VLSI MINI PROJECT

BCD TO EXCESS 3 CODE CONVERTER

SUBMITTED BY:
AGNISH JAIN
(M.E. Microelectronics-1st Yr.)
SR.NO.4610-411-021-02172

HEMANT H PARATE
(M.E Microelectronics-1st Yr.)
SR.NO.4610-411-021-02195

Guided By:
Dr. Navakant Bhat
Dr. H.S. Jamdagni
I.I.Sc, Bangalore.

BCD to EXCESS-3 CODE Converter

Design Implementation

- **STATIC CMOS LOGIC**

Parameters of Technology Node.

- **Transistor Count –38 (without Buffers including ERROR signal)**
  -42(with buffers)
- **Power Dissipation** = 7.5256e-011 W
- **Propagation Delay TPHL=2.035 nsec. (worst case)**
- **Area**
  - **Active=10325 sq. units**
  - **Total=52000 sq. units**

Additional Resources- More no. of Transistors.

- **DYNAMIC LOGIC**
  - **Transistor Count –33 (without Buffers)**
    -39(with buffers)
  - **Power Dissipation** = 1.4275e-010 W
  - **Propagation Delay TPHL=2.539 nsec. (worst case)**
  - **Area**
    - **Active=8670 sq. units**
    - **Total=46200 sq. units**

Additional Resources- Clock signal with larger Fan In

NOTE: All the Readings are taken for 4 bit Excess-3 code + one ERROR signal which is used to indicate that the input is other than BCD.

Design Steps:

1. DIGITAL DESIGN
2. VHDL DESIGN
3. TRANSISTOR SIZING
4. LAYOUT DESIGN

1.DIGITAL DESIGN:

- Truth table
- Karnaugh's map
- Logic diagram

Truth Table :

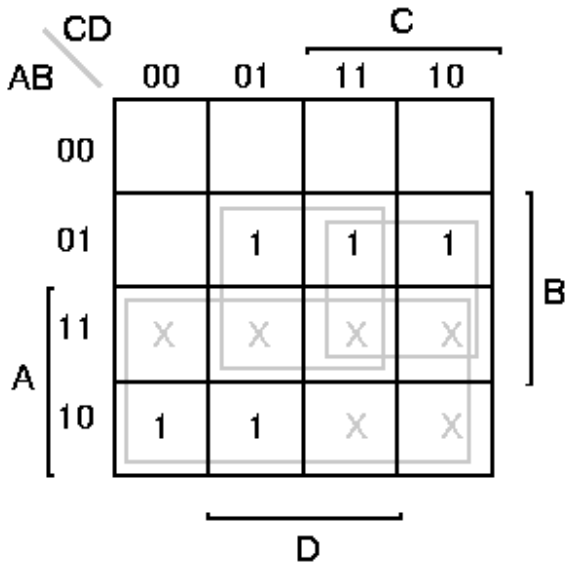
BCD INPUT ABCD	EXCESS-3 OUTPUT W3W2W1W0(WXYZ)	ERROR ER
0000	0011	0
0001	0100	0
0010	0101	0
0011	0110	0
0100	0111	0
0101	1000	0
0110	1001	0
0111	1010	0
1000	1011	0
1001	1100	0
Illegal input(ERROR)		
1010	XXXX	1
1011	XXXX	1
1100	XXXX	1
1101	XXXX	1
1110	XXXX	1
1111	XXXX	1

Karnaugh's Map:

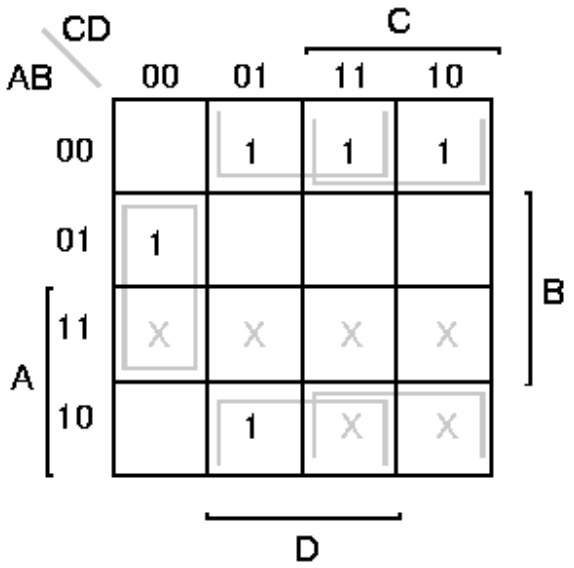
Equations:

(W)W3=A OR (B AND (C OR D))
(X)W2=B XOR (C OR D)
(Y)W1=C XOR D
(Z)W0=NOT D
(ER)ERROR=A AND (B OR C)

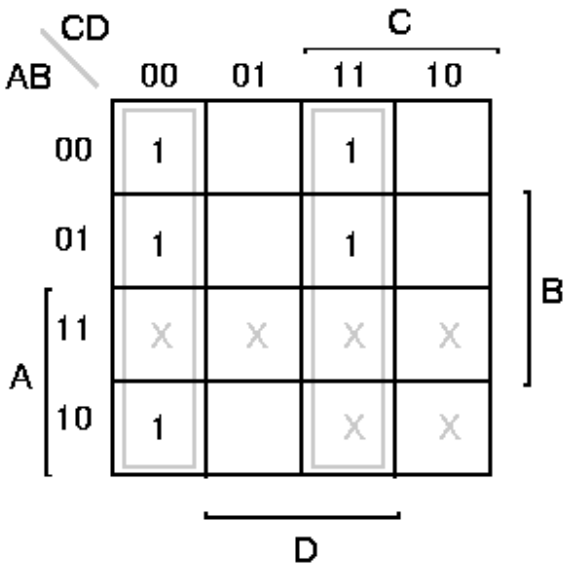
GRAPHICAL REPRESENTATIONS OF K-Map



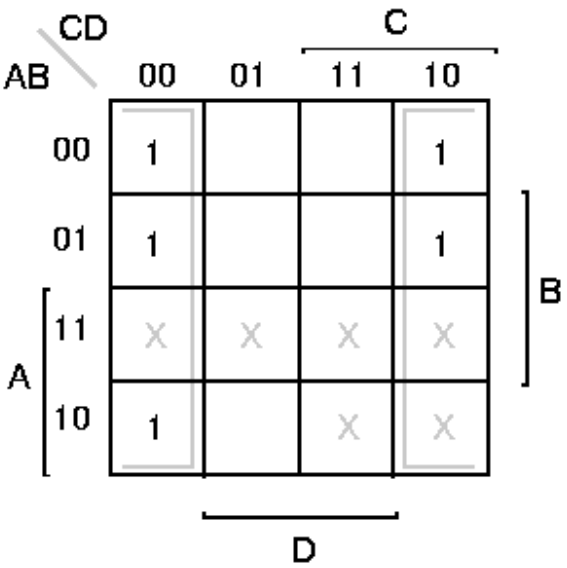
$W = A + BC + BD$



$X = \overline{B}C + BD + B\overline{C}\overline{D}$



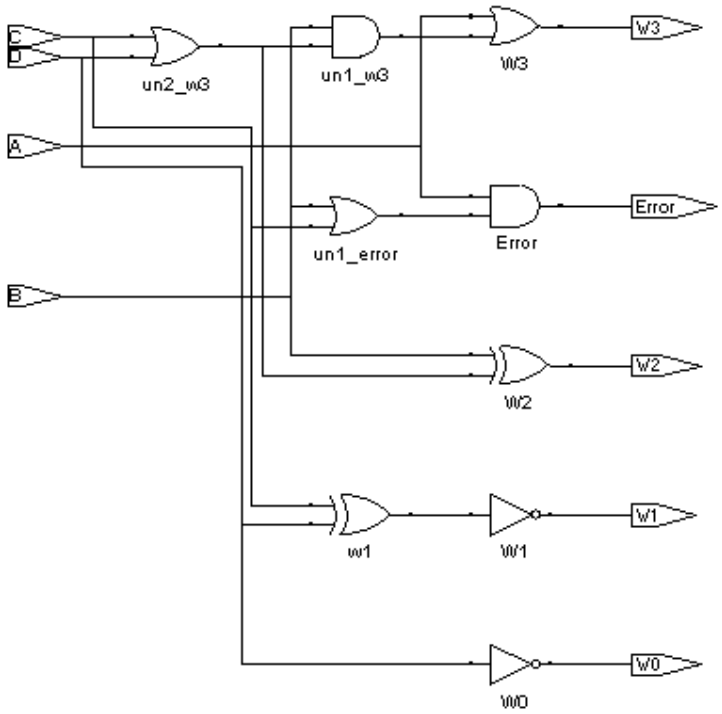
$Y = CD + \overline{C}\,\overline{D}$



$Z = \overline{D}$

Note: **The Signals W=W(3), X=W(2), Y=W(1), Z=W(0) are used interchangeably through out the design.**

Logic Diagram:



(Generated From Sinplicity)

2.VHDL :(Software Used is Cypress –Warp and Sinplify)

