University of California
College of Engineering
Department of Electrical Engineering
and Computer Science

Jan M. Rabaey **TuTh9:30-11am**

EECS 141: SPRING 03—MIDTERM 2

NAME	Last First

SID	**Solution**

Problem 1 (12):

Problem 2 (14):

Problem 3 (10):

Total (36)	

Problem 1: Complex Logic: Logical effort and Power (12 pts)

a) (4 pts) Determine sizes of the transistors M_1-M_6 in Figure 1 so that the circuit provides the same pull-up and pull-down current at the output *Z* as a unit inverter. **The input capacitance of all inputs (*A*, *B*, and *C*) should be the same.** The width ratio of PMOS to NMOS in the unit inverter is 2:1. *Hint: Start with the pull-up network.*

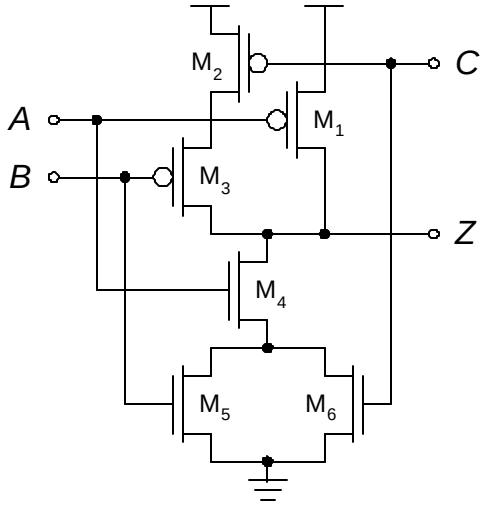


Figure 1. CMOS logic gate

For the same pull-up as an inverter, **$W(M_1)=2$, $W(M_2)=W(M_3)=4$**

Assume **$W(M_4)=x$**, **$W(M_5)=W(M_6)=y$**

- for the same pull-down as an inverter,
  $1/x + 1/y = 1$ (Eq. 1)
- for equal C_{in},
  $4 + y = 2 + x$ (Eq. 2)

$$x = 2+\sqrt{2}$$
$$y = \sqrt{2}$$

M1: 2

M2: 4

M3: 4

M4: $2+\sqrt{2}$

M5: $\sqrt{2}$

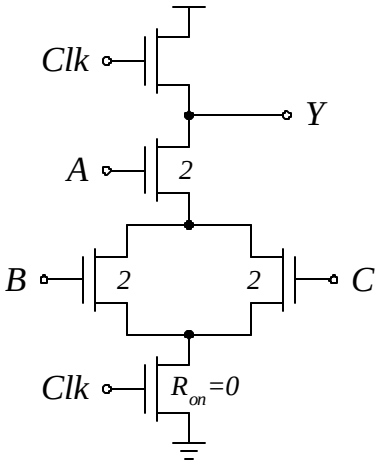
M6: $\sqrt{2}$

b) (1 pts) Determine the logical effort of the resulting gate.

$$g = \frac{C_{in}(gate)}{C_{in}(inv)} = \frac{4+\sqrt{2}}{3}$$

c) (3 pts) Draw dynamic implementation of the gate in (a). What is the logical effort of this dynamic gate? Assume that $R_{on} = 0$ for the evaluation switch transistor.

Dynamic implementation of the gate is shown below.



The logical effort of the gate is:

g = 2/3

d) (4 pts) Construct the truth table for the gate in (a) and compute the probability of the energy consuming transitions of the output, *Y*, if the input probabilities are $p(A = 0) = 0.2$, $p(B = 0) = 0.5$, $p(C = 0) = 0.4$

The truth table for function *Y* is as follows:

A	B	C	Y
0	0	0	**1**
0	0	1	**1**
0	1	0	**1**
0	1	1	**1**
1	0	0	**1**
1	0	1	**0**
1	1	0	**0**
1	1	1	**0**

The probability of the output "0" is:

$$p(Y = 0) = p(A = 1)\cdot p(B = 0)\cdot p(C = 1) + p(A = 1)\cdot p(B = 1)\cdot p(C = 0) + p(A = 1)\cdot p(B = 1)\cdot p(C = 1) = 0.64$$

The probability of the energy consuming transition is equal to:

$$p_{0?\ 1} = p(Y = 0)\cdot(1 - p(Y = 0)) = 0.2304$$

Problem 2: Complex Logic: Logical effort and Power (12 pts)

a) (2 pts) What is the logic function of the following CPL gate?

