50.002 COMPUTATIONAL STRUCTURES

INFORMATION SYSTEMS TECHNOLOGY AND DESIGN

Problem Set 4

1 CMOS Gate

The following diagram shows a schematic for the pulldown circuitry for a particular CMOS gate:

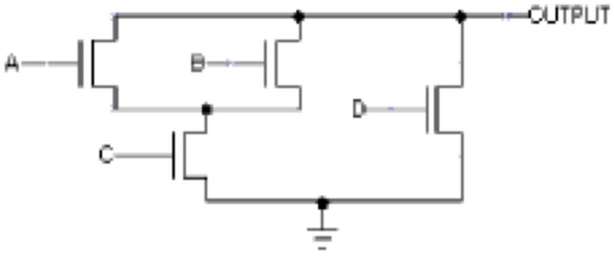

Figure 1

1. What is the correct schematic for the pullup circuitry?
   Solution:
   The output for the pullup circuitry is the inversion of the output of the pulldown circuitry, $\overline{(A+B)C+D} = (\overline{A}\ \overline{B} + \overline{C})\overline{D}$. The plot is located at the first diagram on the next page.

2. Assuming the pullup circuitry is designed correctly, what is the logic function implemented this gate?

   Solution:
   The output for the gate is the output of the pullup circuitry above: $\overline{(A+B)C+D}$.

3. Assuming the pullup circuitry is designed correctly, when the output of the CMOS gate above is a logic ”0”, in the steady state what would we expect the voltage of the output terminal to be? What would be the voltage if the output were a logic ”1”?

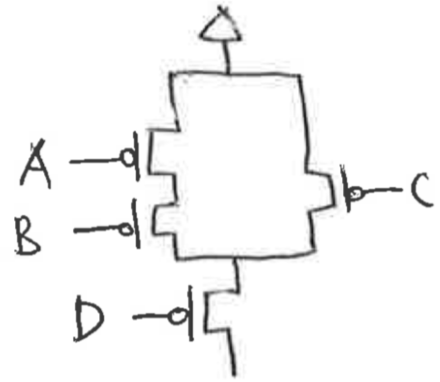

Figure 2

Solution:
The voltage of the output terminal at ”0” steady state is 0 (GND). The voltage of the output terminal at ”1” steady state is VDD’s voltage.

2 CMOS Gate 2

Anna Logue, a circuit designer who missed several early 6.004 lectures, is struggling to design her first CMOS logic gate. She has implemented the following circuit: Anna has fabricated 100 test chips containing this circuit, and has a simple testing circuit which allows her to try out her proposed gate statically for various combinations of the A and B inputs. She has burned out 97 of her chips, and needs your help

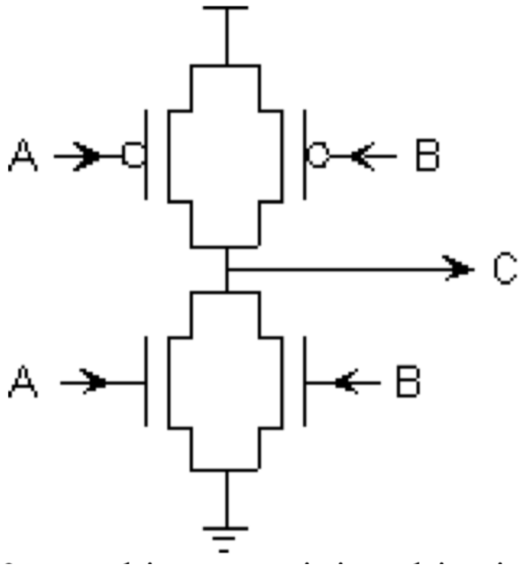

Figure 3

before destroying the remaining three. She is certain she is applying only valid input voltages, and expects to find a valid output at terminal C. Anna also keeps noticing a very faint smell of smoke.

1. What is burning out Anna's test chips? Give a specific scenario, including input values together with a description of the failure scenario. For what input combinations will this failure occur?

   Solution:
   When $A = 0, B = 1$ or $A = 1, B = 0$, then there's an open connection between VDD and GND. This caused the gate to short circuit, and hence its burning out.

2. Are there input combinations for which Anna can expect a valid output at C? Explain.
   Solution:
   Yes, when $A = 1, B = 1$ then $C = 0$, or $A = 0, B = 0$, then $C = 1$. This is when the pullup and pulldown circuit aren't both ON at the same time.

3. One of Anna's test chips has failed by burning out the pullup connected to A as well as the pulldown connected to B. Each of the burned out FETs appears as an open circuit, but the rest of the circuit remains functional. Can the resulting circuit be used as a combinational device whose two inputs are A and B? Explain its behavior for each combination of valid inputs.

   Solution:
   No. When $A = 1, B = 0$, the circuit will burn out again, since the pullup and pulldown will be active, thus burning out the circuit. Also, the output is not defined when $A = 0, B = 1$, since neither the pullup or pulldown are active.

