CSE140 Exercises, Spring 2002

I. A sequential machine outputs a 1 when the input sequence \( x(t-1, t) = 00 \) or 10, otherwise the output is 0.

(1). Draw the state diagram and write the state table.

(2). Design the system with 2 T flip-flops and a minimal number of 4-input multiplexers.

(3). Design the system with 2 J-K flip-flops and a minimal two-level NAND network.

II. Adder:

A \( k \)-bit adder has inputs \((a_{k-1}, \ldots, a_1, a_0), (b_{k-1}, \ldots, b_1, b_0)\), and \( c_{in} \); and outputs \((s_{k-1}, \ldots, s_1, s_0), \text{and } c_{out}.\)

(1) Design a 5-bit adder with a single 4-bit adder and a minimal two-level network of NAND gates. Draw the logic diagram.

(2) Design a 5-bit adder with three 2-bit adders and a minimal two-level network of NOR gates. Draw the logic diagram.

III. Adders: Draw the logic diagram to show the following designs.

(1). Design a full adder with two half adders and a minimal number of 2:1 multiplexers.

(2). 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. Implement the adder with a JK flip-flop and a minimal two level NAND network.

IV. System Designs:

Implement the following algorithm:

Alg: { Input \(X<0:7>\), \(Y<0:7>\) type bit-vector,

\[\begin{align*}
&\text{start type boolean;}
&\text{Local-object } A<0:7>, B<0:7>, C<0:7> \text{ type bit-vector;}
&\text{Output } Z<0:7> \text{ type bit-vector,}
&\text{done type boolean; }
\end{align*}\]

Wait: \text{done<- 1 || C<- 0 || if \text{'start'} goto Wait;}

W1: \text{done<- 0 || A<- X || B<- Y || Z<- 0 || C<- Add(C, 1)}

Loop: \text{B<- Add(A,B)}

L1: \text{if B<7> goto Pass;}

L2: \text{Z<- Add(Z, 1);}

Pass: \text{C<- Add(C, C);}

P1: \text{if C<7> goto Wait else goto Loop;}

end Alg

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

(2). Design the control subsystem (i) draw the state diagram; (ii) implement the control subsystem with a one hot encoding design. Draw the logic diagram.

V. Given a modulo-16 counter

(1). Design a 3-to-13 counter with a minimal AND-OR-NOT network.

(2). Design a counter with output sequence 0, 1, 11, 2, 3, 13, 14, 15 with a minimal two-level network of NAND gates. Assume that the machine is reset to 0 initially. Draw the logic diagram.

(3). Design a modulo-64 counter with a modulo-16 counter, two SR flip-flops and a minimal network of AND, OR, NOT gates.