- code
- equations generated
- delays
- worst case path

*Code

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity bcd_exc_3 is
        port( A,B,C,D : in std_logic;
                Error,W3,W2,W1,W0 : out std_logic
                );
end bcd_exc_3;
architecture arch_bcd_exc_3 of bcd_exc_3 is
begin
        W3<=A or (B and (C or D));
        W2<=B xor (C or D);
        W1<=C xnor D;
        W0<=not D;
        Error<=A and (B or C);
end arch_bcd_exc_3;
```

Signal information:

ABCD: -input signals with A as MSB and D as LSB
W3W2W1W0:-output signals with W3 as MSB and W0 as LSB.
ERROR:- signal used to indicate invalid input BCD number

Equations Generated

error =a * c + a * b
w0 = /d
w1 =c * d + /c * /d
w2 =b * /c * /d + /b * d + /b * c
/w3 =/a * /c * /d + /a * /b

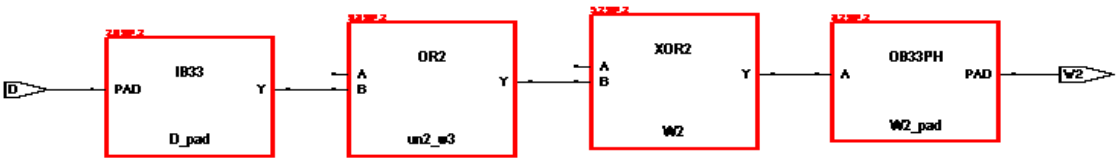
Delays

Signal Name	Delay Type	tmax	Path Description
cmb::w2[24]			
inp::b	tPD	8.5 ns	1 pass
cmb::w3[31]			
inp::a	tPD	8.5 ns	1 pass
cmb::w1[36]			
inp::c	tPD	8.5 ns	1 pass
cmb::w0[37]			
inp::d	tPD	8.5 ns	1 pass
cmb::error[43]			
inp::a	tPD	8.5 ns	1 pass

Worst Case Path Summary

tPD = 8.5 ns for w2

Critical path:
The critical Path is shown in RED colour



(generated by software simplicity)

