交電
100-1期末

Switching Circuits & Logic Design, Fall 2011

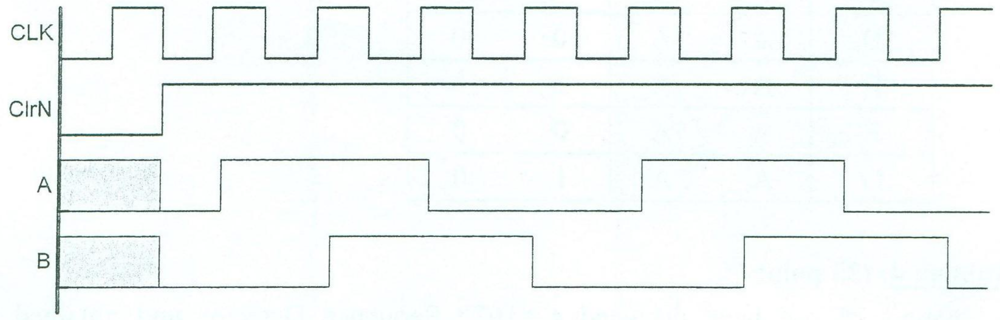
Final Examination (1/13/2012, 3:30pm~5:20pm)

Problem 1: (15 points)

Consider a new FF with three inputs, S, R, and T. No more than one of these inputs can be 1 at any time. The S and R inputs behave exactly as they do in an SR-FF. The T input behaves as it does in a T-FF. Please write the characteristics equation of such a new FF. Please simplify the equation to minimum SOP. (7%) Please construct such a FF with a T-FF and extra logic gates. (8%)

Problem 2: (20 points)

Design a periodical waveform generator which generates the following waveforms (period = 4 cycles):



(a) This waveform generator can be designed as a sequential circuit with two positive-edge triggered flip-flops. Derive the state graph of this circuit. (Hint: please present the state as AB. For example, AB=00 can be presented as (00) in the state graph.) (3%)

(b) Realize the circuits with one positive-edge triggered JK flip-flop for A, one positive-edge triggered T flip-flop for B, and basic logic gates. (AND, OR, NOT) Please use minimum SOP to realize your circuit. (12%)

(c) Modify the circuits developed in (b) by adding one new control pin "cnt," whose function is shown as the following table: (5%)

1/16 (一) 18:30 ~ 20:00

143

cnt	A^+B^+
0	AB (no change)
1	Generate the waveform

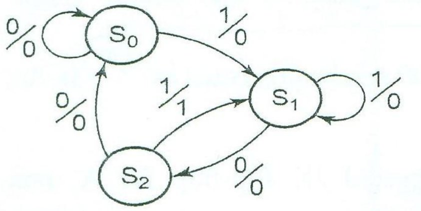
Problem 3: (5 points)

Reduce the following state table to a minimum number of states with row matching technique:

Present	Next State		Present	
State	X=0	1	Output (Z)	
A	B	C	0	0
B	D	E	0	0
C	F	G	0	0
D	A	A	0	0
E	A	A	1	0
F	A	A	0	0
G	A	A	1	0

Problem 4: (25 points)

In Chapter 14, we have designed a "101" Sequence Detector and obtained the following Mealy State Graph and State Table:



Present State	Next State $X=0$	$X=1$	Present Output $X=0$	$X=1$
S_0	S_0	S_1	0	0
S_1	S_2	S_1	0	0
S_2	S_0	S_1	0	1

(a) Try to complete the circuit design by using the following state assignments: S_0=00, S_1=01 and S_2=11, two D-type flip-flops and necessary logic gates. Show your transition table and K-maps, then, plot the final circuit with minimum SOP. (13%)

	AB	A^+B^+		Z (output)	
		X=0	1	X=0	1
S_0	00				
S_1	01				
S_2	11				
	10	x	x	x	x

(b) Try to complete the circuit design by using the following (one-hot) state assignments: S_0=001, S_1=010 and S_2=100, three D-type flip-flops (one for each State) and necessary logic gates. Show your transition table, find the three flip-flop inputs and output Z directly from the table (without simplification), and plot the final circuit diagram. (12%)

	ABC	$A^+B^+C^+$		Z (output)	
		X=0	1	X=0	1
S_0	001				
S_1	010				
S_2	100				

Problem 5: (10 points)

Determine if the following statements are correct (true/false questions). If false, please correct the statement(s) with proper explanations.

