BILKENT UNIVERSITY
Department of Electrical and Electronics Engineering
EEE102 Introduction to Digital Circuit Design
Final Exam

24-05-2007

Duration 150 minutes

Surname: ________________________________

Name: ________________________________

ID-Number: ________________________________

Signature: ________________________________

There are 9 questions. Solve all.
Do not detach pages. Show all your work.

Q1	
Q2	
Q3	
Q4	
Q5	
Q6	
Q7	
Q8	
Q9	
Total	

Q1) (15 points)

a) What is a bistable element? Briefly explain. Draw an example circuit.

b) What is the formal definition of a counter?

c) Explain the fundamental difference between sequential logic circuits and combinational logic circuits.

d) What is the difference between DRAM and SRAM.

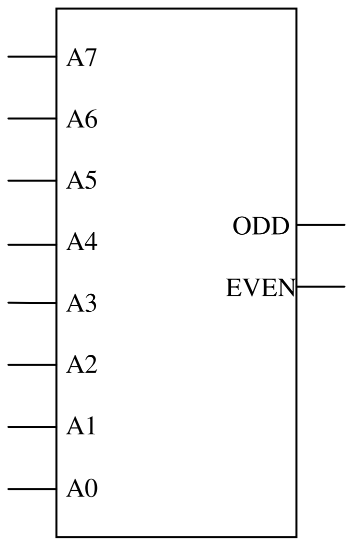
e) What kind of logic components are there in a configurable logic block (CLB) of a XILINX FPGA? (Just name the logic components.)

Q2) (10 points) State the fundamental difference between a D-latch and a D-flip flop. Write VHDL modules one for a D-latch and another for a D-flip flop.

Q3) (10 points)
Implement the following function by using two 16x1 RAMs and one 8x1 RAM. No other components are allowed.

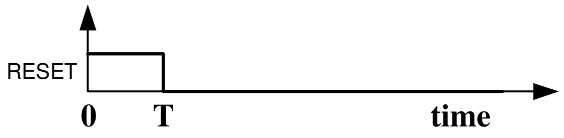
A4	A3	A2	A1	A0	F
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	1	0

Q4) (10 points)
Design an 8-bit parity circuit by using two-input logic gates only.

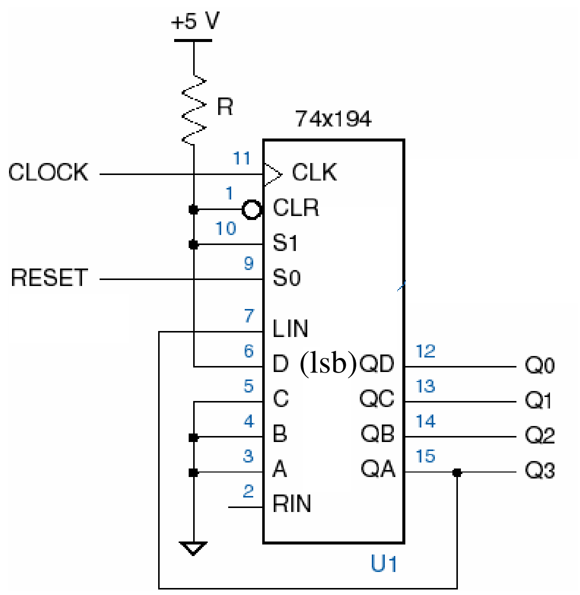


Q5) (10 points)
What is the counting sequence of the below ring counter assuming that the following RESET signal is applied to S0. Is this counter self correcting? If yes, show that it is self correcting. Otherwise, modify the circuit so that it becomes self correcting.