4. In order to salvage her remaining two chips, Anna connects the A and B inputs of each and tries to use it as a single-input gate. Can the result be used as a single-input combinational device? Explain.

   Solution:
   Yes. It exhibits the behavior of an inverter, i.e: A and B are connected to the same V_{IN}.

3 Sum Of Products

A certain function F has the following truth table:
Answer the following questions based on the truth table:

1. Write a sum-of-products expression for F

   Solution:
   $\overline{A}\ \overline{B}\ \overline{C} + \overline{A}BC + A\overline{B}\ \overline{C} + A\overline{B}C + ABC.$

```
A   B   C | F
========|===
0   0   0 | 1
0   0   1 | 0
0   1   0 | 0
0   1   1 | 1
1   0   0 | 1
1   0   1 | 1
1   1   0 | 0
1   1   1 | 1
```

Figure 4

2. Write a minimal sum-of-products expression for F. Show a combinational circuit that implements F using only INV and NAND gates.
   Solution:
   The minimal sum of products is : $\overline{B}A+\overline{B}\,\overline{C}+BC$. You can draw a combinational circuit of this by adding OR gate for every +, inverter, and AND gate for every pair of input in the minimal sum of products.

3. Implement F using one 4-input MUX and inverter.

   Solution:
   If we use A and B as the select inputs for the MUX then the four data inputs of the MUX should be tied to one of ”0” (ground), ”1” (Vdd), ”C” or ”not C”. For this function the following is the correct schematic. Note that by changing the connections on the data inputs we could implement any function of A, B and C.

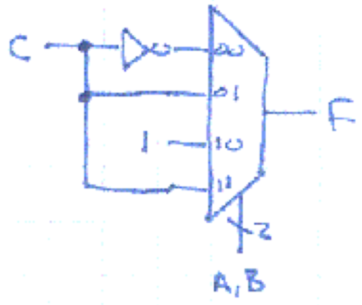

Figure 5

4. Write a minimal sum-of-products expression for NOT(F).
   Solution:
   We can just write equations for patches that cover the ’0’s in the table above, and then reduce the expression into: $\overline{F} = B\overline{C} + \overline{A}\,\overline{B}C$

4 FPGA

The Xilinx 4000 series field-programmable gate array (FPGA) can be programmed to emulate a circuit made up of many thousands of gates; for example, the XC4025E can emulate circuits with up to 25,000 gates. The heart of the FPGA architecture is a configurable logic block (CLB) which has a combinational logic subsection with the following circuit diagram:

