

Exercise 7: Combinational Arithmetic Circuits



Revision: August 31, 2009

215 E Main Suite D | Pullman, WA 99163
(509) 334 6306 Voice and Fax

STUDENT					GRADER			
I am submitting my own work, and I understand penalties will be assessed if I submit work for credit that is not my own.					#	Points	Score	Total Score
<div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> _____ _____ </div> <div style="display: flex; justify-content: space-between;"> Print Name ID Number </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> _____ _____ </div> <div style="display: flex; justify-content: space-between;"> Sign Name Date </div>					1	3		
<div style="text-align: center; margin-bottom: 10px;">Estimated Work Hours</div> <div style="border: 1px solid blue; padding: 2px; display: flex; justify-content: space-around; width: 100%;"> 12345678910 </div> <div style="border: 1px solid blue; padding: 2px; display: flex; justify-content: space-around; width: 100%; margin-bottom: 10px;"> 12345678910 </div> <div style="text-align: center;">Overall Weight</div>					2	3		
					3	4		Adjusted Score
					4	3		
					5	4		
					6	4		
					7	6		
					8	4		
					9	5		
					10	5		
					11	3		
					12	6		
					13	4		
					14	4		
					15	3		
					16	3		
					17	4		
					18	0.1		
					<i>Adjusted Score:</i> Deduct 20% from score for each week late			

Problem 1. Sketch a block diagram for a magnitude comparator bit-slice circuit. Create K-maps to define the bit-slice circuit, and use them to find optimal logic equations. Sketch the circuit.

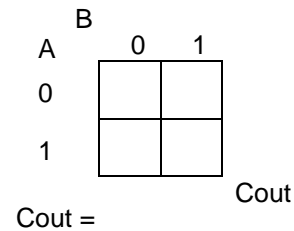
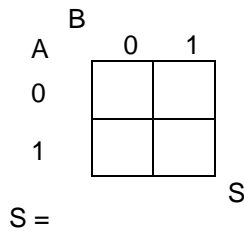
Problem 2. Modify the block diagram and circuit of problem 1 by removing the logic gates and signals that form the EQ output. Sketch a circuit diagram for a 4-bit comparator that uses the modified bit slice blocks, and add a single gate to form the EQ output from the LT and GT outputs from the MSB (most significant bit). Comment on the differences in the new circuit (i.e., which circuit is more efficient? Which is easier to design and implement? Which might run faster? Anything else?)

Could you make the bit-slice modules even more efficient by leaving in the EQ logic and removing some other logic? Explain.

Problem 3. Complete truth tables and K-maps for HA and FA circuits, using XOR patterns where appropriate. Loop minimum SOP equations, and sketch the circuits (assume all inputs and outputs are active high).

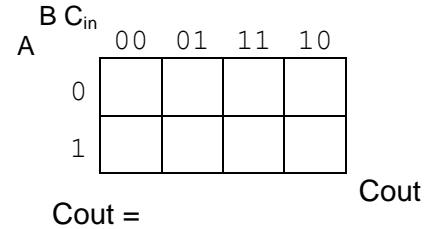
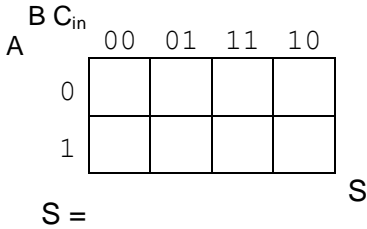
Half Adder

A	B	S	Cout
0	0		
0	1		
1	0		
1	1		



Full Adder

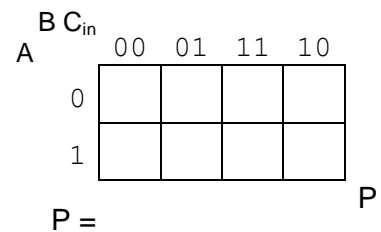
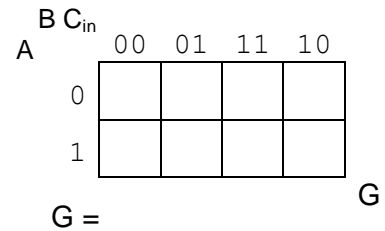
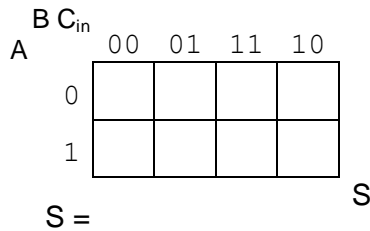
A	B	Cin	S	Cout
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Problem 4. Sketch a circuit for a full adder using two half-adder blocks and an OR gate.

