

CSE140 - HW #5

Due Monday June 4, 2018, 11:59PM

We practice the standard module designs and applications. For the first problem, we use subtraction to rehearse the design process of adders. For the second and third problems, we use examples to comprehend the mechanism of prefix and carry save adders. For the last problem, we use standard modulo counters to construct various counters.

1. Subtractor: A full subtractor inputs three bits (x, y, b_{in}) , and outputs two bits (b_{out}, d) , to perform the subtraction $x - y$ with borrow-in b_{in} and generate the borrow-out b_{out} with the difference d .

1.1. Design the subtractor with minimal sum of product expressions. Describe the design using Boolean expressions.

1.2. Design a four-bit borrow look ahead subtractor, i.e. inputs two four bits binary numbers (x_3, x_2, x_1, x_0) , (y_3, y_2, y_1, y_0) with borrow-in b_{in} and outputs borrow-out b_{out} and the difference (d_3, d_2, d_1, d_0) . Use the full subtractors and a minimal network of AND, OR gates. Describe your solution with logic diagrams (ref slide 22 of lecture 14).

2. Prefix Adder: Given a prefix adder in slide 27 of lecture 14, manually simulate the addition of two numbers.

2.1 Simulate the addition of 289 and 756. Write the value of $(P_{i,j}, G_{i,j})$ output of each black box. Show the sum $S_i = A_i \oplus B_i \oplus G_{i-1,-1}$ to verify the results. (hint: write directly on the schematic diagram of slide 27.)

2.2 Repeat 2.1 with the addition of 321 and 456.

3. Carry Save Adder: Given an eight bit carry save adder, manually simulate the addition of four numbers.

3.1 Simulate the addition of 28, 19, 11, 14. Write the table as slide 31 of lecture 14.

3.1 Simulate the addition of 17, 18, 31, 15. Write the table as slide 31 of lecture 14.

3.3. Prove that we need to advance the carry out by one digit only to add each number to the partial result.

4. Counter: Use standard modulo counters to design various counters.

4.1 Given a modulo 16 counter, construct a 3 to 10 counter.

4.2 Given a modulo 16 counter, construct a counter that counts in sequence 1, 2, 5, 6, 7, 0, 4 with minimal networks of AND, OR gates.

4.3 Construct a modulo 64 counter using modulo 8 counters.