S1	S0	Function
0	0	LAST Q
0	1	Shift Right
1	0	Shift Left
1	1	LOAD



T is sufficiently long to accomodate at least one clock tick



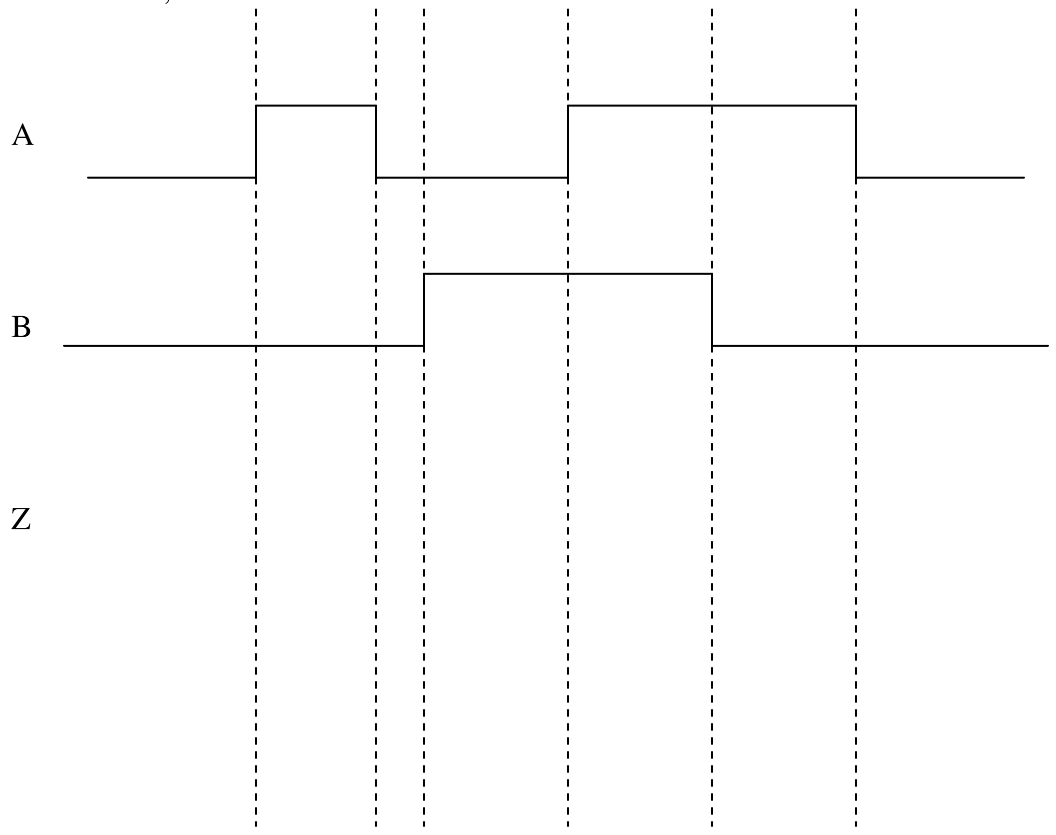
Q6) (15 points)
Design and draw a two input (X, Y), and one output (F) FSM that works as the following:

If X>Y, then FSM goes to state 1
If X<Y, then FSM goes to state 2
If X=Y, then FSM goes to state 3
F = 1 if X=Y at the last clock tick and also at present.
Power-on state is state 0.