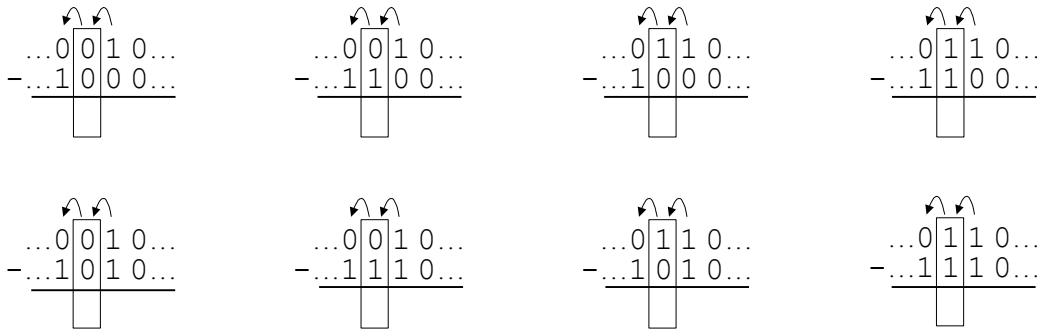
Problem 5. Complete the truth table and K-maps for a CLA adder bit-slice module, and sketch a minimal SOP circuit (be sure to use XOR's where appropriate!).

A	B	Cin	S	G	P
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

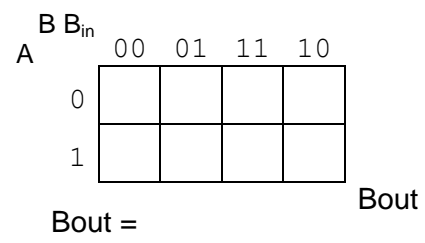
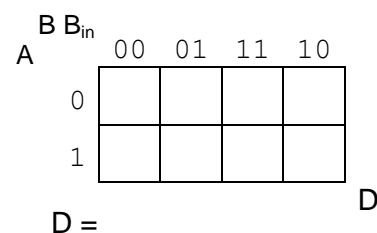


Problem 6. Sketch a Carry-Propagate-Generate circuit that can form the carry-ins for a 4-bit CLA.

Problem 7. Design a full-subtractor bit-slice circuit. Label the inputs A, B, and Bin, and label the outputs D and Bout. Start by completing the subtraction examples, then complete the truth table and K-maps, and then sketch the circuit.



A	B	Bin	D	Bout
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Problem 8: Complete the number conversions indicated. Note that all binary numbers are two's complement representations.

-19 = _____ 10011010 = _____

10000000 = _____ -101 = _____

Problem 9: Complete the four 2's complement arithmetic problems below, showing both the decimal and binary numbers in each case.

$$\begin{array}{r} 17 \quad 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1 \\ -11 \quad 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1 \\ \hline \end{array}$$

$$\begin{array}{r} -22 \\ + 6 \\ \hline \end{array}$$

$$\begin{array}{r} 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0 \\ - \quad 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1 \\ \hline \end{array}$$

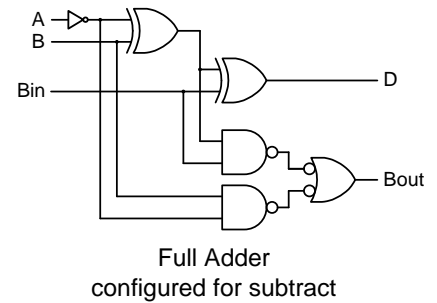
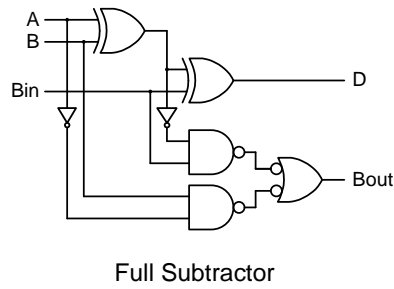
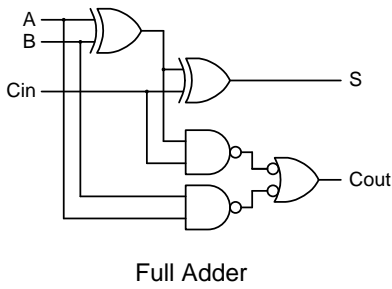
$$\begin{array}{r} 35 \\ - 42 \\ \hline \end{array}$$

$$\begin{array}{r} 19 \\ - -7 \\ \hline \end{array}$$

Is the above answer correct in 8 bits? Explain.

