

Comp 120 Computer Organization
Spring 2005

Solutions to Problem Set #2
Issued Thursday, 27/1/05

Problem 1 (30 points)

(A) (6 points)

| X | Y | Z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

(B) (6 points) Label each element's output node with a truth table containing the inputs into the element and the output value for those inputs. Once each of the circuit's outputs is labeled, you have the functional specification for the circuit. From here, we want to reduce the tables until we have one. We can start from the final output, look at its table, and then combine the table from one of its inputs (assuming there is another element in front of it). Then we can make a new table with the original inputs, except now we replace the one input, which is an output of another element, with all of its inputs. We repeat until we are down to one table that has all the inputs and the output. For example:

Label the nodes after each element. In this case, we can label the output of the first element, and let's call it A, as a function of x, y. Then final output, let's call it z, is now a function of x and A(x,y). Putting this on a chart:

| | Inputs | | Internal | Output |
|-----|--------|---|----------|------------------|
| | X | Y | A(x,y) | Z = B(x, A(x,y)) |
| | 0 | 0 | 1 | 0 |
| | 0 | 1 | 0 | 0 |
| | 1 | 0 | 0 | 0 |
| | 1 | 1 | 1 | 1 |
| tpd | 0ns | | 3ns | 5ns |

(C) (6 points) 5ns.

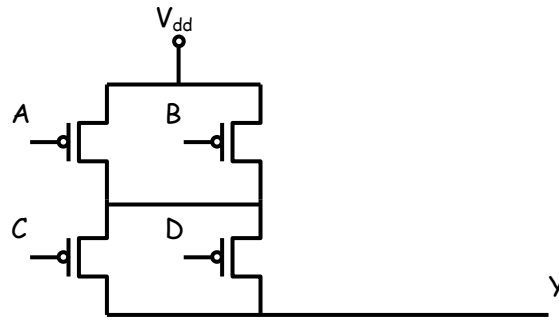
(D) (6 points)

- 1) Label each INPUT to the circuit with tpd=0
- 2) Repeatedly
 - Find a circuit element E whose input nodes are each labeled with a prop delay but whose output nodes are not.
 - Label each output node of E with the delay M, where M is the prop delay of E plus the MAXIMUM of the times on the input nodes.
 - When you can't find an unlabeled output node, stop.
- 3) The prop delay spec for the device is the MAX of the prop delay labels on the output nodes.

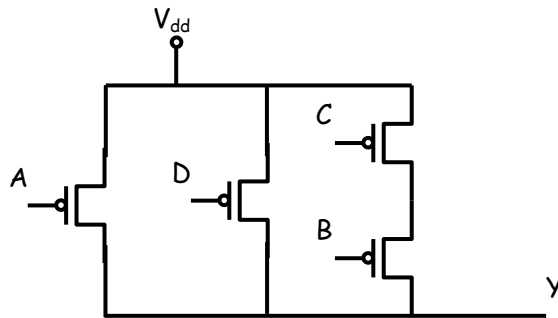
(E) (6 points) No. Without cycles, you're guaranteed to be able to find a new output node to label (i.e. the output of some element E whose inputs are already labeled) until the entire circuit is labeled. If you have cycles, the algorithm breaks down. You can be left with a cycle of elements whose outputs are unlabeled and some of whose inputs are unlabeled.

Problem 2 (20 points)