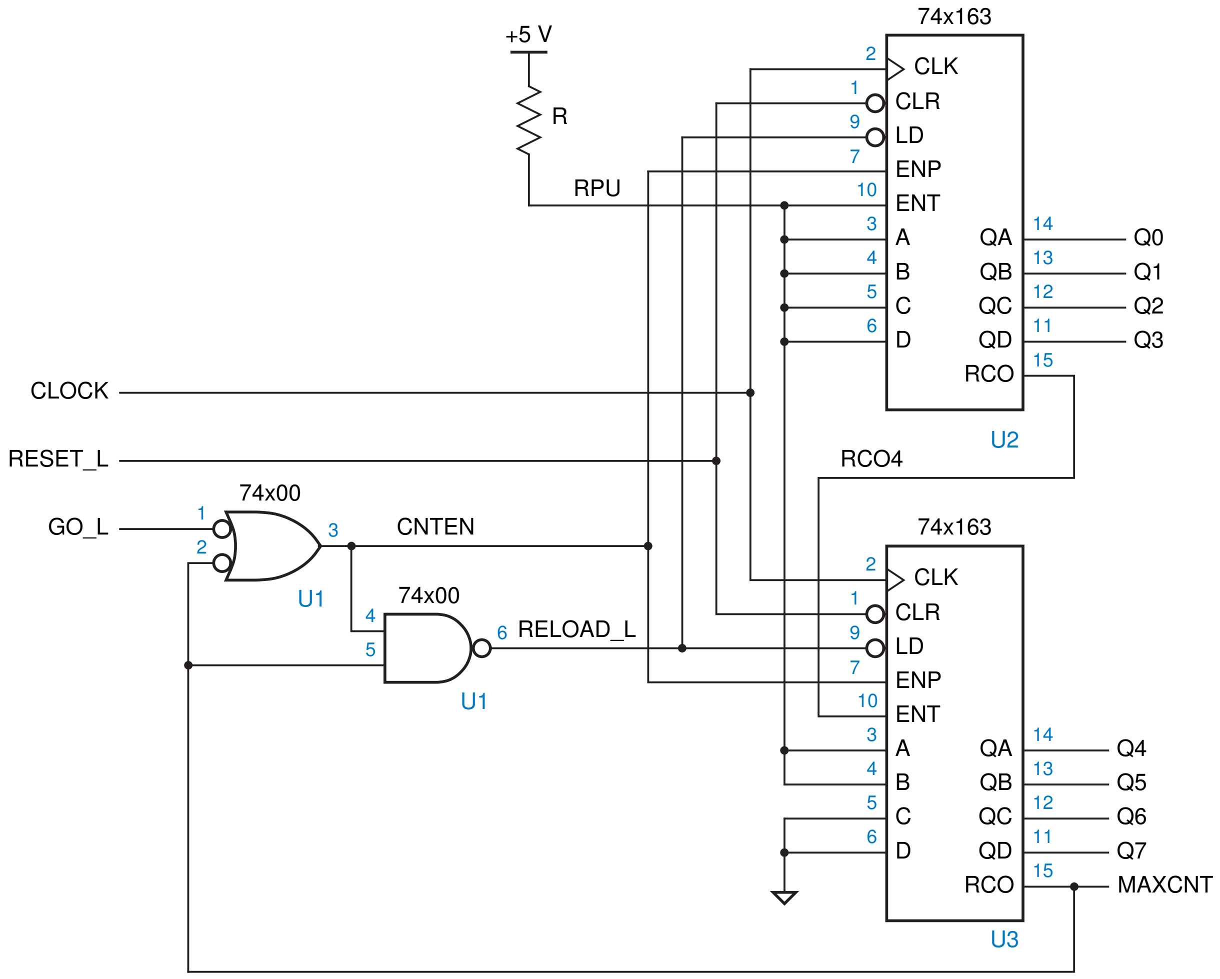
Show all your design steps.

Q7) (10 points) For the following code draw the simulation waveform of Z for the below given waveforms of A and B.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity circuit is
   Port ( A : in  STD_LOGIC;
          B : in  STD_LOGIC;
          Z : out  STD_LOGIC);
end circuit;

