German University in Cairo - GUC
Faculty of Media Engineering and Technology- MET
Assoc. Prof. Milad Ghantous

CSEN605 – Digital System Design

Winter Semester 2020/2021

Final Exam

Bar Code

Tick your major

☐ CS ☐ DMET

Instructions: **Read Carefully Before Proceeding.**

1- Closed book examination. Non-programmable calculators are allowed
2- Write your solutions in the space provided
3- The exam consists of **(XX) questions**
4- This exam booklet contains **(XX) pages** including this page, and the scratch sheet.
5- Total time allowed for this exam is **(180) minutes**
6- When you are told that time is up, stop working on the test

Good Luck!

Question	1	2	3	4	5	∑
Possible Marks	15	15	20	30	20	100
Final Marks						

Question 1 – (15 pts)

Multiple choice questions.

Mark the correct answer at the bottom of the page.

1) Consider a 12-bit counter-up driven by a 1 MHz clock and output C(11 downto 0). Which output has a frequency of approximately 31 KHz.
a) C(3)
b) C(4)
c) C(5)
d) C(6)

2) Consider a generic register with generic length N, having a default value of 8. How to initialize the output Q to zeros.
a) `Q <= "00000000"`
b) `Q => (others <= '0' )`
c) `Q <= (others =>'0' )`
d) `Q <= "all => '0' "`

3) We can integrate a generic component into another bigger architecture without specifying the `generic map`.
a) True
b) False

4) The **setup time** is the time the input must hold still in it before the clock edge.
a) True
b) False

5) The **hold time** is the time the output must hold still after the clock edge.
a) True
b) False

	1	2	3	4	5
Your answer	**B**	**C**	**A**	**A**	**B**

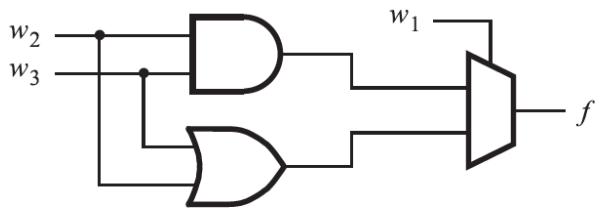
6) If **reset** was included in the process with the clock (process(reset, clock)), this means:
a) The reset is synchronous
b) The reset is asynchronous
c) Could be both

7) What's the behavior of the following code assuming Q was '1' initially.

```
process
begin
wait until clk'event and clk='0';
Q <='0';
Q <='1';
End process;
```

a) On the rising edge of the clock, Q turns off then turns on.
b) On the rising edge of the clock, Q remains on.
c) On the falling edge of the clock, Q turns off then turns on.
d) On the falling edge of the clock, Q remains on.

8) What expression does represent the following mux circuit?



a) w1(w2w3) +w1'(w2w3)
b) w1(w2+w3) + w1'(w2+w3)
c) w1'w2w3 + w1w2 + w1w3

9) The '`event` attribute can be only used to detect clock edges.
a) True
b) False

10) TcQ is the time taken for the input to affect the output after the clock edge.
a) True
b) False

	6	7	8	9	10
Your answer	C	D	C	B	A

Question 2 – (15 pts)

A) If the clock was included in the `process (…, clock),` what's the purpose of using `clock'event` inside and can we detect a rising edge by just using `if clock='1` inside the process?

no, not enough because the process might contain other signals that might trigger the process activity which might be in the middle of the clock period. The if clock=1 would be true indicating as if there was an edge even though it's not.

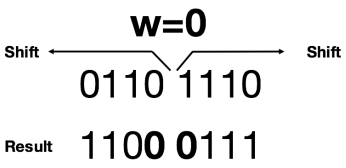
2) Write a function called truncate that takes an 8-bit input A and an integer N. The output Q is an 8-bit number equal to A but with the lower N bits set to 0. If N is larger than 8, Q is set to all zeros.

```
FUNCTION truncate (A: STD_LOGIC_VECTOR(7 downto 0); N: integer)
RETURN  STD_LOGIC_VECTOR    IS
variable trunc_out : STD_LOGIC_VECTOR(7 downto 0) := A;


BEGIN
 If N <=8 then


   For i IN 0 to N-1 loop
   trunc_out(i) := '0';
   End loop;
Else trunc_out := "00000000";
return trunc_out;


END truncate;
```

Question 3 – (20 pts)

Construct a Generic **N**-bit Custom shift register that shifts to the right the least significant half of the output **Q**, and shifts to the left the most significant half, both by the input **w**, and only if enable **En** is 1. The **clock** is positive edge triggered. Also there is an asynchronous input **L**, if set to '1', the register is loaded with the N-bit input **R**. Assume N is always even.

