CSE140 Exercise 6

1. Adders: Prove that for two's complement number system arithmetic, the overflow of the addition is determined by the last two carry bits, i.e. $overflow\_flag = c_n \bigoplus c_{n-1}$.
2. Adders: A carry look ahead adder inputs two-bit numbers (a_1, a_0) and (b_1, b_0), and a carry in c_0. Use a minimal two-level NAND gate network to implement the carry out c_2.
3. Subtracter: A subtracter inputs a two-bit number (x_1, x_0), a subtrahend (y_1, y_0) and a borrow-in bit b_0, and outputs the difference (d_1, d_0) and a borrow-out bit b_2.
3.1. Write the boolean expression of borrow-out bit b_2 as a function of variables x_1, x_0, y_1, y_0, b_0.
3.2. Use two full adders and a minimal number of AND, OR, NOT gates to implement a look-ahead subtracter. Draw the schematic diagram.
4. Serial Adders: A sequential adder inputs a_i, b_i, the i'th bit of two binary numbers in each clock cycle for $i = 0$ to $n - 1$ and outputs the sum s_i. Implement the adder with a JK flip-flop, and a minimal AND-OR-NOT network (if the network is needed). Draw the schematic diagram.
5. Counters: Given modulo-16 counters, draw the logic diagram to show the following designs.
5.1. Design a module-200 counter with a repeated output.
5.2. Design a counter with a repeated output sequence 15, 0, 1, 2, 8, 9, 10, 6, 7, with a modulo-16 counter and a minimal combinational network.Write the Boolean expression and draw the schematic diagram.
6. Design a counter with a repeated output sequence 0, 1, 2, 4, 5, 6, 3, with a modulo-8 counter and a minimal AND-OR-NOT network. Write the Boolean expression and draw the schematic diagram.
7. 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: $done <= 0$ || $A <= X$ || $B <= Y$ || $C <= Z$;
S3: $A <= Add(A, B)$;
S4: If $B'[7]$ goto S3 || $B <= Inc(B)$;
S5: If $C'[7]$ goto S3 || $C <= Inc(C)$;
S6: $U <= A$ || $done <= 1$ || goto S1;
End Alg
7.1. Design a data subsystem that is adequate to execute the algorithm. Draw the schematic diagram to show the design.
7.2. Design the control subsystem (i) draw the state diagram; (ii) draw the logic diagram that implements the control subsystem with a one hot encoding design.