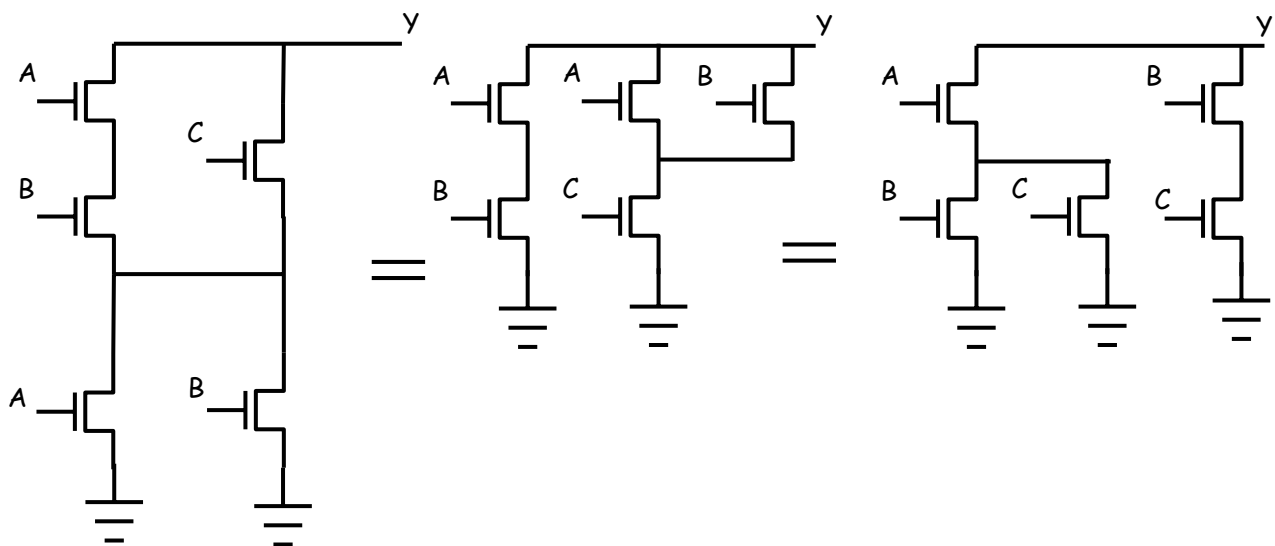
a) (5 points)



b) (5 points)



c) (10 points)



Problem 3. (20 Points)

- a) (5 points) The chips are burning out when the pulldown and pullup are both active. This will occur when $A=0, B=1$ or when $A=1, B=0$.
- b) (5 points) Yes, if $A=1, B=1$, then $C=0$. Or if $A=0, B=0$, then $C=1$.
- c) (5 points) 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.
- d) (5 points) Yes. Since $A=B$, we are left with the following function:

| A | C |
|---|---|
| 0 | 1 |
| 1 | 0 |

Problem #4 (30 points)

- A) (6 points) Let the two 2-bit unsigned integers be represented as A_1A_0 and B_1B_0 , and let the product of these two integers be represented as the binary number $C_3C_2C_1C_0$. The following Karnaugh maps show the values of the C_i bits for all inputs A_1A_0 and B_1B_0 :

| $\backslash B_1B_0$ $A_1A_0 \backslash$ | 00 | 01 | 11 | 10 |
|--|----|----|----|----|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 1 | 0 |
| 10 | 0 | 0 | 0 | 0 |

C_3

| $\backslash B_1B_0$ $A_1A_0 \backslash$ | 00 | 01 | 11 | 10 |
|--|----|----|----|----|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 1 |
| 10 | 0 | 0 | 1 | 1 |

C_2

| $\backslash B_1B_0$ $A_1A_0 \backslash$ | 00 | 01 | 11 | 10 |
|--|----|----|----|----|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 1 |
| 11 | 0 | 1 | 0 | 1 |
| 10 | 0 | 1 | 1 | 0 |

C_1

| $\backslash B_1B_0$ $A_1A_0 \backslash$ | 00 | 01 | 11 | 10 |
|--|----|----|----|----|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 1 | 1 | 0 |
| 11 | 0 | 1 | 1 | 0 |
| 10 | 0 | 0 | 0 | 0 |

C_0

These K-maps translate into the following minimal sums-of-products:

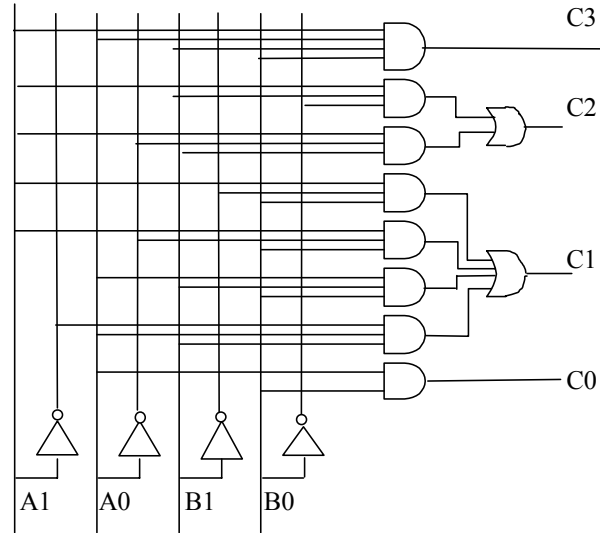
$$C_3 = A_1A_0B_1B_0$$

$$C_2 = A_1B_1\bar{B}_0 + A_1\bar{A}_0B_1$$

$$C_1 = (A_1\bar{B}_1B_0 + A_1\bar{A}_0B_0) + (A_0B_1\bar{B}_0 + \bar{A}_1A_0B_1)$$

$$C_0 = A_0B_0$$

The straightforward way to implement these minimal SOP expressions is the following:



This circuit implementation requires 13 gates (7 AND, 2 OR, 4 INV).

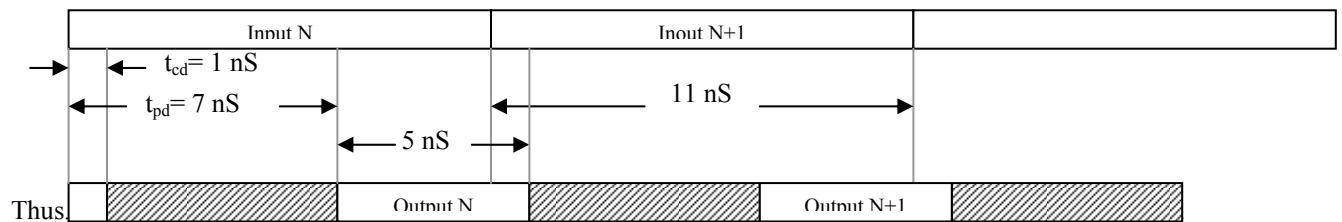
B) (6 points) The following is the truth table for circuit X, which happens to implement the multiplication described in part A:

| A ₂ | A ₁ | B ₂ | B ₁ | P ₈ | P ₄ | P ₂ | P 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 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

| A ₂ | A ₁ | B ₂ | B ₁ | P ₈ | P ₄ | P ₂ | P ₁ |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

C) (6 points) A minimal sum-of-products realization doesn't consider the possibility of multiple outputs that share the same logic blocks. The minimal SOP also has no consideration for the XOR gate or for the reordering of gates, opting instead to use only inverters, ANDs, and ORs in three distinct layers of logic. There are other aspects of minimization that are not listed here as well.

D) (6 points) The contamination delay of the circuit is obtained from the shortest path from an input to an output. In circuit X, this path starts at A₁ (or B₁) and ends at P₁, encountering only one AND gate. Thus, $t_{CD} = 1\text{ns}$. The propagation delay of the circuit is obtained from the longest path from an input to an output. In circuit X, this path starts at any of the inputs and ends at P₄, encountering two AND gates and one XOR gate. Thus, $t_{PD} = 2\text{ns} + 2\text{ns} + 3\text{ns} = 7\text{ns}$. The answer to the next part is best understood by drawing a timing diagram.



E) (6 points) The shortest path from input to output now passes through three AND gates for outputs P_1 and P_8 and one AND gate and a XOR gate for outputs P_2 and P_4 . Thus, $t_{CD} = \min(1\text{ns} + 1\text{ns} + 1\text{ns}, 1\text{ns} + 2\text{ns}) = 3\text{ns}$. The path that creates the largest propagation delay in the circuit is still the path (through three gates) from any input to P_4 , so t_{PD} is still 5 ns. With this new circuit the inputs can transition every 9 ns and still guarantee the outputs will be stable for 5 ns.