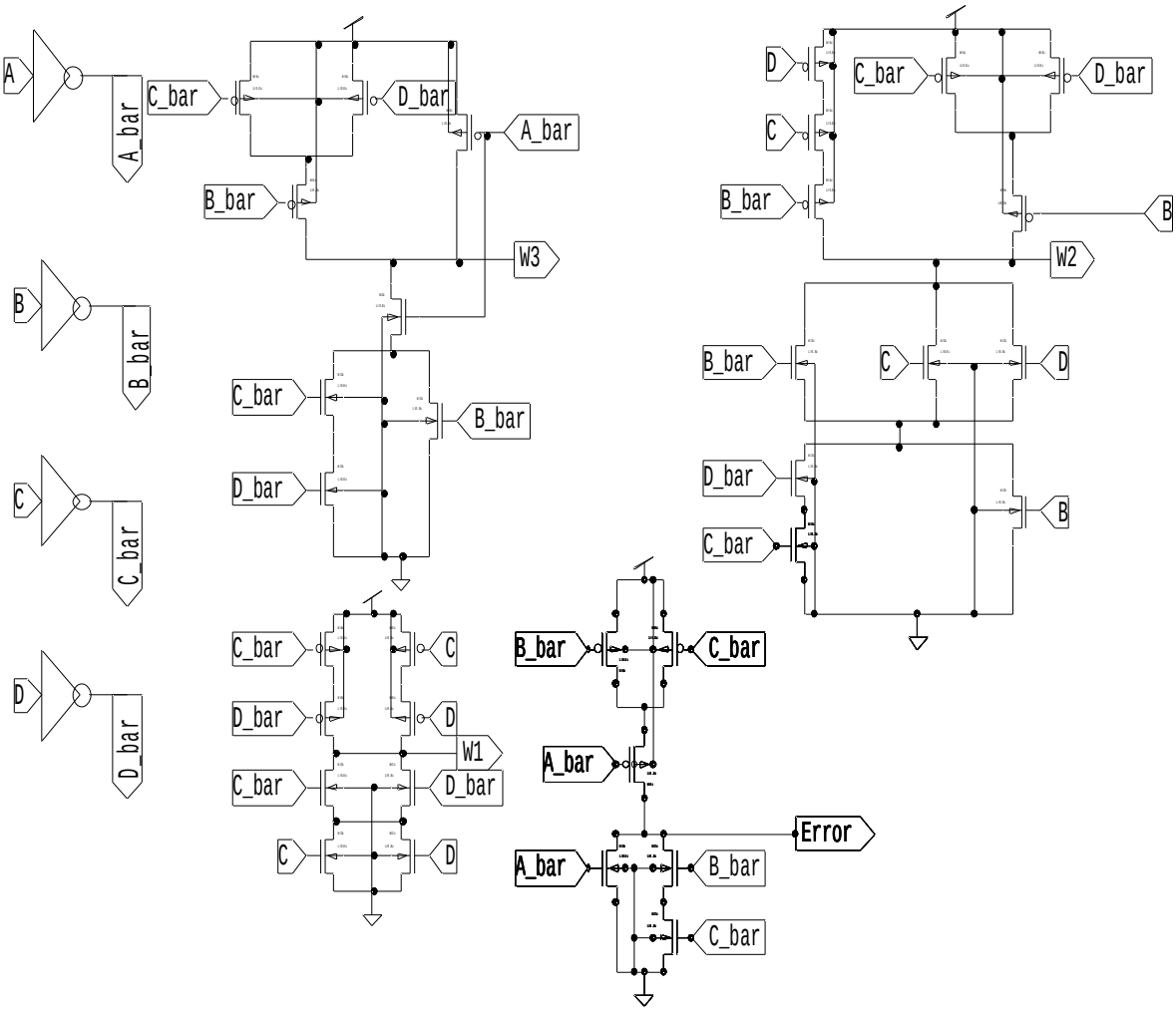
3. Schematic Design:

- S-Edit Schematic
- Transistor sizing
- Load Testing
- Problems and Solution of Load Testing

S-EDIT Diagram

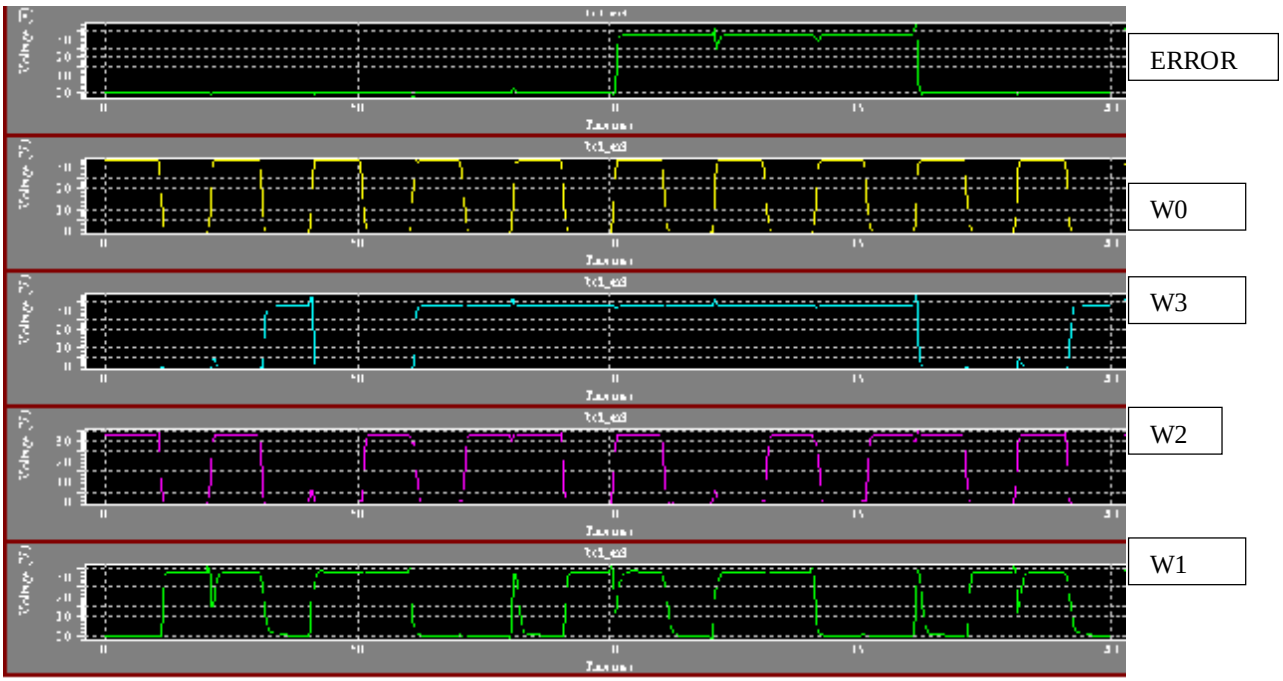
1. **Static CMOS Implementation:**

Static CMOS Design Schematic



Design process followed in Schematic Design:

a. **Design of circuit with minimum size transistors**
ie Ln=0.8um and Wn=2um for NMOS
ie Lp=0.8um and Wp=6um for PMOS
The ratio of Wp/Wn=3.



(output waveforms without transistor sizing)

b.The problem faced is the glitches in the result as shown below-

without transistor sizing:

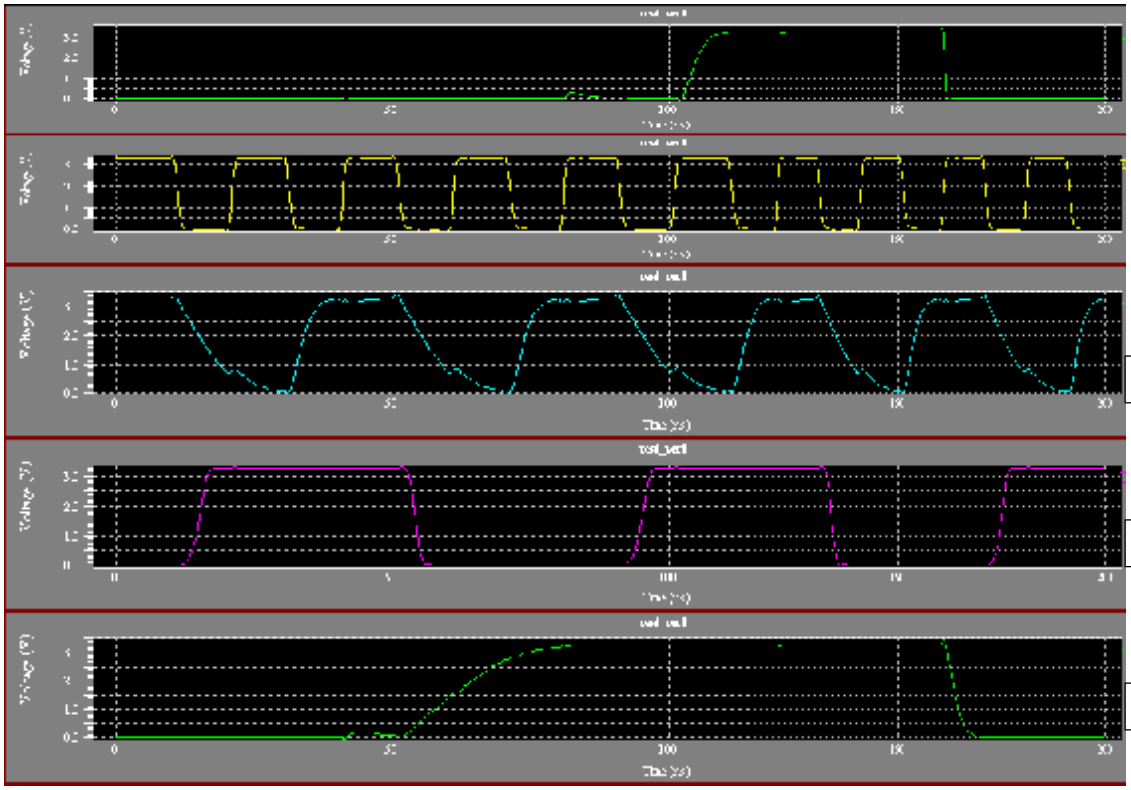
The glitches in the output are due to the charge sharing and the late arriving signals.

- The late arriving signals are connected near the output but it is not the complete remedy. So the next cause is charge sharing .The rising glitches are due to the PMOS and falling Glitches are due to NMOS .by transistor sizing this problem is solved.

2. Transistor sizing is done by iterative measurements i.e. by keeping one parameter constant and other changing i.e. varying the W/L ratio of PMOS
for high to low glitch and of NMOS for low to high glitch the glitch is minimized.

Load Testing:

The Circuit is Tested using the load of 200ff at each node. from waveforms obtained it is observed that the delay is increased due to output load.
This degradation of output is removed by designing a Buffer. by trial and error method the no of buffers used in cascade are 2 so that the logic is maintained and the out quality also increases. The output waveforms obtained with load is given below.



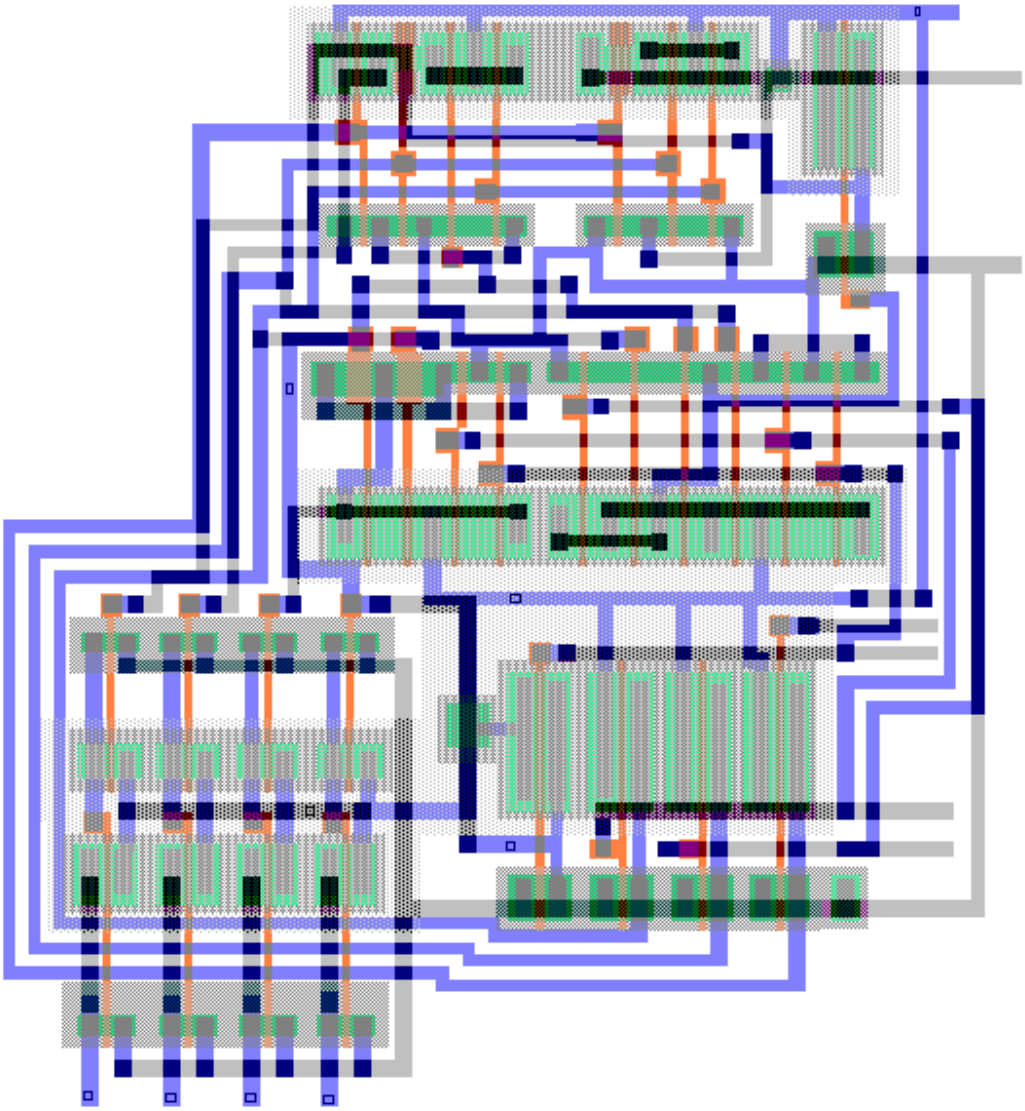
output waveform with load of 200ff without Buffer

Static CMOS LAYOUT:

Waveforms:

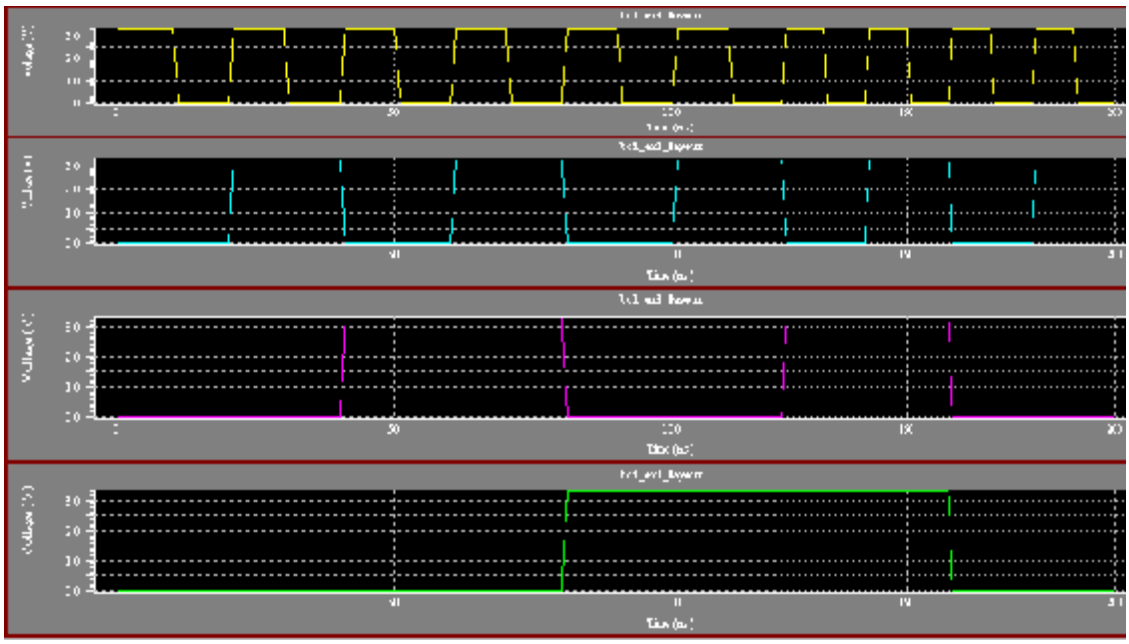
Input sequence for testing:

A(msb)	0000000011111111
B	0000111100001111
C	0011001100110011
D(lsb)	1010101010101010

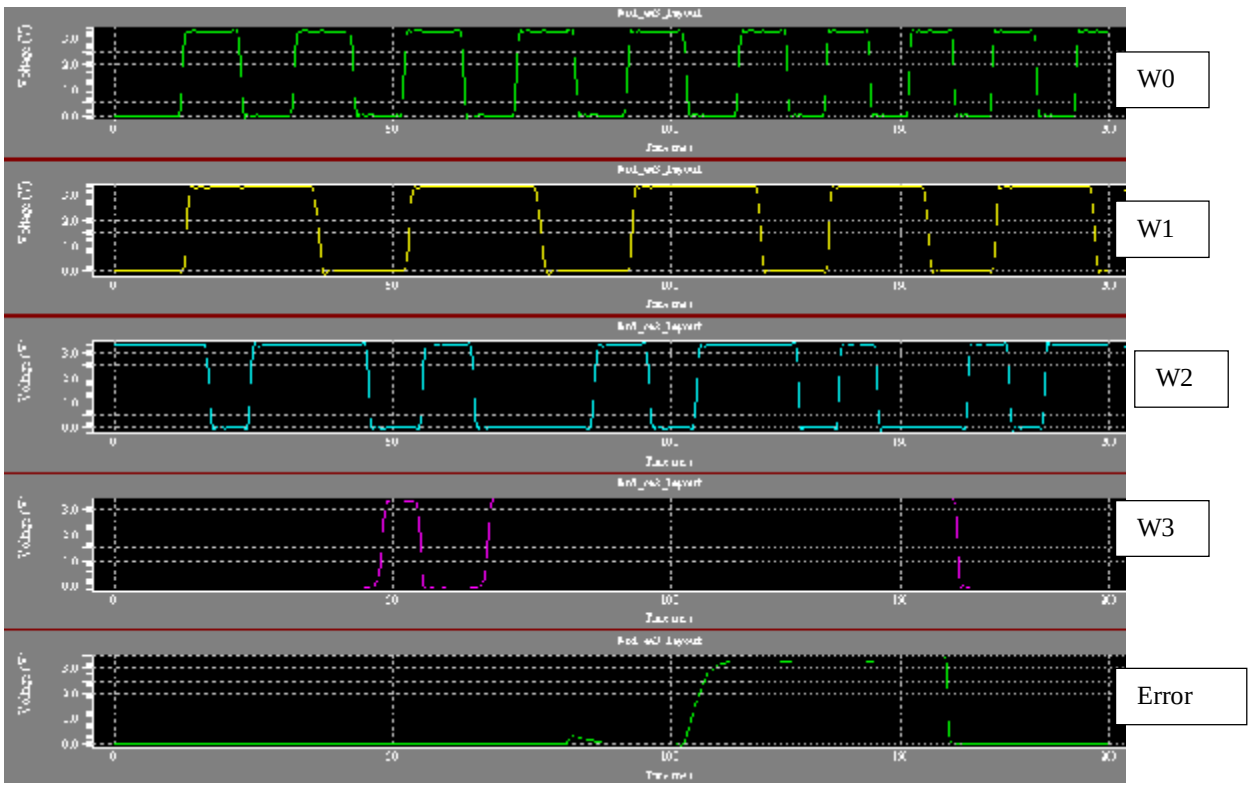


LAYOUT OF BCD to EXCESS-3 CODE Converter using Static Logic.

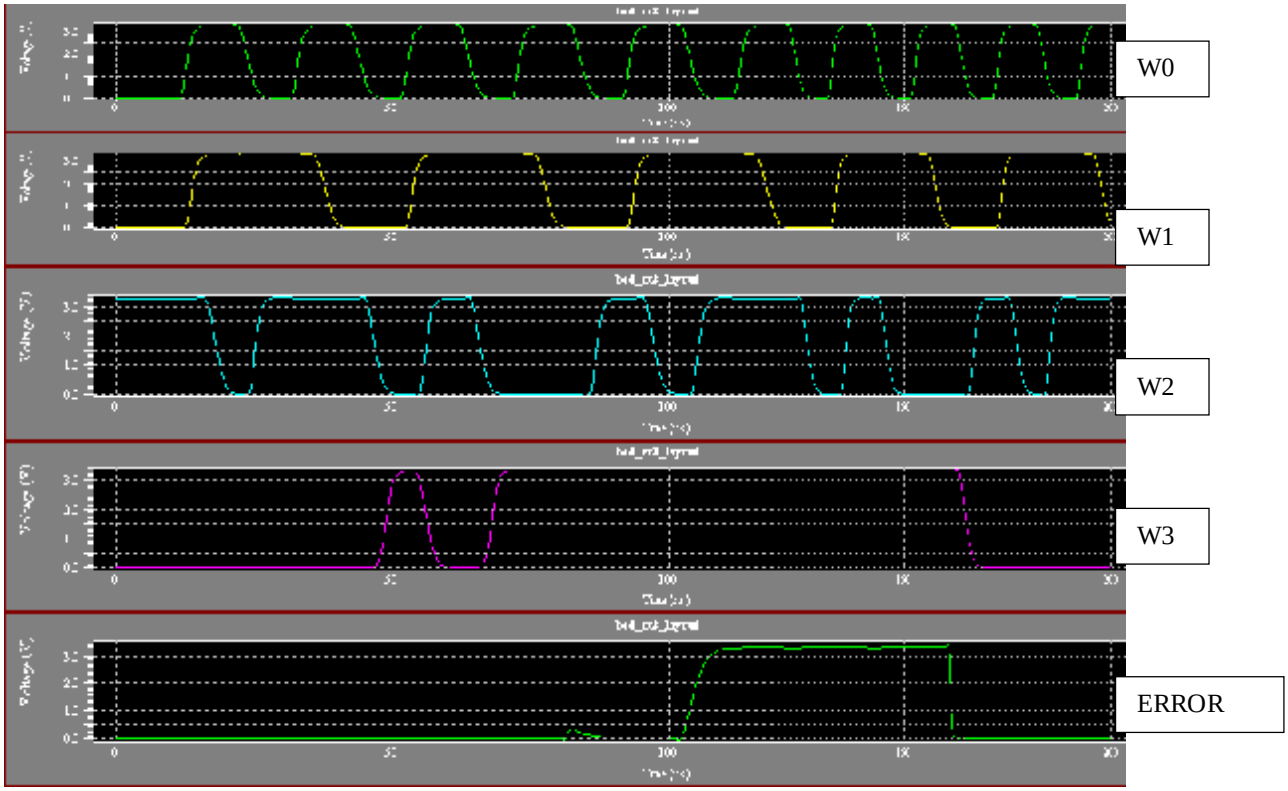
With transistor sizing:
Input waveforms:



Output waveforms without load and with BUFFER



Output waveforms with load 200ff(with Buffer at the output):



Delays in BCD to excess-3 code converter

Without load:

	W3 (ns)	W2 (ns)	W1 (ns)	W0 (ns)	Error (ns)
tplh	7.3766	5.0983	3.2339	1.6268	4.1887
tphl	3.1156	4.3801	5.1374	2.1958	0.91456