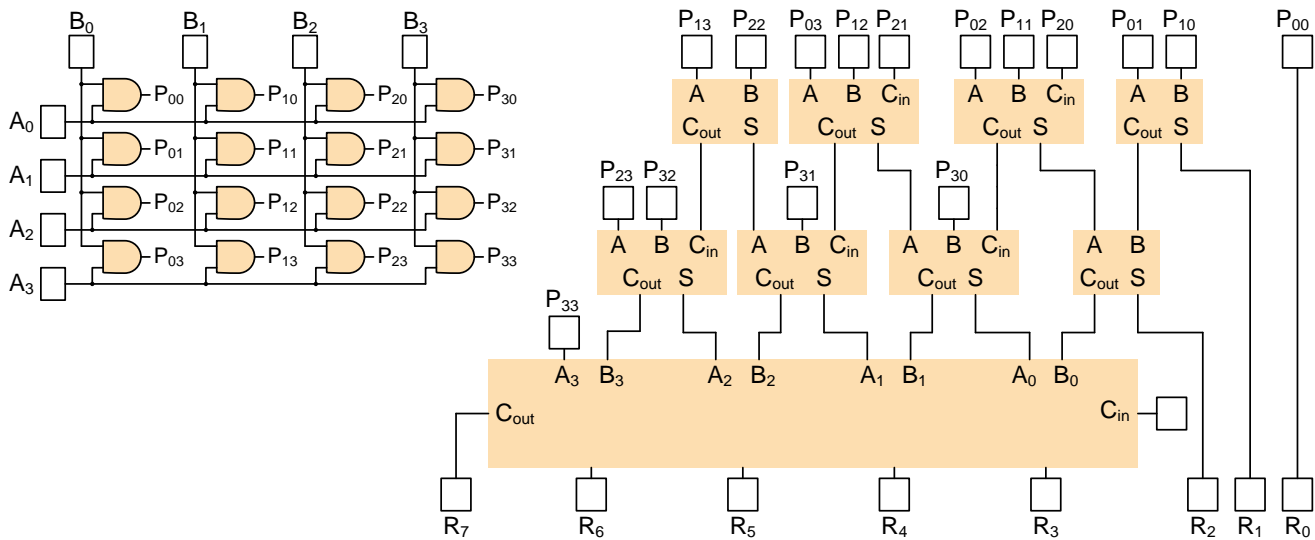
Problem 10: Sketch a circuit to convert a 4-bit binary number to its 2's complement representation. (Hint: can you use only 3 XOR/XNOR gates and 2 AND or OR gates?).

Problem 11: Explain how the circuit structures of a ripple-carry adder circuit configured as a 2's complement subtractor and a ripple-borrow subtractor perform identical functions.



Problem 12: Examine several examples of addition overflow and subtraction underflow, and sketch a circuit below that can output a '1' whenever an addition or subtraction result is incorrect due to underflow or overflow. (Hint: compare the carry in and carry out signals of the most-significant bit).

Problem 13. Fill in the squares below to show all signal values when “1101” and “1010” are multiplied.



Problem 14. Sketch a block diagram for a 4-bit ALU built from bit-slice ALU circuits that can implement the functions shown in the table. Label all signals, and recall that inputs to the bit slices must come from the A and B input busses as well as from neighboring bit slices (and outputs must drive the F output bus as well as neighboring bit slices). To design the signals that communicate information between slices, you must understand the ALU operations and the implications for information transfer (e.g., does the operation A PLUS B require that information be transferred between slices? If so, what? Does the operation A OR B require that information be transferred?).

Operation Code	ALU function
000	A PLUS B
001	A PLUS 1
010	A MINUS B
011	A MINUS 1
100	A XOR B
101	A'
110	A OR B
111	A AND B

Problem 15. In the ALU example in the module it was stated that an 8:1 mux could be use for the F output and a 4:1 mux could be used for the Cout output. Sketch the mux-based circuit.

Problem 16. Transfer the ALU operation table from the module has been reproduced below, but opcode 3 has been redefined as “decrement”. Complete the F and Cout table entries to define the decrement logic functions.

Op Code	Function	F	Cout
000	A PLUS B	$A \text{ xor } B \text{ xor } \text{Cin}$	$(A \text{ and } B) \text{ or } (\text{Cin and } (A \text{ xor } B))$
001	A PLUS 1	$A \text{ xor } \text{Cin}$	$A \text{ and } \text{Cin}$
010	A MINUS B	$A \text{ xor } B \text{ xor } \text{Cin}$	$(A' \text{ and } B) \text{ or } (\text{Cin and } (A \text{ xor } B)')$
011	A MINUS 1		
100	A XOR B	$A \text{ xor } B$	0
101	A'	A'	0
110	A OR B	$A \text{ or } B$	0
111	A AND B	$A \text{ and } B$	0

Problem 17. Write VHDL statements to define the ALU's operation.