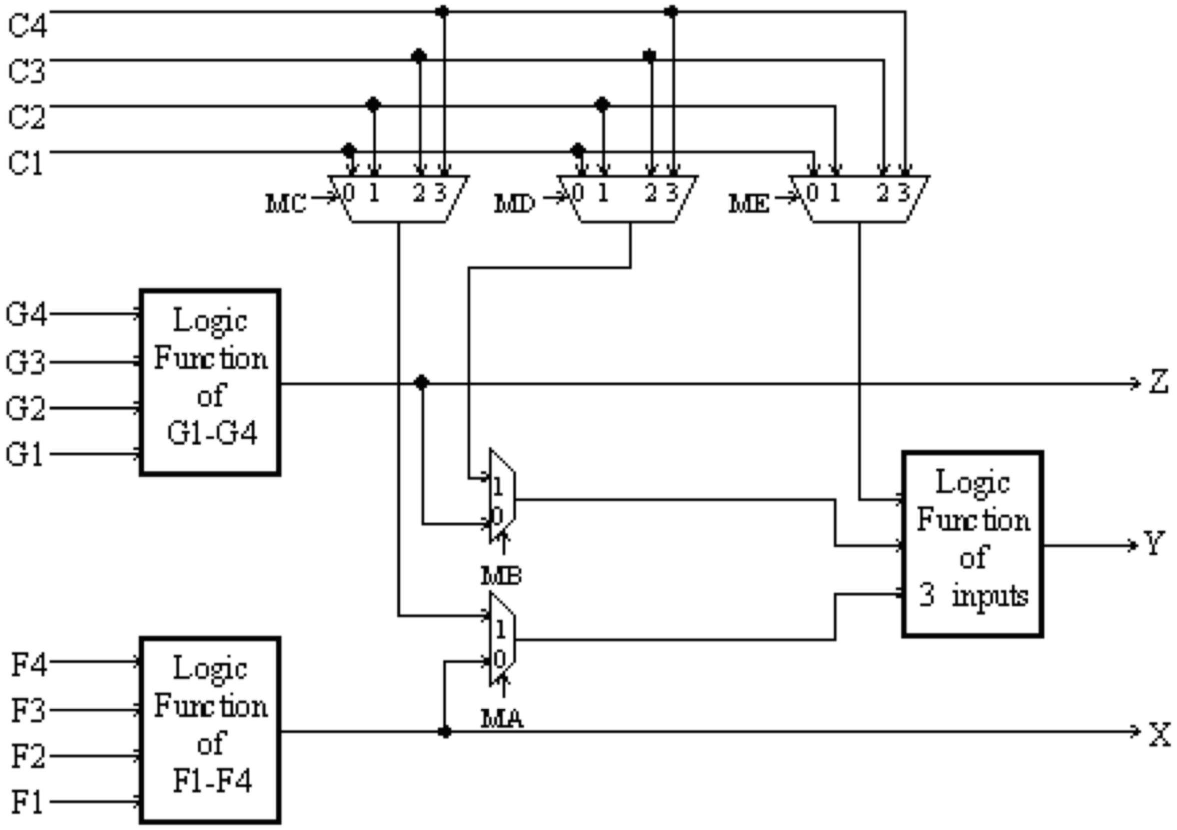

Figure 6

There are two 4-input function generators and one 3-input function generator, each capable of implementing an arbitrary Boolean function of its inputs. The function generators are actually small 16-by-1 and 8-by-1 memories that are used as lookup tables; when the Xilinx device is ”programmed” these memories are filled with the appropriate values so that each generator produces the desired outputs. The multiplexer select signals (labeled ”Mx” in the diagram) are also set by the programming process to configure the CLB. After programming, these Mx signals remain constant during CLB operation. The following is a list of the possible configurations. For each configuration indicate how each the control signals should be programmed, which of the input lines (C1-C4, F1-F4, and G1-G4) are used, and what output lines (X, Y, or Z) the result(s) appear on.

1. An arbitrary function F of up to four input variables, plus another arbitrary

function G of up to four unrelated input variables, plus a third arbitrary function H of up to three unrelated input variables.

Solution:
Let X = F(F1, F2, F3, F4), Z = G(G1, G2, G3, G4), Y = H(C1, C2, C3). The necessary control signals are:

(a) MA = 1

(b) MB = 1

(c) MC = 0 (select C1)

(d) MD = 1 (select C2)

(e) ME = 2 (select C3)

2. An arbitrary single function of five variables.

Solution:
Let Y = F(A1, A2, A3, A4, A5). This can be implemented using both 4-input logic functions, and selecting between the two outputs with the 3-input logic function.

(a) Z=f(A1, A2, A3, A4, 0),

(b) X=f(A1, A2, A3, A4, 1),

(c) Y= Z if A5=0, else Y=X

So Z is calculating F for the case when A5 = 0, X is calculating F for the case when A5 = 1, and Y is selecting between X and Z with a multiplexer function. A1-A4 represents F1-F4 and G1-G4 (they're connected to the same 4 inputs) and A5 represents C1. The necessary control signals are:

(a) MA = 0

(b) MB = 0

(c) MC = X (value doesn't matter)

(d) MD = X (value doesn't matter)

(e) ME = 0 (select C1)

3. An arbitrary function of four variables together with some functions of six variables. Characterize the functions of six variables that can be implemented.

Solution:
Let Z = G(G1, G2, G3, G4) : any function of 4 variables.

(a) X = F(F1, F2, F3, F4)

(b) Y = H(C1, C2, X) = H(C1, C2, F(F1, F2, F3, F4))

The functions of six variables which can be implemented (along with the 4-variable function) are all those functions that can be re-written as a function of 3 variables. The inputs to this function of three variables must be 2 of the original variables and some function of the remaining four variables. The necessary control signals are:

(a) MA = 0
(b) MB = 1
(c) MC = X (value doesn't matter)
(d) MD = 0 (select C1)
(e) ME = 1 (select C2)

4. Some functions of up to nine variables. Characterize the functions of up to nine variables that can be implemented.

Solution:
Let:

(a) X = F(F1, F2, F3, F4)
(b) Z = G(G1, G2, G3, G4)
(c) Y = H(C1, X, Z) = H(C1, F(F1, F2, F3, F4), G(G1, G2, G3, G4))

The functions of nine variables that can be implemented are all those functions that can be re-written as a function of 3 variables. The inputs to this three-variable function will be one of the original variables, plus two separate functions of 4 variables (these two 4-variable functions will have the remaining 8 original variables as inputs).

(a) MA = 0
(b) MB = 0
(c) MC = X (value doesn't matter)
(d) MD = X (value doesn't matter)
(e) ME = 0 (select C1)

5. (Optional challenge) Can every function of six inputs be implemented? If so, explain how. If not, give a 6-input function and explain why it can't be implemented in the CLB.

Solution:

The functions of 6 variables which we can implement must be of the form:

$$Y = y(C1, C2, f(F1, F2, F3, F4))$$

or of the form:

$$Y = y(C1, f(F1, F2, F3, F4), g(G1, G2, G3, G4))$$

(this second function will have some overlap between C1, F1-4, and G1-4; some variables will be connected to multiple inputs) Essentially, the functions we are able to implement are only those for which we can factor a set of 4 variables out of the equation. For example, the following function cannot be implemented by the CLB:

$$Y = A1A2A3A4A5 + A1A2A3A4A6 + A1A2A3A5A6 + A1A2A4A5A6+$$

$$A1A3A4A5A6 + A2A3A4A5A6$$

This function cannot be broken down into either of the forms mentioned above.