When implementing a sequential circuit design using 4 flip-flops and a ROM:

(a) state assignment in straight binary order is as good as any other choices; (2%)

(b) J-K flip-flops are preferable to D flip-flops; (2%)

(c) either Moore-typed or Mealy-typed machine can be implemented; (2%)

(d) the sequential circuit can have at most 4 different states; (2%)

(e) positive-edge triggered flip-flops are less preferred because they may cause false outputs. (2%)

Problem 6: (15 points)

A *serial average circuit* is similar to the *serial accumulator* in the textbook except that the new value of *X* is the average of X and Y. Assume both X and Y are 4-bit positive integers. For example, initially X=$0010_2=2_{10}$ and Y=$1001_2=9_{10}$, then new X = $0101_2=5_{10}$. (Please note that 0.5 in the average is rounded off.) For another example, $X=1111_2=15_{10}$ and $Y=1111_2=15_{10}$, the new $X=1111_2=15_{10}$

(a) Use the following components defined in the textbook, draw your design of the serial average circuit. Please assume that both X and Y are initially loaded in the register before the start signal. Only block diagram is enough. You can add one AND gate if needed. (10%)

X register: 4-bit right shift register with *Shift input* (SI), *Shift* (Sh), *Shift output* (SO).

Y register: 4-bit right shift register with *Shift input* (SI), *Shift* (Sh), *Shift output* (SO). Note: We do NOT need to keep the number Y after the operation.

One-bit serial adder: It has two inputs *a, b*, and outputs *sum* (s). Assume carry is initially zero.

Controller: A Mealy circuit that has one input (*St*) and two outputs.

(b) Please draw the state graph of the controller. No circuit design is needed. (5%)

Problem 7: (10 points)

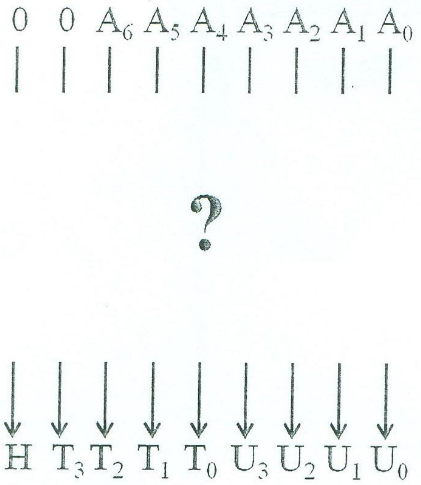
You are asked to design a circuit to compare two 2500-symbol long sequences collected. There are only four different symbols, A, T, C, and G. The comparison process starts from the first symbol, and the process repeats until all the symbols in the sequences have been compared. If two sequences are identical, the final output is 1. Otherwise, the final output is 0.

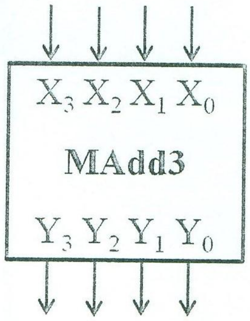
(a) Please draw the block diagram for the iterative circuit that consists of identical cells. Label the numbers of input/output bits for each cell. However, you don't have to provide detailed circuit design within the cell. (5%)

(b) Please provide the state table for a single cell. You can use symbolic inputs in the table, for example, X_1X_2=AT. (5%)

Bonus Problem: (5 points)

The following Verilog code describes a 7-bit binary-to-BCD converter as in our Verilog LAB. Draw the circuit and label each MAdd3 module corresponding to the instance name in the code. The order of input ports and output ports in each MAdd3 module should follow the rule in the figure.


0 0 A_6 A_5 A_4 A_3 A_2 A_1 A_0
?
H T_3 T_2 T_1 T_0 U_3 U_2 U_1 U_0



Mbinary2BCD.v

```
module Mbinary2BCD ( a, h, t, u );
  input [6:0] a;
  output h;
  output [3:0] t;
  output [3:0] u;

  wire [3:0] in1;
  wire [3:0] out1;
  assign in1 = {1'b0, a[6:4]};
  MAdd3 add1 (.x(in1), .y(out1));
  wire [3:0] in2;
  wire [3:0] out2;
  assign in2 = {out1[2:0], a[3]};
  MAdd3 add2 (.x(in2), .y(out2));
  wire [3:0] in3;
  wire [3:0] out3;
  assign in3 = {out2[2:0], a[2]};
  MAdd3 add3 (.x(in3), .y(out3));
  wire [3:0] in4;
  wire [3:0] out4;
  assign in4 = {1'b0, out1[3], out2[3], out3[3]};
  MAdd3 add4 (.x(in4), .y(out4));
  wire [3:0] in5;
  wire [3:0] out5;
  assign in5 = {out3[2:0], a[1]};
  MAdd3 add5 (.x(in5), .y(out5));

  assign u[0] = a[0];
  assign u[3:1] = out5[2:0];
  assign t[0] = out5[3];
  assign t[3:1] = out4[2:0];
  assign h = out4[3];
endmodule
```