MALASPINA UNIVERSITY-COLLEGE
FINAL EXAMINATION DECEMBER 2005
COMPUTING SCIENCE CSCI 355

Instructor: Dr. P Walsh **Duration:** 3 Hours.

TOTAL VALUE	100				
QUESTION	1	2	3	4	5
VALUE	10	25	25	10	30

Instructions

- Students must count the number of pages in this examination paper before beginning to write, and report any discrepancy immediately to the invigilator.
- This examination paper consists of 3 pages not including this cover page.
- Answer all questions in an answer booklet. All questions relate to material covered in class or in the lab. Show all your work.
- This is a CLOSED BOOK - examination. Students may consult notes written on one 8in. by 11 in. sheet of paper as described in class.
- Calculators are NOT permitted.

QUESTION 1.

Let

$$F(A, B, C, D, E) = \Pi M(0, 1, 2, 3, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 24, 25, 26, 27, 30, 31)$$

(a) Find the minimal SOP expression for F.
(b) Find a hazard-free minimal SOP expression for F.
(c) Find the minimal POS expression for F.
(d) Implement F in CMOS (minimize the number of transistors).

QUESTION 2.

Consider the following state table (with state encodings). It has one input X and four states. Your task is implement this system using the following counter component. To this end, develop SOP equations for each of the counter inputs. Note, you may assume that initially, the counter is in state `0` and 'x' means dont-care.

Present State	Next State	
	X=0	X=1
00	01	10
01	10	11
10	10	11
11	11	00

```
component counter
        port (
        Q: OUT std_logic_VECTOR(15 downto 0);
        CLK: IN std_logic;
        LOAD: IN std_logic;
        L: IN std_logic_VECTOR(15 downto 0);
        CE: IN std_logic;
        SCLR: IN std_logic);
end component;
```

LOAD	CE	SCLR	Operation
1	x	x	load
0	1	0	count up
0	1	1	clear

QUESTION 3.

Consider the FSM specified by the following state table.

Present State $QA\ QB\ QC$	Next State Input=X		Output Z	
	X=0	X=1	X=0	X=1
000	100	101	1	0
100	111	110	1	0
101	110	110	0	1
111	011	011	0	1
110	011	010	1	0
011	000	000	0	1
010	000	xxx	x	x
001	xxx	xxx	x	x

Obtain excitation equation(s) for S-R flip-flops and output equation(s).
Note 'x' means dont-care.

QUESTION 4.

Consider the following state table. It has one input X and out output Z. Your task is to identify any equivalent states.

Present State	Next State Input=X		Output Z	
	X=0	X=1	X=0	X=1
a	f	b	0	0
b	d	c	0	0
c	f	e	0	0
d	g	a	1	0
e	d	c	0	0
f	f	b	1	1
g	g	h	0	1
h	g	a	1	0

QUESTION 5.

Consider the following VHDL custom processor component.

```
component customProcessor
        port (
        compute: IN std_logic;
        done: OUT std_logic;
        X: IN std_logic_VECTOR(15 downto 0);
        Y: IN std_logic_VECTOR(15 downto 0);
```

```
        Z: OUT std_logic_VECTOR(15 downto 0));
end component;
```

When `compute` is asserted, the processor performs the following actions:

- un-assert `done`
- input the values on `X` and `Y`
- calculate the product of the two inputs by repetitive addition
- output the product on `Z`
- assert `done`

Your task is to develop the control unit and datapath for the above processor. For the datapath, you may use components from the MU assignment *viz*, counters(s), busmux(s) and alu(s) (interfaces given in QUESTION 2 and below). Again, you must implement multiplication using repetitive addition (eg, 3*4 == 3+3+3+3). For the control unit, only the state transition and output table is required.

```
component busmux is
        port (
        MA: IN std_logic_VECTOR(15 downto 0);
        MB: IN std_logic_VECTOR(15 downto 0);
        S: IN std_logic_VECTOR(0 downto 0);
        O: OUT std_logic_VECTOR(15 downto 0));
end component;

component alu is
        port (
        A: IN std_logic_VECTOR(15 downto 0);
        B: IN std_logic_VECTOR(15 downto 0);
        ADD: IN std_logic;
        S: OUT std_logic_VECTOR(15 downto 0));
end component;
```

Hint: Keep the datapath simple and use 2 ALUs.