architecture Behavioral of circuit is
signal C: std_logic;
begin
        process(A,B)
        begin
                C <= A and B;
                Z <= not C;
        end process;
end Behavioral;
```



Q8) (10 points) For the circuit below the following VHDL code is written.



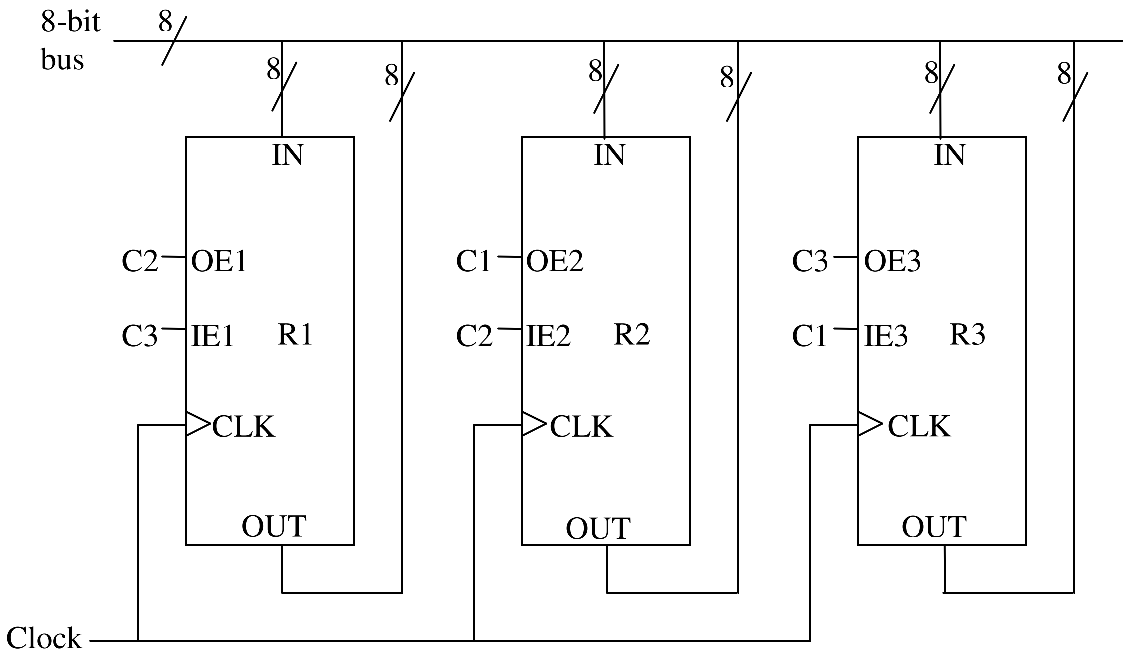
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity modulo_193_counter is
   Port ( CLOCK : in  STD_LOGIC;
          RESET_L : in  STD_LOGIC;
          GO_L : in  STD_LOGIC;
          Q : out unsigned (7 downto 0));
end modulo_193_counter;

architecture Behavioral of modulo_193_counter is
component counter_163 is
   Port ( CLK : in  STD_LOGIC;
          CLR_L : in  STD_LOGIC;
          LD_L : in  STD_LOGIC;
          ENP : in  STD_LOGIC;
```

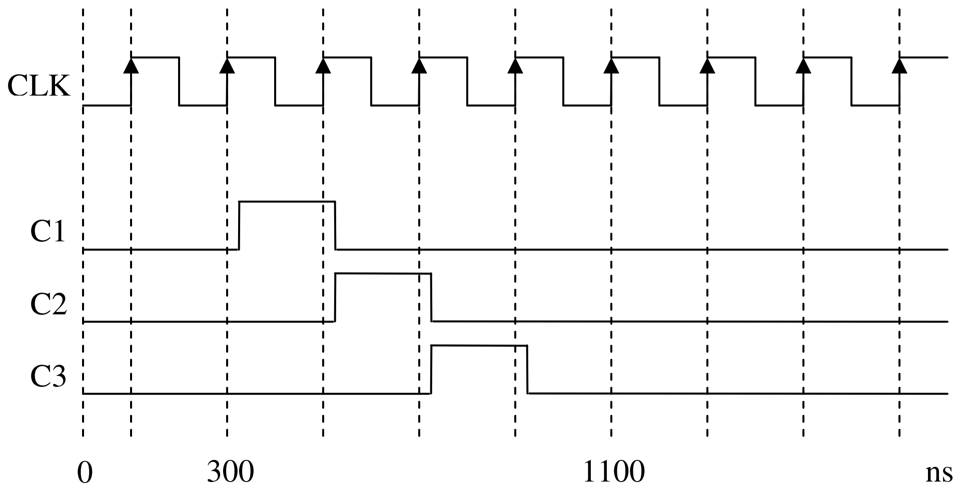
```
        ENT : in  STD_LOGIC;
        D : in  unsigned (3 downto 0);
        Q : out  unsigned (3 downto 0);
        RCO : out  STD_LOGIC);
end component;
signal MAXCNT,CNTEN,RELOAD_L: std_logic;
begin
L1:counter_163 port map (CLOCK, RESET_L, RELOAD_L, CNTEN, RCO, "1100", Q,
MAXCNT);
L2:counter_163 port map (CLOCK, RESET_L, RELOAD_L, CNTEN, '1', "1111", Q,
RCO);
CNTEN<=not GO_L or not MAXCNT;
RELOAD_L<= not (CNTEN and MAXCNT);
end Behavioral;
```

