer Insertion (2)

- Optimal N

N-1 buffers

$$\tau_{all} = R_{DR} \cdot (C_{wire} + N \cdot C_{in}) + R_{wire} \cdot C_{in} + \frac{R_{wire} \cdot C_{wire}}{2N}$$

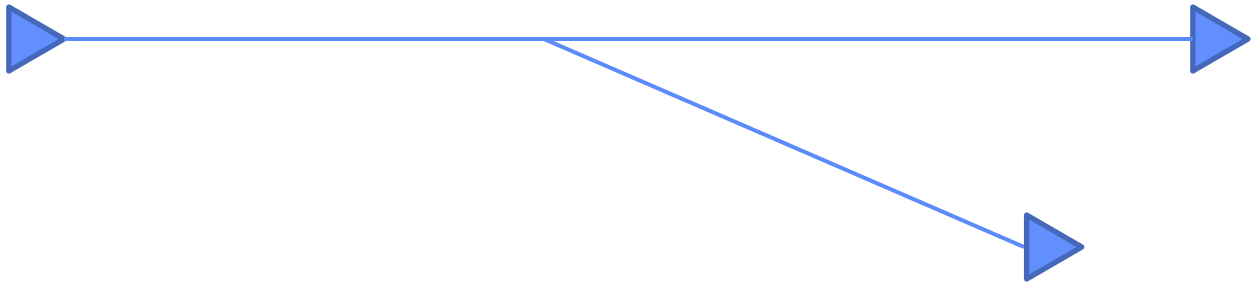
$$\frac{\partial \tau_{all}}{\partial N} = R_{DR} \cdot C_{in} - \frac{R_{wire} \cdot C_{wire}}{2N^2} = 0$$

$$N = \sqrt{\frac{R_{wire} \cdot C_{wire}}{2 \cdot R_{DR} \cdot C_{in}}}$$

Buffer Insertion

- Insert multiple types of buffers.

- Branches



- Blockages
- …

- This will be studied in EE582 in Fall 2015.