With load of 200ff :

	W3 (ns)	W2 (ns)	W1 (ns)	W0 (ns)	Error (ns)
tplh	9.2220	6.6782	4.8016	3.1111	4.1959
tphl	5.6831	6.9615	7.7534	4.7236	9.1406

Worst-case delays:

for W3:

When PMOS B_bar and C_bar or D_bar are conducting

tplh__W3 = 7.5084e-009

when NMOS A_bar C_bar and D_bar are conducting
tphl__W3 = 5.7456e-009

for W2:

when PMOS B_bar and C and D are conducting
tplh__W2 = 4.5541e-009
when NMOS B_bar C_bar and D_bar are conducting
tphl__W2 = 4.5805e-009

for W1:

when PMOS C_bar and D_bar or C and D are conducting
tplh__W1 = 2.4564e-009
when NMOS C_bar and D or C and D_bar are conducting
tphl__W1 = 6.3411e-009

for W0:

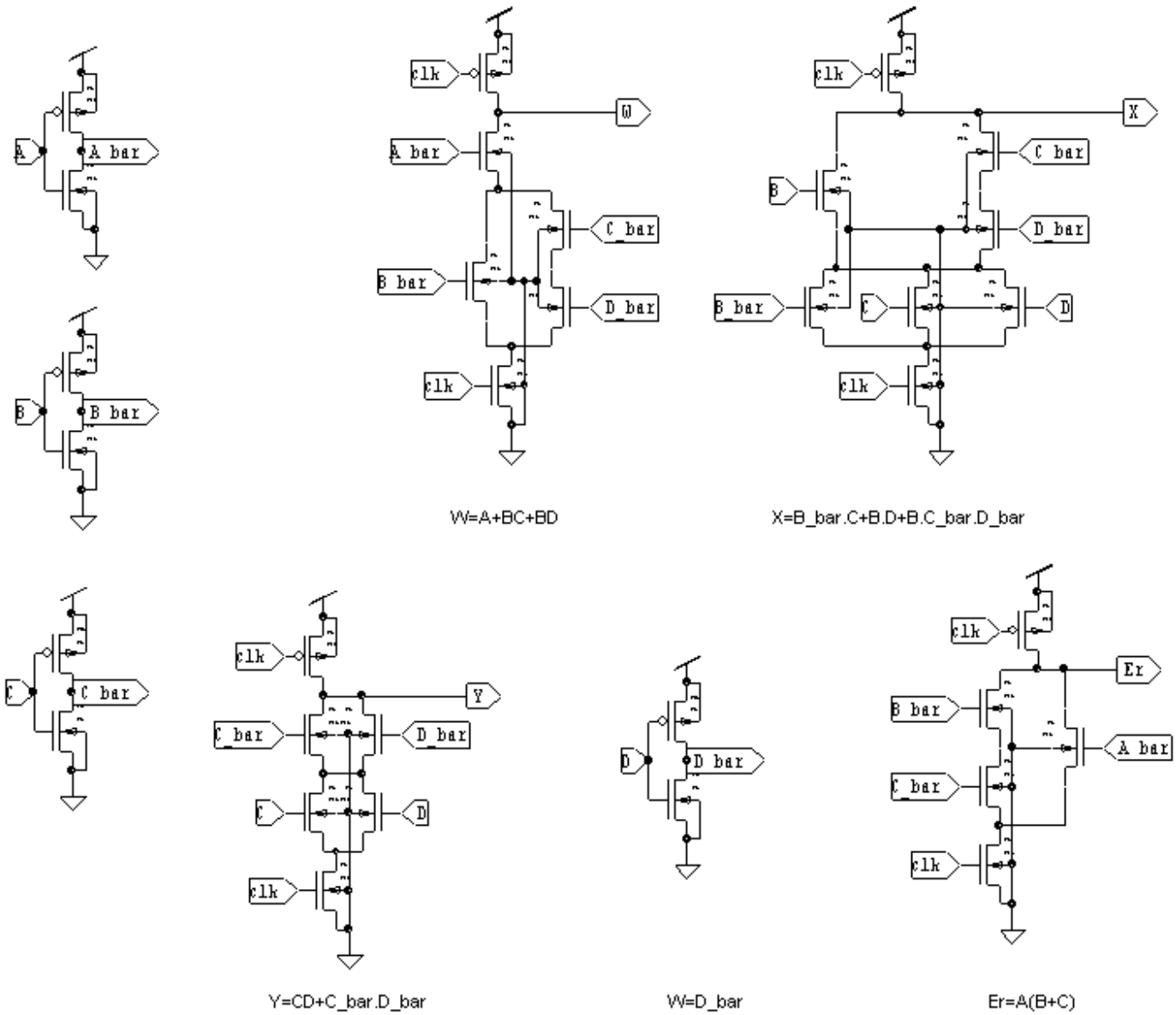
depends on only D
tplh__W0 = 1.5708e-009
tphl__W0 = 2.1738e-009

Power Consumed:

power = 7.4048e-011
From = 0.0000e+000
To = 2.0000e-007

DYNAMIC LOGIC IMPLENTATION

Dynamic Circuit BLOCK Schematic



Schematic Circuit for BCD to EXCESS-3 code convertor using Dynamic Logic.

Transistor sizing in Dynamic logic.

W varies from 2 –8 for inv. & 4-16 for clk NMOs & PMOS inv by 4-16u
Measurement result summary - a=1e-006
tphl = 3.6666e-010
tphl = 7.3252e-010
tphl = 6.0366e-008
tphl = 1.2066e-009
tplh = 9.6248e-010
tplh = 3.5994e-010
