ECE 5362 Homework 7
Due 10:20am Nov 22 (Carmen PDF Submission)

1. (36 points, 12 points each) Below are three test files that are similar to those used for the machine problems. Determine the contents (in hexadecimal) of AC, X, SP, and CVZN (3 pts each) after manually executing each of those test files based on the OSIAC description. Those can be used as test cases for your machine problems.

(a)		(b)		(c)	
0008	AC	A004	AC	0001	AC
0007	X	EEED	X	0007	X
8000	SP	0007	SP	0003	SP
0000	PC	0001	PC	0004	PC
0000	CVZN	0000	CVZN	1101	CVZN
3207		0200		1001	
3004		4012		5204	
3034		5008		0642	
3028		5500		0003	
3401		0007		0650	
FFFF		6004		0000	
0000		0000		0641	
0002		DCBA		FFFC	
0004		00EE		B0C0	

2. (20 pts, 10 pts each) As discussed in class, the two outputs of a full adder, s_i and c_{i+1} can be simplified to $s_i = x_i \oplus y_i \oplus c_i$ and $c_{i+1} = x_i y_i + x_i c_i + y_i c_i$. Starting from the true table given in class, show how they are simplified algebraically, step by step. Be sure to show any intermediate steps (you do not have to give the names or numbers of the theorems). Note that you CANNOT use truth table to show the equivalence or Karnaugh map to do simplification.

3. (44 pts) For the carry-lookahead adder discussed in class, answer the following questions. Note that you may need to read Section 9.2.1 for this problem.

   (a) Use four of the 16-bit carry-lookahead adders (as shown in Figure 9.5 of the textbook) along with additional logic circuits to design a 64-bit adder. You do not need to plot a diagram, but just to derive the logic expressions of c_{16}, c_{32}, c_{48}, and c_{64} as functions of c_0 and the G_i^{II} and P_i^{II} variables. Hint: Section 9.2.1 of the textbook has explained how to derive c_{16} in a 16-bit carry-lookahead adder and you can do this problem similarly. (16 pts, 4 pts each)
   (b) What are the gate delays to produce s_{63} and c_{64} in this 64-bit adder? Explain clearly how you get those delays. (12 pts, 6 each)
   (c) What are the gate delays to produce s_{31} and c_{32} in this 64-bit adder? What are the gate delays for the same variables in a 32-bit adder built from a direct cascade (instead of having a higher-level lookahead logic) of two 16-bit lookahead adders? Explain clearly, step by step, how you get those delays. (16 pts, 4 each)