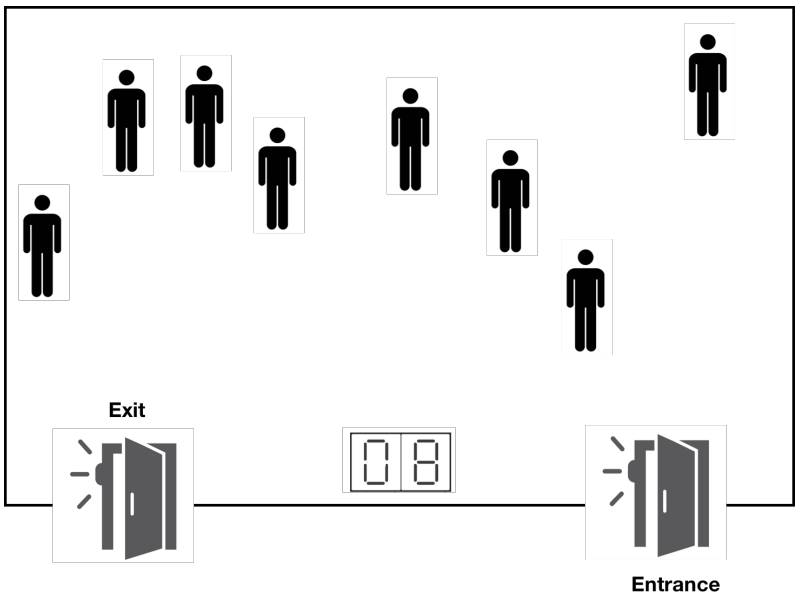
Example: Register is loaded such that Q= "0110 1110". When En='1' and w='0', the output Q becomes: 1100 0111. (Look at figure)



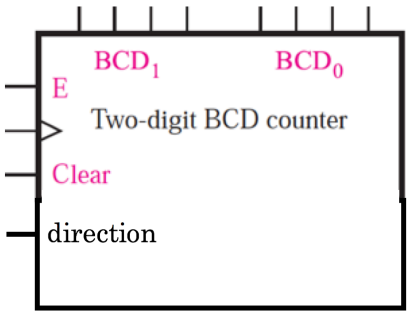

```
entity gendiv is
generic (N: integer :=8);
port (clock, En, L, w: in std_logic; Q: buffer std_logic_vector(N-1
downto 0); R: in std_logic_vector(N-1 downto 0) );
end gendiv;

architecture arch of gendiv is
begin
process(clock,L)
begin
if(L='1') then
Q <= R;
elsif(clock'event and clock='1') then
if En='1' then
rightside: for i IN 0 to (N/2-2) loop
Q(i) <= Q(i+1);
end loop;
Q(N/2-1) <= w;
leftside: for i IN (N/2+1) to N-1 loop
Q(i) <= Q(i-1);
end loop;
Q(N/2) <= w;
end if;
end if;
end process;
end arch;
```

Question 4- (30 pts)

You are required to design a room occupancy tracker system. The room has two doors: entrance and exit with one sensor installed on each door. The sensor sends a HIGH pulse whenever a person is detected. Two seven segment displays are used to show the count of people inside the room from 00 to 99.



