C SCI 113 INTRODUCTION TO COMPUTER ORGANIZATION
Spring 2002

MIDTERM EXAMINATION 2

Name: ______________________________ April 11, 2002, 12:30 - 13:45

I.D. No. ______________________ Total: **40 points**

Problem 1 (18 points)

Design a multi-functional counter, ABC, using JK flip-flops and combining the ***shift*** and ***count*** operations in a single sequential circuit. The count operation counts in the following counting sequence:

$$ABC = 000 \rightarrow 100 \rightarrow 110 \rightarrow 111 \rightarrow 011 \rightarrow 001 \rightarrow 010 \rightarrow 000 \text{ (repeat)}$$

The overall function table is given below:

s_1	s_0	Function	Comment
0	0	Clear	**not** using Preset//Clear input of the flip-flop
0	1	Count	count in the above-given sequence
1	0	Left rotate	rotate left the code ABC in the counter
1	1	Right rotate	rotate right the code ABC in the counter

(a) Write the state-transition table and design the excitation table applicable to JK flip-flops.

(b) Check the self-correcting property of the counter and derive the simplest expressions for the excitation signals for the count operation alone.

(c) Derive the simplest expressions for the excitation signals for the entire circuit combining the shift and count operations.

Your expressions should satisfy the following conditions for simplest implementation:

- use only basic logic gates selected from AND, OR, NAND, NOR, XOR, and XNOR without inverted inputs; the number of inputs per gate should not exceed 3;
- The inverted variables $\overline{s_1}$ and $\overline{s_0}$ are not available. A good solution should not use them at all. However, the inverted variables $\overline{A}$, $\overline{B}$, and $\overline{C}$ are available from the flip-flops.
- The total number of gates should be minimum, not exceeding 27.

Problem 2 (15 points)

Design a 4-bit ALU based on an SN74181 ALU chip according to the following function table:

f_2	f_1	f_0	Function of the ALU	Meaning
0	0	0	OP1 plus Cin	transfer or increment
0	0	1	OP1 plus OP2 plus Cin	add without or with a carry
0	1	0	OP1 minus OP2 minus Cin	subtract without or with a borrow
0	1	1	OP2 minus OP1 minus Cin	inverse subtract without or with a borrow
1	0	0	OP1 minus 1	decrement by 1
1	0	1	OP1 $\wedge$ OP2	OP1 AND OP2
1	1	0	OP1 $\vee$ OP2	OP1 OR OP2
1	1	1	OP1 $\oplus$ OP2	OP1 XOR OP2

(a) Design a truth table such that it will allow the simplest expressions to be derived in Part (b).
Note: (1) The function tables of SN74181 are given on the back side of the previous page.
(2) Fill in the truth table in the following format:

f_2	f_1	f_0	A	B	$\overline{C_{-1}}$	M	s3	s2	s1	s0	Comments

(b) Derive the simplest expressions for all the dependent variables in the truth table. Your expressions must be maximally simplified for optimal implementation satisfying the following conditions:

- share as many common terms as possible;
- use only **2-input** basic gates selected from AND, OR, NAND, NOR, XOR and XNOR without inverted inputs;
- Only two inverted variables $\overline{f_1}$ and $\overline{f_0}$ are given, while all the other inverted variables, $\overline{f_2}$, $\overline{OP1}$, $\overline{OP2}$, and $\overline{C_{in}}$, are not available, so you should not use them;
- The total number of gates used by all the expressions must be minimum, not exceeding 8n+11, where n is the wordlength. (Continue on the back side of the previous page if necessary.)

Function table for SN74181 when M = 1

Function code s_3 s_2 s_1 s_0	Operations $F_i(A_i,B_i)$
0 0 0 0	$\overline{A_i}$
0 0 0 1	$\overline{A_i}\cdot\overline{B_i}$
0 0 1 0	$\overline{A_i}\cdot B_i$
0 0 1 1	0
0 1 0 0	$\overline{A_i}+\overline{B_i}$
0 1 0 1	$\overline{B_i}$
0 1 1 0	$A_i\oplus B_i$
0 1 1 1	$A_i.\overline{B_i}$
1 0 0 0	$\overline{A_i}+B_i$
1 0 0 1	$\overline{A_i\oplus B_i}$
1 0 1 0	B_i
1 0 1 1	$A_i\cdot B_i$
1 1 0 0	1
1 1 0 1	$A_i+\overline{B_i}$
1 1 1 0	A_i+B_i
1 1 1 1	A_i

Function table for SN74181 when M = 0

Function code s_3 s_2 s_1 s_0	Operations $F(A,B,C_{-1})$
0 0 0 0	A plus C_{-1}
0 0 0 1	(A+B) plus C_{-1}
0 0 1 0	(A+$\overline{B}$) plus C_{-1}
0 0 1 1	11...1 plus C_{-1}
0 1 0 0	A plus A$\overline{B}$ plus C_{-1}
0 1 0 1	(A+B) plus A$\overline{B}$ plus C_{-1}
0 1 1 0	A minus B minus $\overline{C_{-1}}$
0 1 1 1	A$\overline{B}$ plus 11...1 plus C_{-1}
1 0 0 0	A plus AB plus C_{-1}
1 0 0 1	A plus B plus C_{-1}
1 0 1 0	(A+$\overline{B}$) plus AB plus C_{-1}
1 0 1 1	AB plus 11...1 plus C_{-1}
1 1 0 0	A plus A plus C_{-1}
1 1 0 1	(A+B) plus A plus C_{-1}
1 1 1 0	(A+$\overline{B}$) plus A plus C_{-1}
1 1 1 1	A plus 11...1 plus C_{-1}

Problem 3 (7 points)

(a) Design a 4-bit shifter attached to the output of an ALU according to the following function table:

h_1h_0	Function	Comment
0 0	No shift	forward the ALU result, **F3-F0**, to the output of the shifter with no shift
0 1	External input	forward an external data, **I3-I0**, to the output of the shifter with no shift
1 0	Left logic shift	left shift the ALU result and a serial input, SIL, to the output of the shifter
1 1	Right logic shift	right shift the ALU result and a serial input, SIR, to the output of the shifter

(a) Draw the circuit diagram of the shifter, using h_1, h_0, F3-F0, I3-I0, SIL, and SIR as inputs and OUT_3-OUT_0 as output.

(b) The shifter constructed above can be used in different ways to perform different shift operations. ind cate in the above diagram how it can be connected to perform the following two operations:

(i) Arithmetic right shift one bit

(ii) Left rotate one bit