Find and correct the mistakes in this code. Explain your corrections.

Q9) (10 points)
We have an 8-bit bus and three 8-bit positive edge triggered registers, R1, R2, and R3 connected to it as shown in the figure below. The register outputs are three-state and they have active-high output enable (OE) controls. The registers also have active-high input enable (IE) controls which gate their clock inputs. The contents of the registers R1, R2, and R3 are "00001111", "10101010", and "11100011" respectively at time 0.
The waveforms of the signals C1, C2, and C3 are shown below.

At time 1100 ns what are the contents of the registers? Explain your reasoning. Only writing the contents will not be graded.





Some VHDL Templates:

ENTITY DECLARATION

```
entity entity_name is
        generic ( constant_names : constant type;
                  constant_names : constant type;
                  …
                  constant_names : constant type);
        port (  signal_names : mode signal_type;
                signal_names : mode signal_type;
                …
                signal_names : mode signal_type);
end entity_name;
```

ARCHITECTURE DEFINITIONS

```
architecture architecture-name of entity-name is
        type declarations
        signal declarations
        constant declarations
        function definitions
        procedure definitions
        component declarations
 begin
        concurrent statement
        ...
        concurrent statement
end architecture-name;
```

COMPONENT DECLARATION

```
component component_name
      port ( signal_names : mode signal type;
             signal_names : mode signal type;
             …
             signal_names : mode signal type);
end component;
```

COMPONENT INSTANTIATION

```
label: component_name port map (signal1, signal2, …,signaln);
or,
label: component_name port map (port1 =>signal1, port2 =>signal2, …, portn =>signaln);
```

DATAFLOW TYPE STATEMENTS:

Simple concurrent assignment statement

```
signal_name <= expression;
```

Conditional concurrent assignment statement

```
signal_name <=
     expression when boolean-expression else
     expression when boolean-expression else
     …
     expression when boolean-expression else
     expression;
```

with-select statement

```
with expression select
      signal_name <= signal_value when choices,
                     signal_value when choices,
                     …
                     signal_value when choices;
```

Note that conditional concurrent assignment statement and with-select statement cannot be used in a process statement. Instead, in a process, one can use the sequential conditional assignment statements if and case.

BEVAVIORAL TYPE STATEMENTS:

process statement
process(*signal_name*, *signal_name, …, signal_name*)
    type_declarations
    variable declarations
    constant declarations
begin
    sequential-statement
    …
    sequential-statement
end process;
Simple sequential assignment statement
signal_name <= expression;
if statement in its general form
if *boolean_ expression* then *sequential_statements*
elsif *boolean_ expression* then *sequential_statements*
…
elsif *boolean_ expression* then *sequential_statements*
else *sequential_statements*
end if;
Note that you may not use the else and/or the elsif.
case-when statement
case *expression* is
    when *choices* => *sequential_statements*
    …
    when *choices* => *sequential_statements*
end case;
loop statement
loop
    sequential_statement
    …
    sequential_statement
end loop;
for-loop statement
for *identifier* in *range* loop
    sequential_statement
    …
    sequential_statement
end loop;
while statement
while *boolean_expression* loop
    sequential_statement
    …
    sequential_statement
end loop;

Note that the if, case, loop, for, and while statements are called sequential statements and they can only be used in a process statement. Also note that each process is one concurrent statement.

Concatenation operator & is used as follows: If A and B are 2 bit numbers then A&B is a four bit number with A being more significant**.**