MALASPINA UNIVERSITY-COLLEGE
FINAL EXAMINATION DECEMBER 2007
COMPUTING SCIENCE CSCI 355

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

TOTAL VALUE	100				
QUESTION	1	2	3	4	5
VALUE	10	20	25	20	25

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. In addition, there is one handout labelled RTN Syntax.
- 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

$$F(A, B, C, D, E) = \Sigma m(5, 7, 13, 15, 21, 23, 29, 31)$$

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

QUESTION 2.

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 N-R flip flops and output equation(s). An N-R flip-flop has two inputs (N, R) and two outputs (Q, Q_{BAR}). Input R behaves like the R input to an S-R flip flop. Input N behaves like the complement of the S input to a S-R flip flop. Note 'x' means dont-care.

QUESTION 3.

Consider the state table from QUESTION 2. Your task is implement state transition 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 `00` and 'x' means dont-care.

```
component counter
        port (
        Q: OUT std_logic_VECTOR(2 downto 0);
        CLK: IN std_logic;
        LOAD: IN std_logic;
        L: IN std_logic_VECTOR(2 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 4.

Design a single-purpose processor that outputs *sigmaRX(n)* where

$$n \geq 0$$

$$sigmaRX(0) = 0$$

$$sigmaRX(1) = 1$$

$$sigmaRX(n) = n + sigmaRX(n-2)$$

For example, $sigmaRX(7)$=7+5+3+1=16 and $sigmaRX(8)$=8+6+4+2+0=20.

Start off with an iterative function computing the desired result. Then derive the datapath and control unit. Note, all registers are to be 8 bits wide.

QUESTION 5.

Consider a processor ISA where the data-bus is 16 bits wide and the address bus is 8 bits wide. The instruction format splits a 16 bit word into an 8 bit opcode followed by an 8 bit address. The processor contains the following registers:

- Accumulator ACC (16 bits wide)
- Program Counter PC (8 bits wide)
- Memory Index register MIR (16 bits wide)
- Instruction Register IR (8 bits wide)

The processor's instruction set is as follows:

Opcode	Mnemonic	Pseudocode
0x0	LDA S	ACC := [S]
0x1	STO S	[S] := ACC
0x2	ADD S	ACC := ACC + [S]
0x3	SUB S	ACC := ACC - [S]
0x4	JMP S	PC := S
0x5	JGE S	if ACC $\geq$ 0, PC := S
0x6	JNE S	if ACC $\neq$ 0, PC := S
0x7	LDI S	MIR := [S]
0x8	STI S	[S] := MIR
0x9	ADI S	ACC := ACC + [S + MIR]
0xa	SUI S	ACC := ACC - [S + MIR]

Your task is to develop an abstract RTN specification for the described processor. You are free to add signals such as reset, fault and halt to your RTN.