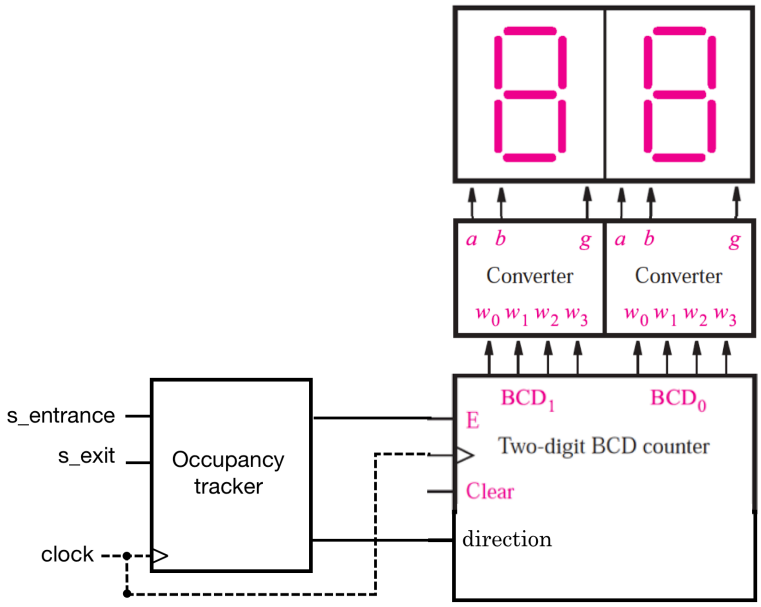
Part 4.1- Design a BCD counter that can count up or down.



The counter counts from 00 to 99.
When E=1, the BCD counter is active and either counts up if "direction" = 1 or counts down if "direction" =0 on each clock edge.
When clear=1, the output is cleared to 00.
When count reaches 99, and we try to count up, it stays 99
When count reaches 00 and we try to count down, it stays 00

```
Entity BCDcount IS
PORT (clock, clear, E, direction: IN STD_LOGIC;
BCD1, BCD0: BUFFER STD_LOGIC_VECTOR(3 downto 0));
END BCDcount;

Architecture behave of BCDcount is
BEGIN
Process (clock)
IF clock'event and clock='1' THEN

IF clear='1' THEN

BCD1 <= "0000";

BCD0<="0000";

ELSE IF E='1' AND direction ="1" then

      If BCD0 = "1001" then

            IF BCD1 != "1001" then

                  BCD0 <="0000"

                  BCD1 <= BCD1+1;

            End if;

      Else BCD0 <= BCD0+1;

      End if;

Elsif E='1' AND direction ="0" then

If BCD0 = "0000" then

 If BCD1!= "0000" then

BCD0 <= "1001";

BCD1 <= BCD1-1;

End if;

Else BCD0 <= BCD0-1;

End if;

END process;
END behave;
```

Part 4.2- The overall circuit is depicted here.



Complete the code of the occupancy tracker component where s_entrance and s_exit represent the sensors on the doors.

```
Entity occupancy_tracker IS
PORT (clock, s_entrance, s_exit: IN STD_LOGIC;
to_enable, to_direction: OUT STD_LOGIC);
END occupancy_tracker;
Architecture behave of occupancy_tracker IS
BEGIN
Process (clock)

IF clock'event and clock='1' then

If s_entrance='1' then

to_enable <='1';

to_direction <='1';

elsif s_exit='1' then

to_enable<='1';

to_direction<='0'

end if;    END behave;
```

Part 4.3- Design the whole system. Assume you already defined the components from parts a and b. Also assume you have a component called `7_seg_converter (bcdin: std_logic_vector(3 downto 0), ledsout: std_logic_vector(1 to 7));` Assume all components are included in the library.
The inputs to the entire system are the sensors, the clock, the clear of the bcd counter, while the outputs are the outputs of the two converters.

