

CSE 140 Homework One

October 7, 2014

*Only **Problem Set Part B** will be graded. Turn in only **Problem Set Part B** which will be due on October 22, 2014 (Wednesday) at 4:30pm.*

1 Problem Set Part A

- Roth&Kinney, 6th Ed: 1.1
- Roth&Kinney, 6th Ed: 1.3
- Roth&Kinney, 6th Ed: 1.4
- Roth&Kinney, 6th Ed: 1.5
- Roth&Kinney, 6th Ed: 1.7
- Roth&Kinney, 6th Ed: 1.15
- Roth&Kinney, 6th Ed: 1.19
- Roth&Kinney, 6th Ed: 1.29
- Roth&Kinney, 6th Ed: 1.31
- Roth&Kinney, 6th Ed: 2.1
- Roth&Kinney, 6th Ed: 2.3
- Roth&Kinney, 6th Ed: 2.5
- Roth&Kinney, 6th Ed: 2.6
- Roth&Kinney, 6th Ed: 2.7
- Roth&Kinney, 6th Ed: 2.8
- Roth&Kinney, 6th Ed: 2.11
- Roth&Kinney, 6th Ed: 2.22
- Roth&Kinney, 6th Ed: 2.30

2 Problem Set Part B

1 (Base Conversion)

(a) If $(143)_7 = (x)_5$, find x .

(b) If $(21321234)_6 = (x)_{36}$, find x .

(c) If $(aaaa)_{a^2} = (x)_a$, find x .

(d) If $(110010.11001)_{a^2} = (___9._____)_{a^3}$, find a and fill in the blanks (with '0' if necessary).

(e) If $(\alpha 2)_{r^3} = (r 3 \beta)_{r^2}$, find α , β and r .

(f) If $(1101.1010)_2 = (x)_{1/2}$, find x . (**Hint:** Think about the positional weights for a Base $1/2$ number.)

2 (Multiplication Algorithms)

The following question requires knowledge about the Basic and Modified Booth's algorithms that you have learned in class. As you remember, in Basic Booth's algorithm, multipliers with alternating 0's and 1's result in sequential addition and subtraction operations, doubling the operations for those cases when compared to Robertson's algorithm. Modified Booth's circumvents this by performing alternative steps for the '010' and '101' cases, the specifics of which can be found in the back of the exam.

(Part A) *Upon graduating from UCSD, you find yourself working at a local processor manufacturer. Your boss has assigned you to help develop the ALU of the new processor that they intend to release later this year. Apparently, the previous processor that they released utilized the Basic Booth's algorithm in its multiplication hardware, and your boss has pushed the team to instead have it use the Modified Booth's algorithm. However, due to the increased amount of hardware needed to perform Modified Booth's, your boss has decided to ignore the '101' special case, instead implementing it in the way Basic Booth's algorithm would do so, where the operation performed will be a '1' and the next value in the flip flop f will be '0'. The '010' case, on the other hand, will be implemented according to the standard Modified Booth's algorithm. Your boss is curious about the efficiency tradeoffs of this design decision, so he assigns you to investigate the matter.*

For the following 8-bit multipliers, please list the number of additions and subtractions that will be performed given this implementation constraint, and state whether this is better, worse, or equivalent to standard Modified Booth's in terms of overall operation count.

Multiplier	# Additions	# Subtractions	Comparison
10010101			
00101011			
10110110			

(Part B) Upon the presentation of your data, your boss has a change of heart and asks you to experiment with ignoring the '010' case, having it now generate an operation of '1' and a new flip flop value of '1'. The '101' case, however, will remain the same as in Modified Booth's algorithm. Please perform the same analysis as in (Part A), but this time with the new constraint in mind.

Multiplier	# Additions	# Subtractions	Comparison
10010101			
00101011			
10110110			

(Part C) *Upon completing your analysis, your company has unexpectedly received a massive amount of funding from an anonymous donor, which excites you as you believe it means that your salary will increase. Your boss, however, has different plans, as more funding also indicates that more money can be spent on the multiplication hardware, thus meaning that the hardware can finally implement the standard Modified Booth's algorithm.* Unfortunately, some of the chips have already been fabricated that implement the two cases described in (Part A) and (Part B). To complicate matters more, some of the circuits have also been fabricated to implement the standard Modified Booth's algorithm, and due to a clerical error, the chips have been mixed into the same shipment. Your boss wishes to diagnose which of the alternate cases, if any, that each of these circuits implement, so he instructs you to test the chips and observe the resulting multiplier encodings. Luckily, your boss informs you that there is no possibility that the circuit only implements Basic Booth's algorithm.

For the encodings below, please diagnose whether the '101' case is ignored, whether the '010' case is ignored, or whether neither case is ignored (and hence the Modified Booth's algorithm is implemented correctly). Furthermore, once you diagnose the error, please reconstruct the 8-bit two's-complement multiplier.

Circuit	Encoded Multiplier	Incorrect Case	Multiplier
1	100 $\bar{1}$ 0 $\bar{1}$ 01		
2	000 $\bar{1}$ 001 $\bar{1}$		
3	$\bar{1}$ 00110 $\bar{1}$ 0		
4	01010 $\bar{1}$ 0 $\bar{1}$		

(Part D) *Now that the company has decided to move forward with manufacturing Modified Booth's multiplication hardware for their ALUs, you believe that your job is done and you can finally move on to bigger and better things.* Little do you know, a large percentage of the ALUs shipped contain a manufacturing error which results in the subsequent operation after every addition/subtraction being lost. For example, one encoding that results from this faulty hardware is shown below, where x indicates a lost operation:

$$0x1x\bar{1}0x1$$

Your boss assigns you to resolve the issue. He believes that Modified Booth's encodings possess certain invariants that allow for the recovery of these lost operations, which he hopes you can utilize in designing some sort of fault tolerance logic to protect against this error in the future. Do you agree with your boss? If so, please state what the lost operations must be along with your associated reasoning. If not, provide a counterexample in which the operations cannot be reconstructed.

(Part E) Your boss also wishes for you to add an additional layer of fault tolerance by inserting 2 counters, one of which counts the number of additions for a given multiplication while the other counts the number of subtractions. He is curious about whether you can reconstruct the two's complement multiplier for a variety of cases, which are given below.

For these cases, inform your boss of the results. If the multiplier can be reconstructed exactly, mark a 'yes' in the 'Exact?' column and write the reconstructed multiplier in **two's complement form**. If the multiplier cannot be reconstructed exactly, mark a 'no' in the 'Exact?' column and write two possible multipliers, again in **two's complement form**. If no multiplier could possibly produce the additions and subtractions specified, please write 'not possible' in one of the columns. Assume that the multipliers are 8 bits.

# Additions	# Subtractions	Exact?	Reconstructed Multiplier
4	0		
3	1		
3	2		
0	4		

3 (Boolean Algebra)

As you remember in your discussion about Boolean Algebra from lecture, the two-valued system utilizes two operators, the $+$ operator and the $\bullet$ operator which satisfies the following truth tables:

$+$	0	1
0	0	1
1	1	1

$\bullet$	0	1
0	0	0
1	0	1

As you have also learned in class, Boolean Algebras do not need to be 2-valued; 4-valued Boolean Algebras may exist as well. In this part of the question, we propose a 4-valued system that may or may not be a Boolean Algebra that acts as follows:

1. For the $+$ operator, given an input of (x_1, y_1) and (x_2, y_2) , the output would be $(x_1 + x_2, y_1 \bullet y_2)$.
2. For the $\bullet$ operator, given an input of (x_1, y_1) and (x_2, y_2) , the output would be $(x_1 \bullet x_2, y_1 + y_2)$.

(Part A) Please fill out the tables using the aforementioned rules. Several of them have been filled out for you in order to get you started.

$+$	(0, 0)	(0, 1)	(1, 0)	(1, 1)
(0, 0)	(0, 0)	(0, 0)		(1, 0)
(0, 1)			(1, 0)	
(1, 0)	(1, 0)		(1, 0)	
(1, 1)		(1, 1)		(1, 1)

$\bullet$	(0, 0)	(0, 1)	(1, 0)	(1, 1)
(0, 0)	(0, 0)	(0, 1)		(0, 1)
(0, 1)			(0, 1)	
(1, 0)	(0, 0)		(1, 0)	
(1, 1)		(0, 1)		(1, 1)

(Part B) Given that the aforementioned system is distributive, does it constitute a valid Boolean Algebra? If you think it does, please provide a reasoning or verify it by checking the axioms. If you think it is not, then identify at least one theorem or axiom that is violated. If you have forgotten the Boolean Axioms or theorems we covered in class, do not fret; they can be found in the back of this exam.

(Part C) As you remember from lecture, it is not possible to have a 5-valued Boolean Algebra. However, that has not stopped one notorious UCSD professor from attempting to construct one. He proposes the following:

+	0	A	B	C	1
0	0	A	B	C	1
A	A	B	C	1	0
B	B	C	1	0	A
C	C	1	0	A	B
1	1	0	A	B	C

•	0	A	B	C	1
0	0	A	B	C	1
A	A	B	C	1	0
B	B	C	1	0	A
C	C	1	0	A	B
1	1	0	A	B	C

Please prove that the above is not a Boolean Algebra by specifying which axiom or theorem it breaks. Again, the axioms and theorems are listed in the back of the exam for your convenience.

The list of Boolean algebra **axioms** follows:

Axiom 1 (Closure property):

- (a) B is closed with respect to the operator $+$;
- (b) B is also closed with respect to the operator $\cdot$;

Axiom 2 (Identity element):

- (a) B has an identity element i with respect to $\cdot$, such that $B \cdot i = B$;
- (b) B also has an identity element j with respect to $+$, such that $B + j = B$;

Axiom 3 (Commutativity Property):

- (a) B is commutative with respect to $+$;
- (b) B is also commutative with respect to $\cdot$;

Axiom 4 (Distributivity Property):

- (a) The operator $\cdot$ is distributive over $+$;
- (b) Similarly, the operator $+$ is distributive over $\cdot$;

Axiom 5 (Complement Element): For every $x \in B$, there exists an element $x' \in B$, the complement of x , such that:

- (a) $x + x' = i$, where i is the identity element with respect to the $\cdot$ operator, and
- (b) $x \cdot x' = j$, where j is the identity element with respect to the $+$ operator.

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

A list of some Boolean algebra **theorems** follows:

Theorem 1 (Idempotency):

- (a) $x + x = x$ and
- (b) $x \cdot x = x$.

Theorem 2 (Identity Absorption):

- (a) $x + i = i$, where i is the identity element with respect to the $\cdot$ operator, and
- (b) $x \cdot j = j$, where j is the identity element with respect to the $+$ operator.

Theorem 3 (Absorption):

- (a) $yx + x = x$ and
- (b) $(y + x)x = x$.

Theorem 4 (Involution): $(x')' = x$

Theorem 5 (Associativity):

- (a) $x + (y + z) = (x + y) + z$ and
- (b) $x(yz) = (xy)z$.

Theorem 6 (De Morgan's law):

- (a) $(x + y)' = x'y'$ and
- (b) $(xy)' = x' + y'$.