

CSE140 - HW #5

Due Monday June 3, 2019, 11:59PM

We practice the standard module designs and applications. For the first problem, we use addition/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 practice system design processes.

1. Adder/Subtractor: A full adder/subtractor inputs four bits (a, b, c_{in}, t) , and outputs two bits (c_{out}, s) . If bit $t = 0$, the operator functions as a full adder on input (a, b, c_{in}) . Otherwise, the operator functions as a full adder on (a, b', c_{in}) (one's complement).

1.1. Write the truth table of the operator.

1.2. Derive a minimal network of AND, XOR, OR gates assuming that NOT gates are free in terms of area and delay. Describe your solution with logic diagrams and explain why this solution is minimal in the following sequence of priority: first the performance (level of gates) and second the area (number of gates).

2. Prefix Adder: Given a prefix adder in figure 16 of the paper by Zhu, et al. [1] (posted with hw5 in the class website) manually simulate the addition of two numbers using the first 15 bits of the diagram assuming that the primary carry-in is 0.

2.1 Simulate the addition of 238 and 675. 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.

2.2 Repeat 2.1 with the addition of 432 and 521.

3. Carry Save Adder: The carry save adder in lecture 14 adds two numbers for each addition operation. Design a carry save adder that adds five numbers for each addition operation.

3.1 Write your number system and algorithm (hint: pages 29 and 34 in lecture 14).

3.2 Simulate the addition of 12, 7, 9, 10, 17, 11, 5, 14, 16. Write the table as slide 31 in lecture 14.

3.3 Prove that we need to advance the carry out by two digits only for each operation.

4. System Designs: Implement the following algorithm:

Alg(X,Y,Z,start,U,done)

Input X[7:0], Y[7:0], Z[7:0], start;

Output U[7:0], done;

```

Local-object A[7:0], B[7:0], C[7:0];
S1: If start' goto S1;
S2: done  $\leftarrow$  0  $\parallel$  A  $\leftarrow$  X  $\parallel$  B  $\leftarrow$  Y  $\parallel$  C  $\leftarrow$  Z;
S3: A  $\leftarrow$  Add(A,B)  $\parallel$  B  $\leftarrow$  Inc(B);
S4: If B'[7] goto S3  $\parallel$  C  $\leftarrow$  Inc(C);
S5: If C'[7] goto S3  $\parallel$  A  $\leftarrow$  Inc(A);
S6: U  $\leftarrow$  A  $\parallel$  done  $\leftarrow$  1  $\parallel$  goto S1;
End Alg

```

4.1 Design a data subsystem that is adequate to execute the algorithm and draw the schematic diagram.

4.2 Design a control subsystem and draw the state diagram.

4.3 Implement the control subsystem with a one hot encoding design. Draw the logic diagram.

References

- [1] Zhu, H., Cheng, C.K. and Graham, "On the construction of zero-deficiency parallel prefix circuits with minimum depth." ACM Transactions on Design Automation of Electronic Systems (TODAES), 11(2), pp.387-409.