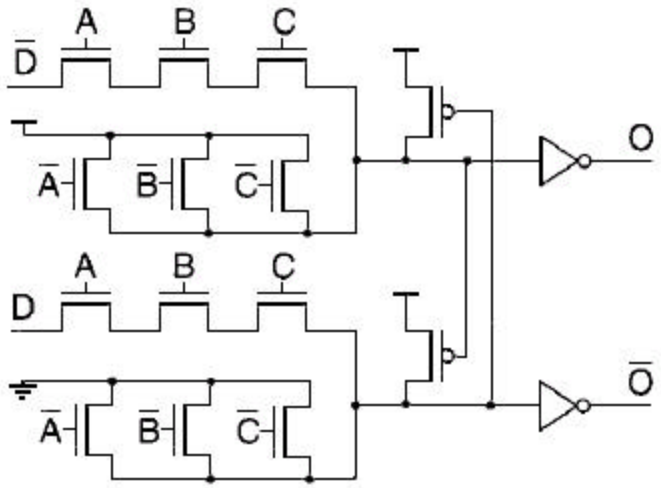
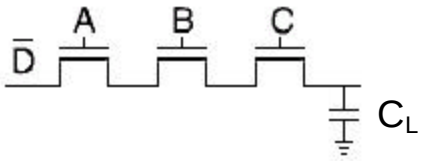
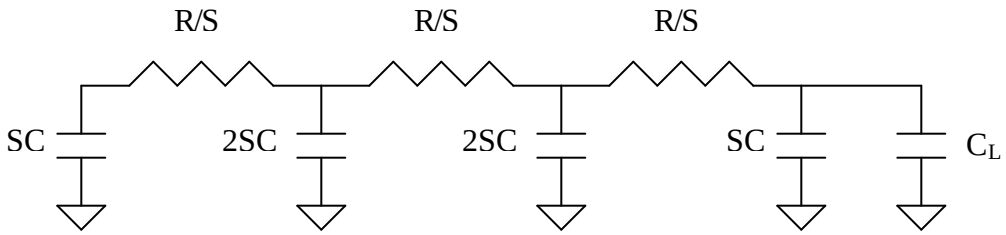


Figure 2. CPL Logic gate

O = ABCD

b) (3pts) The serial NMOS transistors in the logic section of the CPL gate are clearly on the critical path. We have extracted that critical path in the figure shown below. We want to investigate sizing of these transistors so that the delay is minimized. You may assume that a minimum sized NMOS transistor has an on-resistance equal to **R.** In addition, the only parasitic capacitances that you should consider are $C_{gs} = C_{gd} =$ **C** (again for a minimum sized transistor). Finally, we decided all three transistors should remain of the same size S during the sizing. Derive an expression for the propagation delay of this critical path as a function of R, C, the load capacitance C_L and S. Determine qualitatively how you would size the devices to make the propagation delay minimum.



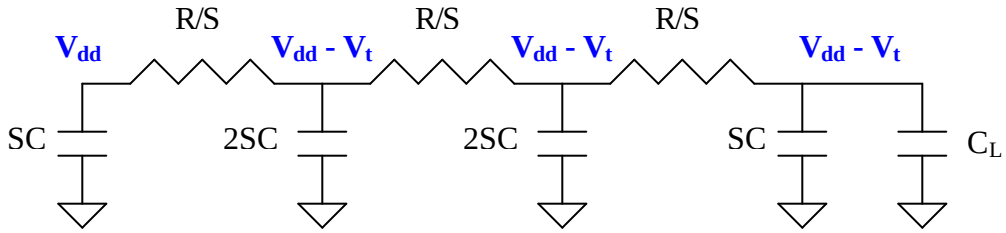


The RC network under worst case input pattern can be modeled as above.

T_p = 0.69*((R/S)*(2SC) + (2R/S)*(2SC) + (3R/S)*(SC+C_L))
= 0.69 * (9RC + 3RC_L/S)

Size the transistors large to reduce delay.

c) (3pts) Derive the energy consumption of this path during one transition (input from high to low) as expression of R, C, S, C_L, V_{dd} and V_t. Again, determine qualitatively how you would size the transistors to minimize the energy consumption/transistion.



E = V_{dd} * (V_{dd} - V_t) * (5CS+C_L) +C*S* V_{dd}^2

Use minimum size transistors for the optimum energy consumption.

d) (3pts) Determine the size S that minimizes the energy delay production (EDP).

EDP = 0.69 * V_{dd} * (9RC + 3RC_L/S) * ((5CS+C_L) * (V_{dd} - V_t) + C*S* V_{dd})

S_{opt} = $\frac{C_L}{C}\sqrt{\frac{V_{dd}-V_t}{3(6V_{dd}-5V_t)}}$

e) (3pts) Design a 4 input multiplexer (see the truth-table below for its fuction) in the complementary pass-transistor logic style using a minimum number of transistors.