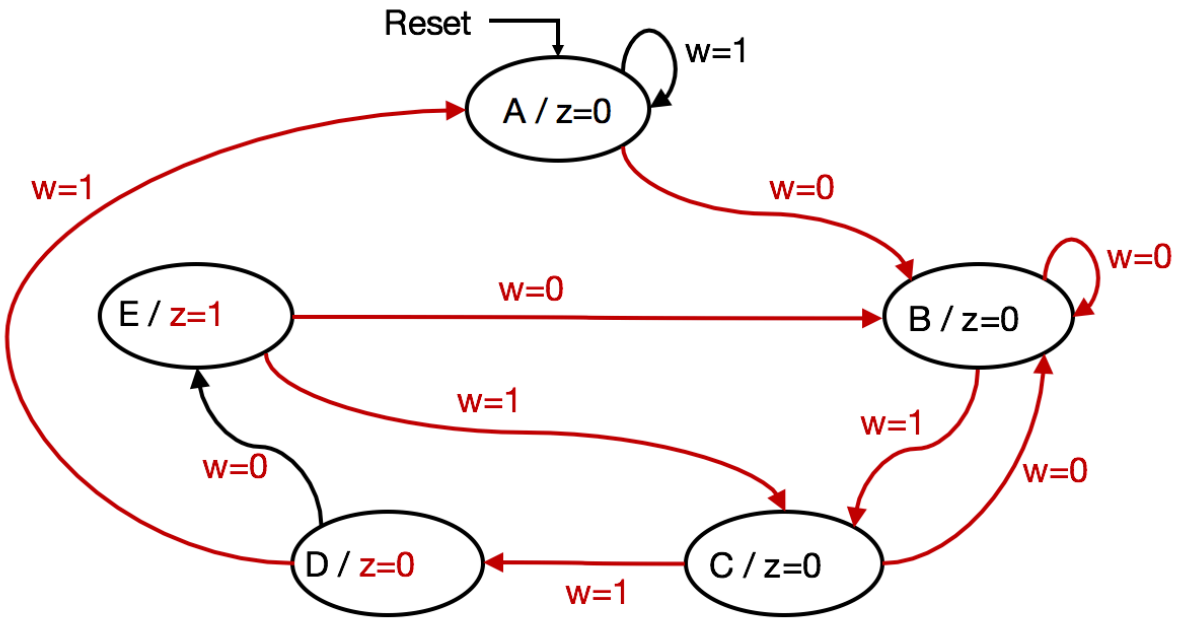
```
ENTITY peoplecount IS
PORT (   s_entrance, s_exit, clock, clear: in std_logic; seg1, seg0: out std_logic_vector(6 downto 0)
);
END peoplecount;

ARCHITECTURE arch OF peoplecount IS


Signal sig_enable, sig_direction: std_logic;

Signal sig_bcd1, sig_bcd0: std_logic_vector(3 downto 0)



Begin
tracker: occupancy_tracker port map(clock, s_entrance, s_exit, sig_enable, sig_direction);

counter: BCDcount port map(clock, clear, sig_enable, sig_direction, sig_bcd1, sig_bcd0);

Conv1: 7_seg_converter port map(sig_bcd1, seg1);

Conv2: 7_seg_converter port map(sig_bcd0, seg0);









END arch
```

Question 5 – (20 pts)

You are required to design a sequence detector that detects the occurrence of 0110 in a stream of inputs w. When the sequence is detected, the output Z is set to 1.
Example:

Stream W	0	0	**0**	**1**	**1**	**0**	1	0	0	**0**	**1**	**1**	**0**	**1**	**1**	**0**	1	1	1	1
Output Z	0	0	0	0	0	0	**1**	0	0	0	0	0	0	**1**	0	0	**1**	0	0	0

a) Design the Moore FSM state diagram by completing this diagram.



b) Complete this VHDL code to implement the FSM in part (a). The clock is negative edge triggered and the reset is asynchronous and active high.

```
ENTITY sequenceDetector IS

PORT ( Clock, Resetn, w : IN STD LOGIC ;

 z : OUT STD LOGIC ) ;

 END sequenceDetector ;


 ARCHITECTURE Behavior OF sequenceDetector IS

 TYPE myState IS (  A,B,C,D,E) ;

 SIGNAL y: MYSTATE;

 BEGIN

 PROCESS (RESETN,CLOCK)

 BEGIN

 If (  resetn='1'      ) then

 y <= A   ;

 elsif (clock'event and clock='0') then

CASE y IS

 WHEN A =>

If w='0' then y <= B; end if;  (the else here is redundant but not
false)


WHEN B =>

If w='1' then y<= C; end if; (else is redundant)
```

```
 WHEN C =>
If w='0' then y<= B; else y<= D; end if;

WHEN D =>
If w='0' then y<= E; else y<= A; end if;

 WHEN E =>
If w='0' then y<= B; else y<= C; end if;


 END CASE ;

END PROCESS;
Z <= '1' when y=E else '0';


 END Behavior
```

Scratch Sheet