tphl = 6.1638e-010

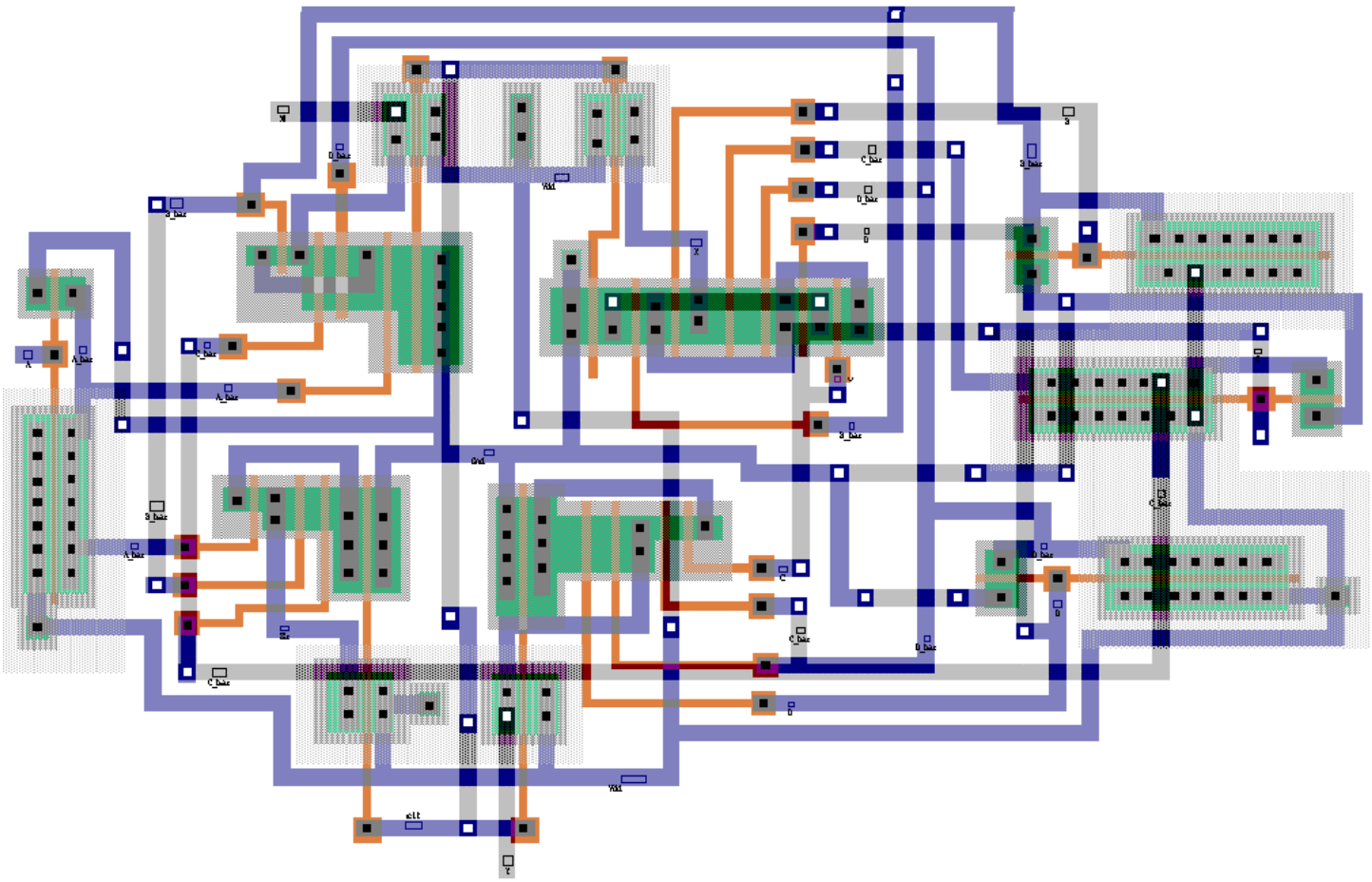
Measurement result summary - a=2e-006
tphl = 3.5909e-010
tphl = 7.9951e-010
tphl = 3.8034e-010
tphl = 1.0505e-009
tplh = 8.9149e-010
tplh = 2.9400e-010
tphl = 3.8159e-010

Measurement result summary - a=3e-006
tphl = 3.3364e-010
tphl = 7.6720e-010
tphl = 3.3674e-010
tphl = 1.0021e-009
tplh = 8.3782e-010
tplh = 2.6435e-010
tphl = 3.2539e-010

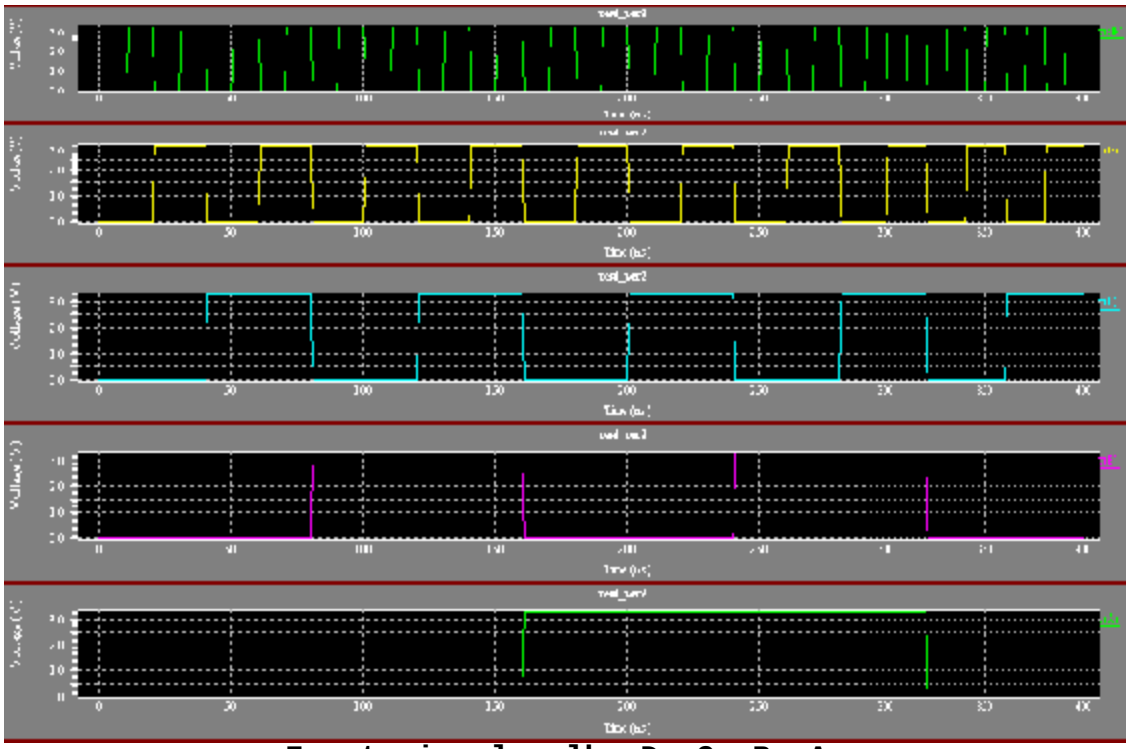
Measurement result summary - a=4e-006
tphl = 3.0872e-010
tphl = 7.9940e-010
tphl = 3.0806e-010
tphl = 9.5909e-010
tplh = 8.3673e-010
tplh = 2.4374e-010
tphl = 2.9360e-010

Power Results for pw=40n & clk pw=20n
vdd from time 0 to 4e-008
integral power consumed -> 8.840991e-011 watts

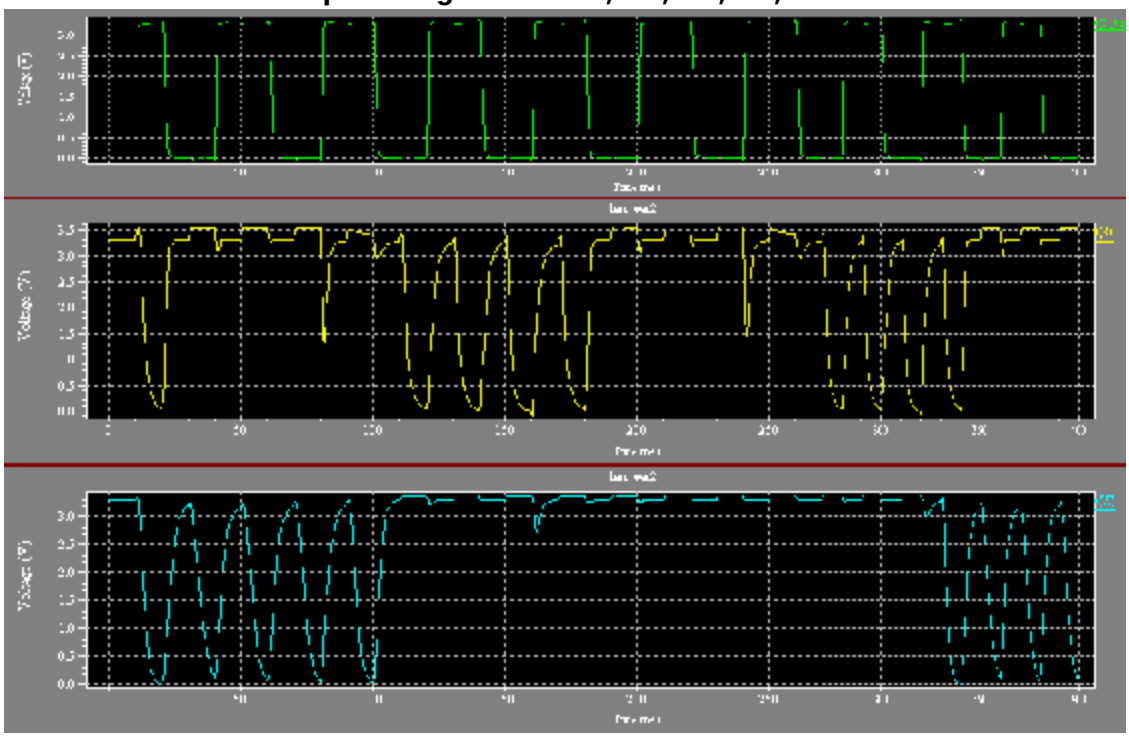
The above reading show a favourable trend towards increasing the size of W=16 u for Clk NMOS Transistor since it needs to pass current in small time to reduce Tphl.



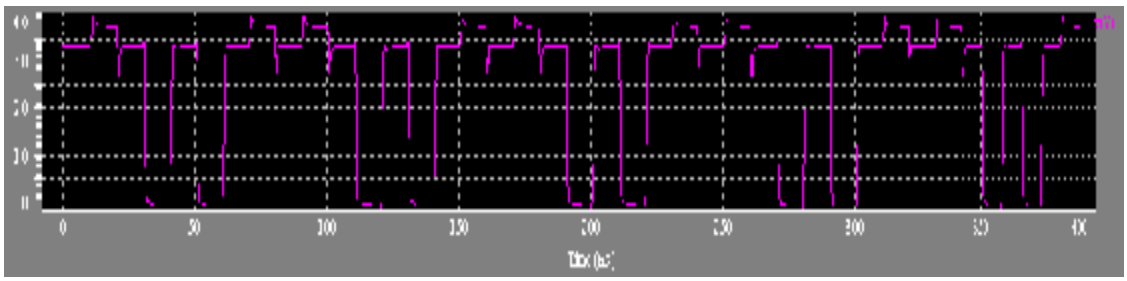
Layout of BCD to EXCESS-3 code converter using dynamic Logic.



Input signals clk, D, C, B, A



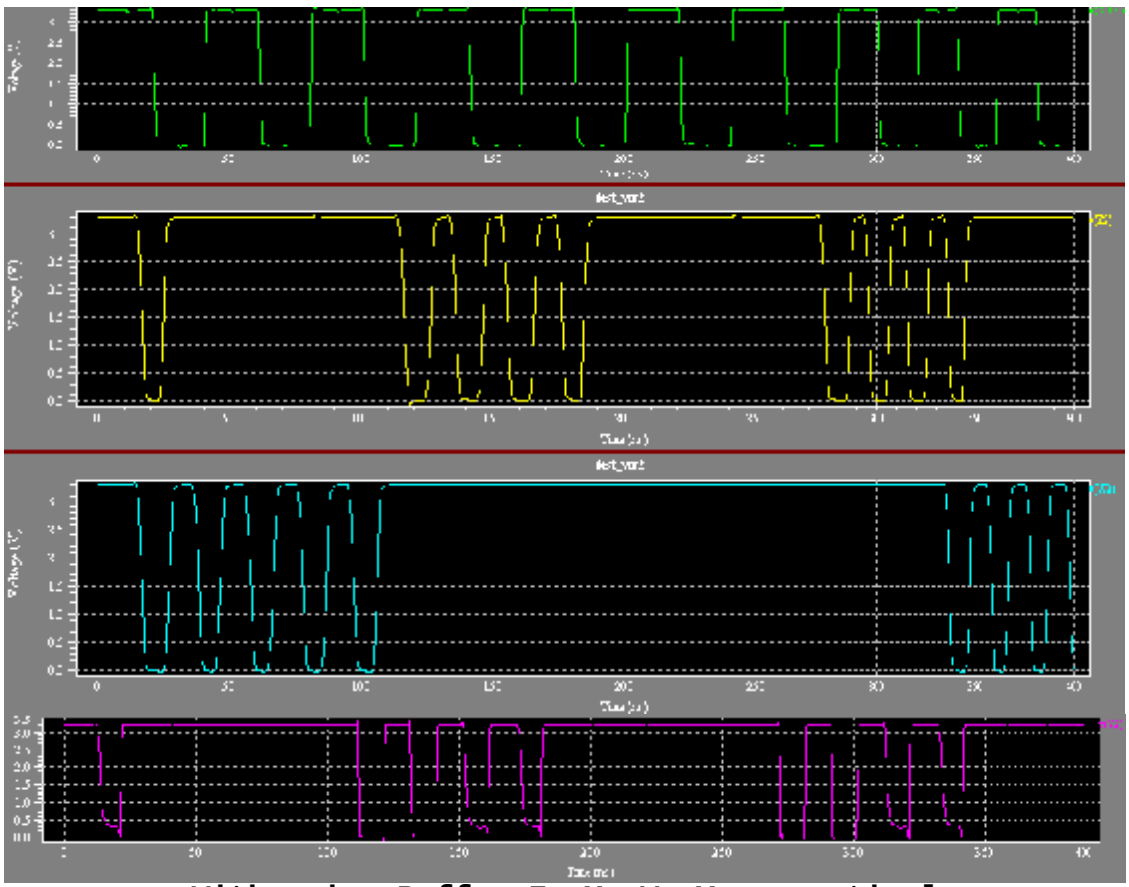
contd.



Without using Buffer Z, X, W, Y respectively.

Measurement result summary

```
   tphl__Y =   2.0327e-009
   tphl__X =   2.7627e-009
   tphl__W =   2.4527e-009
   tphl__Z =   1.0307e-009
   power =     1.1839e-010
```



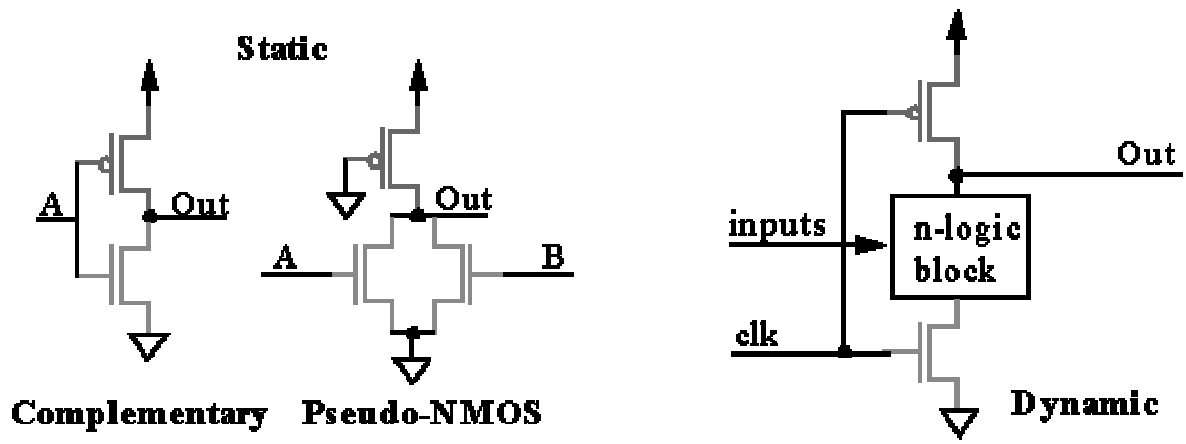
With using Buffer Z, X, W, Y respectively

```
Measurement result summary
     tphl__Y =   1.4528e-009
     tphl__X =   2.7153e-009
     tphl__W =   2.5934e-009
     tphl__Z =   1.0255e-009
     power =     1.4275e-010
```

Comparison of Static vs. dynamic Logic

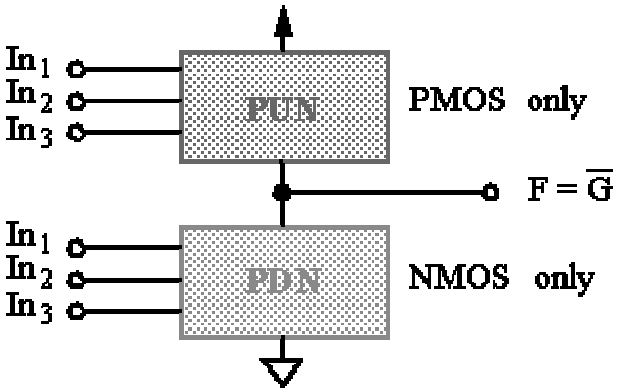
Combinational Logic: Static versus Dynamic

- Static:
  - At every point in time (except during the switching transient), each gate output is connected to either V_{DD} or V_{SS} via a low-resistance path.
  - Slower and more complex than dynamic but "safer".
- Dynamic:
  - Rely on the temporary storage of signal values on the capacitance of high-impedance circuit nodes.
  - Simplier in design and faster than static but more complicated in operation and are sensitive to noise.



Combinational (Non-Regenerative) Circuits

- We've already looked at full complementary design.
- In summary.



Suppose PDN implements G.

But G is connected to GND, so it implements the inverse $F = \overline{G}$

The PUN must implement F, since it's connected to V_{DD}.

- Therefore, the following must hold.

$$\overline{G(In_1, In_2, In_3, \ldots)} = F(\overline{In_1}, \overline{In_2}, \overline{In_3}, \ldots)$$

  - This condition is met if (but not only if) F and G are dual equations, e.g., AND s in F are OR s in G.

Complementary CMOS Gates

- Static CMOS gates inherit the nice properties of the basic CMOS inverter.
- High noise margins.
- No static power consumption.
- Comparable rise and fall times (under the appropriate scaling conditions).

- The last point needs further clarification:
  - This is true if the PUN and PDN networks have identical current-driving capabilities.
  - For the inverter, this required that p-transistors be widened by a factor of m_n / m_p.
  - This is complicated for complex gates since the current driving capabilities are determined by the values of the input signals as well.
  - As we've done in the lab, characterize based on the worst case.

Complementary CMOS Gates

- Performing a manual analysis of the dynamic behavior of complex gates is only tractable via a switch model.

- Here, the transistor is modeled as a switch with an infinite off-resistance and a finite on resistance, R_{on}.
- R_{on} is chosen so that the equivalent RC-circuit has a propagation delay identical to the original transistor-capacitor model.
- R_{on} is inversely proportional to the W/L ratio but varies during the switching transient.
- As we did for (dis)charge currents, we estimate R_{on} at the endpoints of the transitions, e.g., for t_{pHL}:

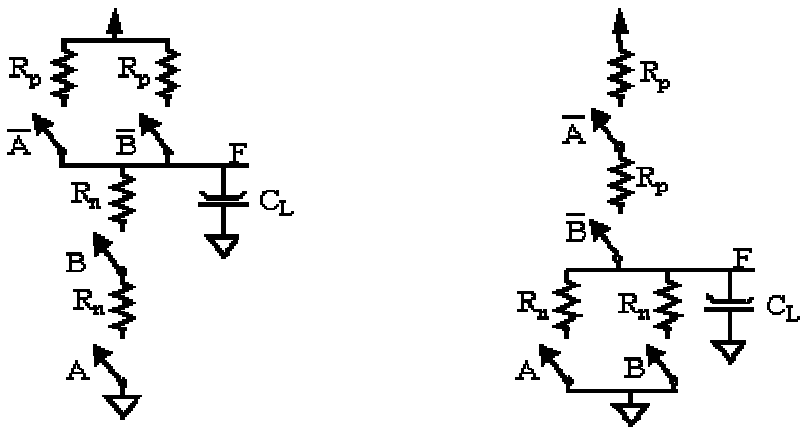
$$R_{on} = \frac{1}{2}\left(R_{NMOS}(V_{out}=V_{DD}) + R_{NMOS}\left(V_{out}=\frac{V_{DD}}{2}\right)\right)$$

Complementary CMOS Gates

- This simplifies to:

$$R_{on} = \frac{1}{2}\left[\left(\frac{V_{DS}}{I_D}\right)_{V_{out}=V_{DD}} + \left(\frac{V_{DS}}{I_D}\right)_{V_{out}=\frac{V_{DD}}{2}}\right]$$

- The text gives an example using 1.2 m m CMOS process with V_{DD} = 5V.
  - R_n is found to be 9.7 k W for t_{pHL} and R_p is 9.6 k W for a W/L_{eff} of 2 and 6 respectively.
- Deriving propagation delay can be done by analyzing the RC network.



Complementary CMOS Gates

- Propagation delay is computed for the worst-case delay over all possible input combinations.

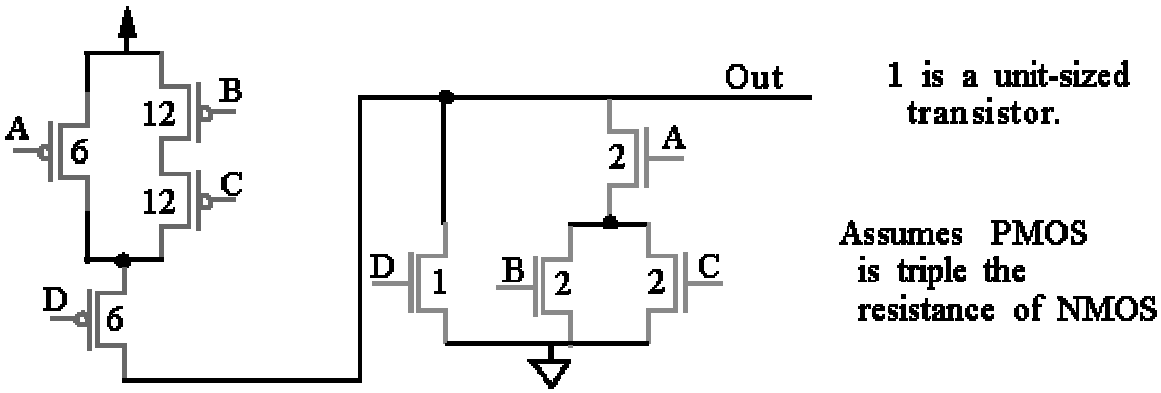
- For the two-input NAND, the worst-case rise time occurs for one PMOS:

$$t_{pLH} = 0.69 R_p C_L$$

- However, the worst-case (only) fall time occurs for two series NMOS:

$$t_{pHL} = 2 \times 0.69 R_n C_L$$

- This suggests the a 2-to-1 width scaling factor of NMOS to PMOS.



Complementary CMOS Gates

- This analysis indicates the deficiencies of implementing gates with large fan-in values:

- A gate with N inputs requires 2N transistors.
  - Other circuit styles require at most N+1 transistors, which can be a substantial advantage in area, e.g., 8 versus 5 for a 4-input gate.

- The propagation delay of a complementary gate deteriorates rapidly as a function of fan-in.
  - First, the larger number of transistors increases the overall capacitance of the gate.
  - Second, the series connection in the PUN and PDN slows the gate.
    - Widening does not improve the performance as much as predicted, since widening increases

gate and diffusion capacitance.

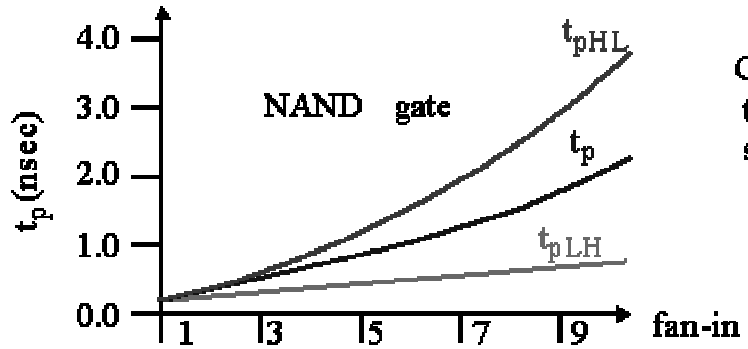
- Fan-out in complementary gates has a larger impact on gate delay than in other circuit styles.
  - Downstream gate capacitance is always two per fan-+ut in contrast to one in other styles.

Complementary CMOS Gates

- Fan-in and fan-out modeling:

$$t_p = a_1 FI + a_2 FI^2 + a_3 FO$$

  - a_1, a_2 and a_3 are technology-dependent weighting factors.
  - The linear dependence on fan-out is easy to understand since load increases linearly with fan-out.
  - There is a quadratic dependence on fan-in since increasing fan-in raises both C_L and (dis)charging resistance in a linear way (under no scaling).



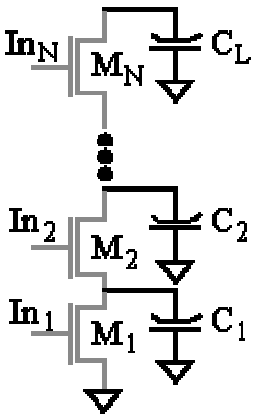
Gates with a fan-in greater than 4 become excessively slow and must be avoided.

Complementary CMOS Gates

- Several approaches may be used to alleviate this problem:

- Transistor sizing
  - Increasing size decreases the second-order factor in the t_p expression.
  - However, as indicated above, if load is dominated by intrinsic capacitance (self-loading), propagation delay is not improved.
- Progressive transistor sizing

- Previous analysis lumped capacitance at the output node and internal node capacitance was ignored.
- This model becomes increasingly inaccurate for large fan-in.

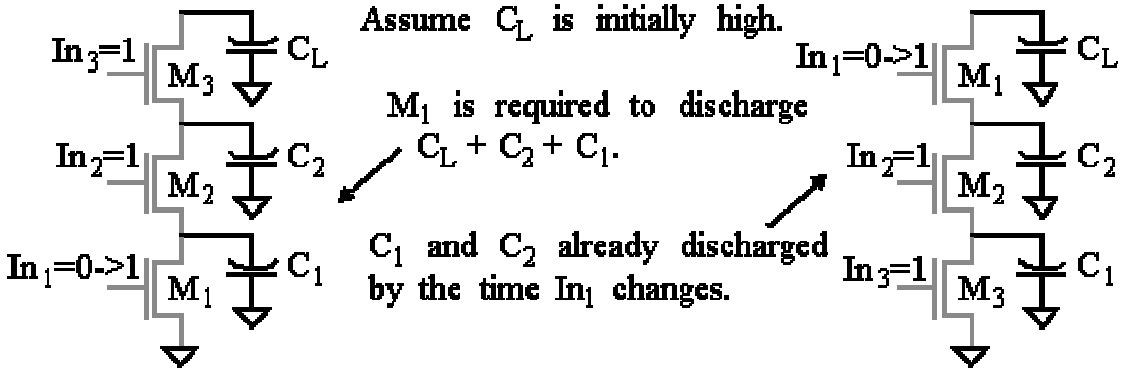


While M_N has to conduct the discharge current of the load capacitance, C_L, M_1 has to carry the discharge current $C_{tot} = C_L + ... + C_2 + C_1$

Therefore, progressive scaling is beneficial: $M_1 > M_2 > ... > M_N$

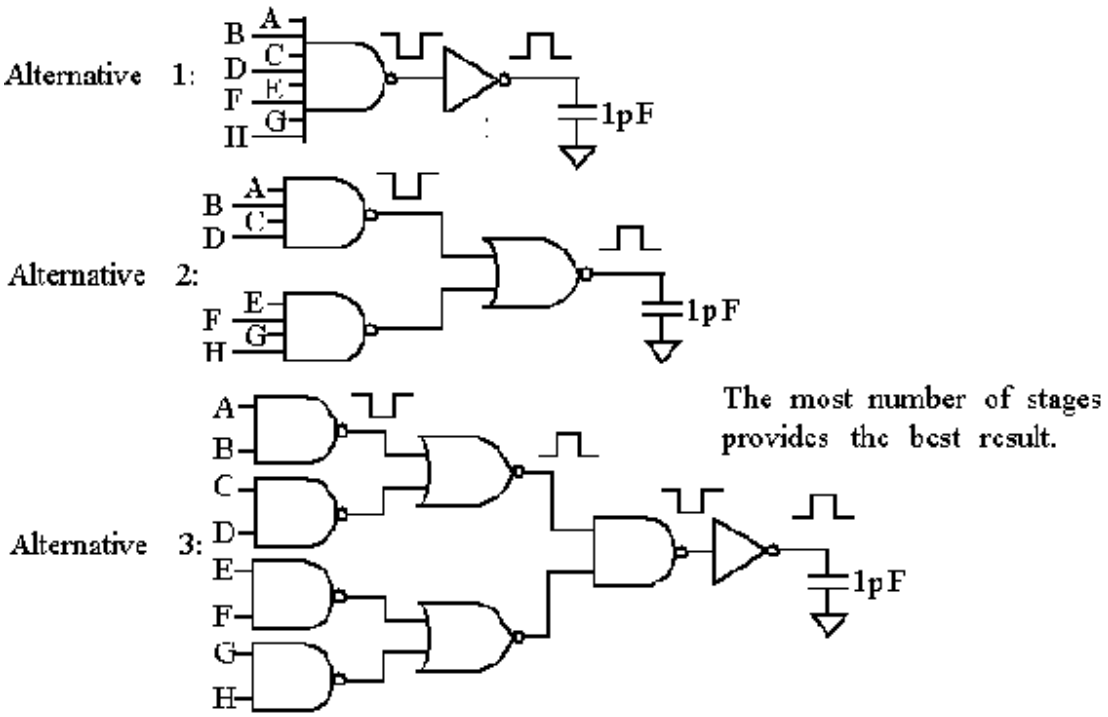
Complementary CMOS Gates

- Transistor ordering
  - Not all input signals to a gate arrive at the same time.
  - Let's call the last arriving input signal critical , which is propagated by a critical path .
  - Putting the critical-path transistor closer to the output of the gate can result in a speed-up.



Complementary CMOS Gates

- Improved Logic Design



Complementary CMOS Gates

- Use Another Circuit Style
  - Ratioed
  - Pass-transitor logic
  - Plus others to be discussed

- These techniques deal with improving performance of gates with large fan-ins .
  - Often speed is dominated by the fan-out factor.
  - Scaling the transistors up in complex logic gates to drive large loads is expensive in terms of area.
  - Instead, a buffer (an inverter, or sequence of inverters) can be inserted between the complex gate and the fan-out.
  - Scaling is applied to the buffer transistors -- the complex gate uses minimum size transistors.

Conclusion

Although one of the goals of Ultra-Low-Power CMOS is to maintain portability of existing circuits, this can be achieved only to a limited extent. This is due to the fact that most high-performance digital VLSI systems employ dynamic circuits (such as domino logic) to achieve higher speed and density Unfortunately, dynamic circuits require a small off-state current which sets a lower limit to the threshold voltage. Although dynamic circuits would work down to 500mV supply voltage with high power efficiency, they would not meet the performance requirements. Static logic, on the other hand, is very robust against high leakage currents and can therefore be scaled to much lower voltages than dynamic logic. Unfortunately, static logic consumes more space on a chip than dynamic logic. There are, however, alternatives to pure static or dynamic circuits, which combine the robustness of static logic with the speed and density of dynamic logic. For example, one could use latched dynamic logic, pseudo-NMOS logic (for large-fan-in NOR gates), or cascade voltage switch logic (CVSL) in time-critical signal paths

BIBLOGRAPHY

1. DIGITAL INTEGRATED CIRCUITS- JAN M. RAEBAY.
2. CMOS VLSI DESIGN –WESTE AND ESHRAGHIAN
3. CMOS VLSI CIRCUIT LAB LABACEE.