S0	S1	Output
0	0	A
0	1	B
1	0	C
1	1	D

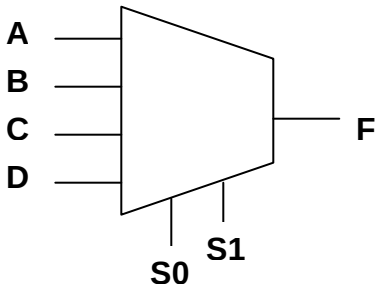
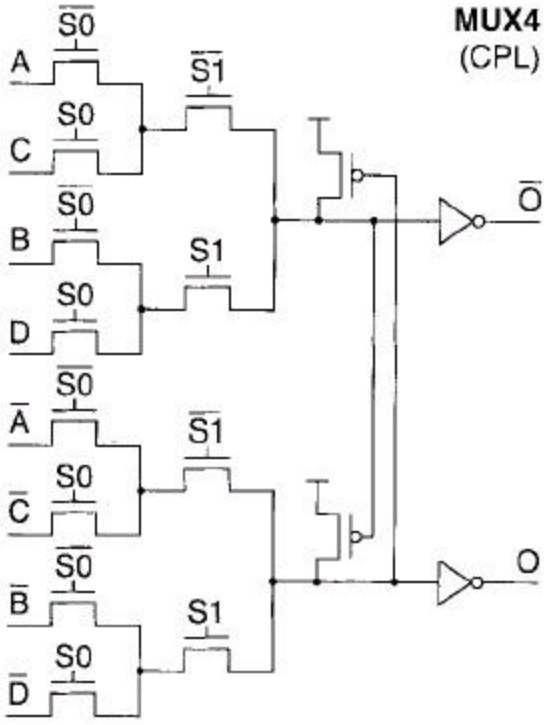


Figure 3. 4-input MUX

MUX4 design using CPL:



Problem 3. Pipelined adder (12 pts)

You have a carry-bypass adder with 4 bits per stage but you find that it is too slow for large total number of bits. Being lazy to go for a different design, you pipeline the adder. A 12-bit section of your circuit is shown in Fig. 4. Answer the questions (a)-(b) in terms of the total number of bits, *N*, and the following one-bit delays:

tp_{pg}	delay through the propagate/generate block = 0.6ns
tp_{carry}	delay of a single carry bit = 1ns
tp_{sum}	delay of a single sum bit = 2ns
tp_{mux}	delay of the multiplexer = 0.4ns
tp_{reg}	delay of the register = 0.5ns

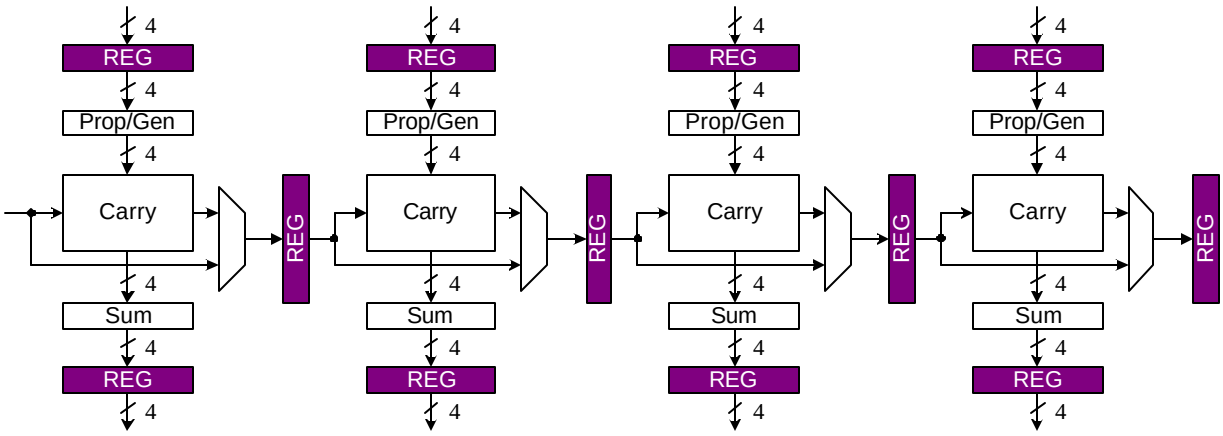


Figure 4. Pipelined carry-bypass adder.

a) (4 pts) What is the minimum clock period you can use for your N-bit adder? (i.e. throughput time). Give your reasoning for full credit. Is the clock period dependent on the total number of bits, N?

The worst-case scenario occurs in the first block, where input carry is not available and needs to be generated.

The worst-case carry propagation delay is:

$$tp(carry) = tp_{reg} + tp_{pg} + 4tp_{carry} + tp_{mux} = 5.5ns$$

The worst-case sum propagation delay is:

$$tp(sum) = tp_{reg} + tp_{pg} + 3tp_{carry} + tp_{sum} = 6.1ns$$

The minimum clock period is therefore:

$$T_{Clk}^{min} = \max\{tp(carry), tp(sum)\} = 6.1ns$$

The clock period does not depend on the number of bits

b) (3 pts) How many clock cycles does it take for the first N-bit addition to complete? (i.e. latency) Is the latency linearly dependent on the number of bits?

There are total of B_{tot} = (N/4 – 1) blocks needed for an N-bit addition. Each block computes with inputs of the preceding blocks. Therefore, it takes N/4 – 1 clock cycles to complete N-bit addition.

The latency is linearly dependent on the number of bits.

c) (3 pts) The pipelined implementation shown above is not entirely correct. Explain why. How would you modify the adder so that correct results are produced?
Hint: Check how a single input word (A, B) propagates through the consecutive stage of the pipeline.

We need to make sure data propagate through the same number of register stages through any path: