

Example Exam1

ECE331

Jens-Peter Kaps

(the material shown might not be representative for the material covered in the actual exam)

- Closed book, one 3"x5" note card with **handwritten** notes allowed. Calculators permitted for basic math only (no text, etc.)
- Show all of your work. Use written English, where applicable. Always write neatly.
- A solution requiring physical units is *incorrect* if the units are omitted from the result.
- Underline, circle or box each result.

- 1) (10 pts) Convert the following numbers in each row from the given base to the remaining three bases listed in the table (i.e., fill-in the open cells in each row). Place your final results directly into the open cells in the table. Express your result in positional notation, i.e., a plus or minus sign (plus sign is optional), followed by the integer portion, followed by a radix point, followed by the fractional portion (**3 digits maximum**). The radix point and fractional portion are only required if the fractional part is non-zero.

Decimal	Binary	Octal	Hexadecimal
-31.75			
			AB.C

- 2) (5 pts) Perform the following computations directly in the base given:

Base 16:

$$\begin{array}{r} 49C \\ + AE1 \\ \hline \end{array}$$

Base 2:

$$\begin{array}{r} 101110 \\ \times 101 \\ \hline \end{array}$$

- 3) (5 pts) Write “Engineer” in ASCII (the full ASCII code is appended to the exam), using an 8-bit *binary* code word in which the leftmost bit is selected to produce **odd parity**. Arrange your solution in the box below:

Letter	ASCII Code (in Binary Form) with Odd Parity
E	
n	
g	
i	
n	
e	
e	
r	

- 4) (10 pts) Fill in the Truth Table below to represent: $F_{W,X,Y,Z} = \overline{W} \cdot \overline{X} \cdot \overline{Y} \cdot Z + \overline{W} \cdot X \cdot Z + W \cdot \overline{X} \cdot \overline{Y}$.

W	X	Y	Z	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

- 5) (15 pts) Use Boolean algebra to determine if the following Boolean statement is true or false. State the identities/properties used in *each* step. (The table below is provided for convenience — use only as many steps as you need.)

$$\overline{\overline{\bar{a} \cdot b + c \cdot \bar{d} \cdot e}} \stackrel{?}{=} \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot b \cdot d + \bar{a} \cdot b \cdot \bar{e}$$

Step	Boolean Expression	Identities/ Properties Used
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		

- 6) (10 pts) Use the Karnaugh Map provided to minimize the following function in **POS** form:

$$F_{X,Y,Z} = \sum(0, 2, 5, 7).$$

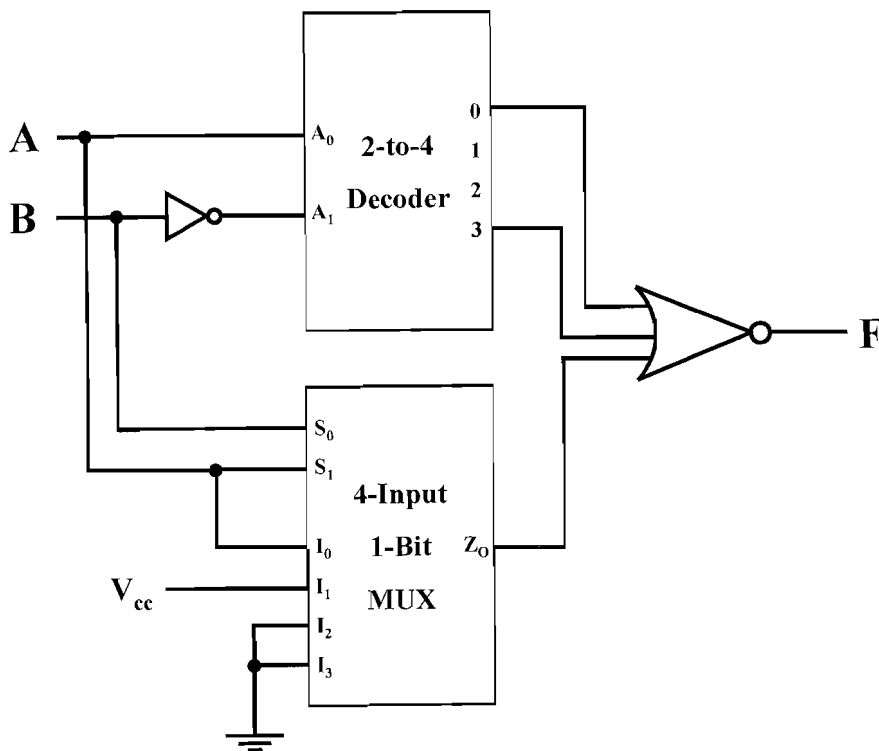
- 7) (10 pts) Use the Karnaugh Map provided to minimize the following function in **SOP** form:

$$F_{W,X,Y,Z} = \sum(0, 1, 9-11) + d(2, 4, 5, 8).$$

8) (25 pts) Let b_0, b_1, b_2 and b_3 form a BCD number (b_0 is the LSB). Design and implement a **minimum SOP** combinational logic circuit to detect when the coded digit is greater than 3. Show the four-step design process of: Step 1 = define inputs and outputs; Step 2 = Truth Table; Step 3 = K-map minimizations (SOP form *only*; do not count inversions of input bits towards the minimum gate count); Step 4 = circuit implementation. Assume that unused input combinations *never* occur. Use generic AND, OR and NOT (inverter) logic gates (*maximum of 4 inputs per gate*) for your implementation.

9) (10 pts) Determine the Truth Table for the logic function implemented in the circuit below, placing your result in the Truth Table below. A_0 is the LSB for the Decoder. S_0 and I_0 are the LSBs for the MUX.

A	B	F
0	0	
0	1	
1	0	
1	1	



- 10) (10 pts) For the pair of decimal values A and B in the table below, (i) express each number in the 1's complement system in columns 3 and 4, respectively, **using 8 bits** (ii) express each number in the 2's complement system in columns 5 and 6, respectively, **using 8 bits** and (iii) express their sum and difference in the 2's complement system in columns 7 and 8, respectively, **using 8 bits**.

In all cases, write a result as "OVERFLOW" if it cannot be contained in the 8-bit representation.

Decimal Value:		1's Comp. Number System (8 Bits)		2's Comp. Number System (8 Bits)			
A	B	A	B	A	B	A + B	A - B
-97	30						

- 11) (15 pts) Use basic logic gates (AND, OR, NOT — any number of inputs) to **design** a **minimum SOP** combinational logic circuit to add a **two-bit** unsigned binary number (e.g. $00_2 \rightarrow 0_{10}$, $01_2 \rightarrow 1_{10}$, $10_2 \rightarrow 2_{10}$ and $11_2 \rightarrow 3_{10}$) to a **one-bit** unsigned binary number, with sufficient output bit length to contain any result.

Your **design** should be a three-input (two bits for the two-bit unsigned integer and one bit for the one-bit unsigned integer), three-output (for the three-bit unsigned integer sum) combinational logic circuit. (Note that overflow is not possible.)

Show your truth table, method of logic function minimization, and final logic equations for your outputs. To save time, you need **not** draw the resultant logic circuit.