CSE 140 Homework Two

November 1, 2009

*Only Problem Set Part B will be graded. Turn in only Problem Set Part B which will be due on November 2, 2009 (Monday) at 4:00pm.*

1 Problem Set Part A

- textbook 3.1(a)(d)
- textbook 3.3
- textbook 3.6(a)(c)
- textbook 3.9
- textbook 3.10(c)(d)
- textbook 3.12(a)(d)
- textbook 3.14(a)(b)
- textbook 4.4(d)

Using the map method, determine the prime implicates of 4.4(d)

- textbook 4.5(a)(d)
- textbook 4.6(a)(d)
2 Problem Set Part B

1 (Boolean algebra and Axioms)

Recall that a Boolean algebra is an algebraic structure that is defined on a set of elements $B$ with two operators, $+$ and $\cdot$, which satisfies the following axioms:

- **Axiom 1 (Closure Property):** $B$ is closed with respect to $+$ and $\cdot$.

- **Axiom 2 (Identity Element):** $B$ has an identity element $1$ with respect to $\cdot$, and an identity element $0$ with respect to $+$.

- **Axiom 3 (Commutativity Property):** $B$ is commutative with respect to $+$ and $\cdot$.

- **Axiom 4 (Distributivity Property):** $+$ is distributive over $\cdot$, and $\cdot$ is distributive over $+$.

- **Axiom 5 (Complement Element):** for every $x \in B$, there exists an element $x' \in B$ such that $x + x' = 1$ and $x \cdot x' = 0$.

- **Axiom 6 (Cardinality Bound):** there are at least two elements $x, y \in B$ such that $x \neq y$.

The binary Boolean algebra that we are familiar with is built on $B = \{1, 0\}$ over the operations AND and OR.

For each of the following parts, decide whether the given elements in $B$ with the pair of operations given constitute a Boolean algebra. Provide a reasoning if you think they do. If you think they do not constitute a Boolean algebra, please support your answer by telling us at least one axiom which is violated.

(Part A) $B = \{1, 0\}$ over the operations inhibition and implication. Inhibition $(x/y)$ and implication $(x \subset y)$ are defined in the truth table below:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>Inhibition</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
(Part B) $B = \{1, 0\}$ over the operations $\text{NAND}$ and $\text{NOR}$.

(Part C) $B = \{1, 0\}$ over the operations $\text{XOR}$ and $\text{XNOR}$.

(Part D) Consider a 4-variable algebra over $B = \{00, 01, 10, 11\}$. The operation $+$ is defined as bit-wise $\text{OR}$ for each bit while the operation $\cdot$ is defined as bit-wise $\text{AND}$ for each bit. For example, $01 \cdot 11 = (0 \text{ AND } 1)(1 \text{ AND } 1) = 01$; $10 + 01 = (1 \text{ OR } 0)(0 \text{ OR } 1) = 11$. 

2 (Boolean Transformations)

This question concerns the fate of a CSE140 student who is prone to making mistakes during his engineering internship at a local company. This particular student is given a function as a behavioral specification. His job is to use a 2-level OR-AND gate circuit to correctly implement this function as shown in Figure 1.

![Figure 1](image.png)

Figure 1: A correct implementation

After finding the product of sums for the function, the student incorrectly implements it with one of the following mistakes. Either he:

- Mist1 flips all the inputs, or
- Mist2 exchanges the OR gates with the AND gates but keeps all the interconnects identical, or
- Mist3 flips all the inputs and exchanges the OR gates with the AND gates but keeps all the interconnects identical.

Figure 2 provides you with a visual representation of what these mistakes look like. We would like you to identify, for each of the mistakes this student makes, whether:
(A) He will turn out to be lucky and the function he implements will be correct for every possible input combination despite the mistake (thus allowing him to keep his job), or

(B) The particular mistake will cause his implementation to be incorrect for every possible input combination (thus ensuring that he will be fired, and forced to spend more time on his fall quarter courses), or

(C) This mistake will cause his implementation to be correct for some, but not all, input combinations (thus forcing him to demonstrate only the correct cases to his manager if he wants to keep his job).

Clearly show your answer by writing one of A, B, or C in each box of the following tables. (Note that in each case, F is the function the student must implement correctly in the 2-level form above.)
(Part A) In the following cases, “⨀”, denoting the equivalence function, is a behavioral description. All the functions shown are still to be implemented using the 2-level OR-AND gate implementation, as shown in Figure 1.

<table>
<thead>
<tr>
<th>Function</th>
<th>Mist1</th>
<th>Mist2</th>
<th>Mist3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F = x \circ y$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$F = x \circ y \circ z$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$F = u \circ x \circ y \circ z$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Part B) For the first 3 cases of this part, $F$ is a function of 3 variables, while in the last two cases $F$ is a function of 4 variables.

<table>
<thead>
<tr>
<th>Function</th>
<th>Mist1</th>
<th>Mist2</th>
<th>Mist3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F$ is composed of minterms $m_0, m_3, m_4, m_5$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$F$ is composed of maxterms $M_1, M_3, M_5, M_7$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$F$ is composed of maxterms $M_1, M_2, M_3, M_4, M_5, M_6$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$F$ is composed of minterms $m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$F$ is composed of maxterms $M_2, M_5, M_7, M_8, M_{10}, M_{13}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Part C) For this part, $F$ is a function of 5 variables.

<table>
<thead>
<tr>
<th>Function</th>
<th>Mist1</th>
<th>Mist2</th>
<th>Mist3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{31-i}$ is in $F$ iff $M_i$ is in $F$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_{31-i}$ is in $F$ iff $M_i$ is in $F'$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This question concerns various subtractor implementations. You are given a few circuits implementing subtractors using different gates. However, some parts of the circuits are missing. Please complete the circuits by filling in the missing parts.

Figure 3 provides you with an adder implementation with the minimum numbers of operators.
(Part A) Given Figure 3, we want to get similarly a subtractor implementation with the minimum numbers of operators. We provide you with the solution in the following figure, but two critical gates are currently unknown and denoted by square boxes in the figure below; we are asking you to identify them. Please complete the implementation by filling in the square boxes with the appropriate gates without changing any other part of the circuit. Each box corresponds to exactly one 2-input gate.

Hint: Please note the position of the two input signals $B_i$ and $X_i$. What’s more, instead of an AND gate which is in the middle of Figure 1, please note that the new circuit has a NOR gate in the middle of this figure as you can see.

![Diagram](image)

Figure 4: Subtractor with minimum number of operators
(Part B) The problem with the previous solution in (Part A) is that the delay from $B_i$ to $B_{i+1}$ has gotten considerably longer. We have come up with another solution, but two critical gates are still missing. What’s more, another gate should be added to the circuit to get a correct subtraction function. Please complete the implementation by filling in the square boxes with the appropriate gates and adding the missing critical gate to the circuit. Still, each box corresponds to exactly one 2-input gate.

Hint: Please note the position of the two input signals $B_i$ and $X_i$ and the AND gate in the middle of the figure.

Figure 5: An improved subtractor with reduced delay
(Part C) This part concerns the implementation of a subtractor with only NAND and OR gates. Figure 6 presents the implementation of an adder with only NAND and OR gates. To change this implementation to a subtractor, two small changes should be made to the part of the circuit in the dashed square in Figure 6.

A) One NAND gate in the dashed square should be replaced by an OR gate.
B) One line in the dashed square should be changed.

Please complete the implementation by filling in the dashed square boxes on the following figure with only NAND and OR gates. Your answer should be similar to the NAND-OR gate implementation of the adder, with only the two slight changes mentioned above. You cannot change any other part of the circuit.

**Hint: Please note the position of the two input signals $B_i$ and $X_i$.**

While we have given you a number of gate level hints as to what your final answer should look like, you will probably find it easier to approach this problem through Boolean formula transformations rather than random modification of the circuit.