UMBB/ I.G.E.E/ EE222 Time Allowed: 1h 30' **May 31, 2023**

Final Exam

Family Name: .....

Given Name: ..... **SOLUTION SOLUTION SOLUTION**

Group: _____ **You may write on both sides of the sheets**

Exercise # 1 2 Points Comprehension

i- The basic encoder exhibits 2 drawbacks that are solved by the Priority Encoder. What are these 2 drawbacks?

1. Code 00…0 (is it the code word of input 0 or no input?)
2. When more than one input active at a time

ii- What is the main purpose of an LFSR?

To generate pseudo-random sequences

iii- Give the state equations of:

1- An n-bit SISO: $Q_s(t) = I_s(t - n)$

2- An n-bit PIPO: $\underline{Q^+} = \underline{I}$

iv- Why a DEMUX is also called a DEC? Why the reciprocal is not always true?

DEC: $Yi = m_i(\underline{X})$

DEMUX: $Yi = Im_i(\underline{X})$ if I = 1, then DEMUX ≡ DEC

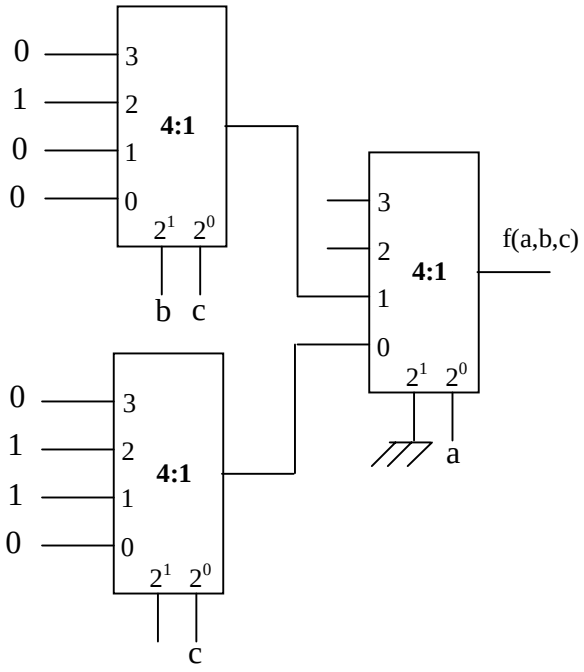
If a DEC is not provided with an En input, then it cannot be used as DEMUX

Exercise # 2 3 Points

We **only** have at our disposal **4:1 MUXs** that are not provided with a strobe input. We wish to construct an 8:1 MUX. Design your circuit and annotate (label) **all** input **and** output signals.

Show how to implement f(a, b, c) = ΠM(0, 3, 4, 5, 7)

$f(a, b, c) = \Pi M(0, 3, 4, 5, 7) = \sum m(1, 2, 6)$



Exercise # 3 3-Points Analysis

Consider the following digital system.

1. Briefly describe the operation of this system.
2. At what rate does the Mod-10 count when S_1S_0 = "01"?

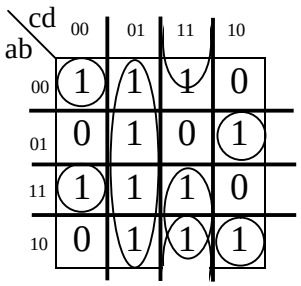
1. The system consists of a 20-bit binary counter used as a 1MHz clock divider, a 4:1 multiplexer used to select one of the four possible clock signals to drive two counters; a positive-edge triggered modulo-10 counter and a negative-edge triggered modulo-16 counter. The selection of the clock to trigger the counters is done via two binary signals S1 and S0.
2. With S_1S_0 = "01", the multiplexer selects bit 15 of the clock divider counter. The frequency of this output signal is 1MHz / 2^{16} = 1,000,000/65536 = 15.2587 Hz. Therefore, the Mod-10 (as well as the Mod-16) counter counts at a rate of 15.2587 Hz or changes state every 0.065536 sec or 65.536 msec.

Exercise # 4 3 Points

Using 4:1 MUXs and a minimum number of gates, implement the following 4-variable Boolean function

F(a, b, c, d) = ∑ m (0, 1, 3, 5, 6, 9, 10, 11, 12, 13, 15)

You should use variables {**c, d**} as controlling variables and variables {**a, b**} as front inputs (residues variables)

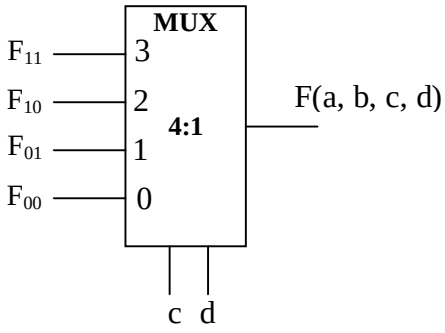


$F_{00}(a,b) = \overline{a \oplus b}$

$F_{01}(a,b) = 1$

$F_{10}(a,b) = a \oplus b$

$F_{11}(a,b) = a + \overline{b}$



Exercise # 5 4 Poınts

A **Mod-8** up counter, an **octal to 7-segment** code converter and a **3:8** decoder are interconnected as shown.
Write the **structural** VHDL code to synthesize this system.
NB: do not write the VHDL codes that implement the components. (Just the top-file)

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity Prob_5_Structural is
port(Clk, Rst, En: in std_logic;
Y7, Y6, Y5, Y4, Y3, Y2, Y1, Y0 : out std_logic;
a, b, c, d, e, f, g: out std_logic);
end Prob_5_Structural;
architecture Structural of Prob_5_Structural is
COMPONENT Counter
Port(Clk, Rst: in std_logic;
Q: out unsigned(2 downto 0));
end COMPONENT;
COMPONENT DEC8
Port(
S: in std_logic;
addr: in unsigned(2 downto 0);
W7, W6, W5, W4, W3, W2, W1, W0 : out std_logic);
end COMPONENT;
COMPONENT Converter
Port(addr: in unsigned(2 downto 0);
M6, M5, M4, M3, M2, M1, M0: out std_logic);
End COMPONENT;
signal state: unsigned(2 downto 0);
begin
Comp1: Counter PORT MAP (Clk, Rst, state);
Comp2: DEC8    PORT MAP (En, state, Y7,
                         Y6, Y5, Y4, Y3, Y2, Y1, Y0);
Comp3: Converter PORT MAP (state,a, b, c, d, e, f, g);
End Structural;
```

Exercise # 6 5-Points

The block diagram of a positive edge-triggered **Mod-9600** synchronous up-down counter is shown in **Fig. 1**. The counter is provided with an asynchronous **Reset** input and a synchronous **GO** input (an input that enables or disables counting).
Write a behavioral VHDL where **Count_Out** should be declared as an **unsigned** data type.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity Counter_9600_UD is
port(Clk, Reset, GO, UD: in std_logic;
Count_Out : out unsigned(13 downto 0));
end Counter_9600_UD;
architecture behavioral of Counter_9600_UD is
signal state: unsigned(13 downto 0);
begin

process(Clk, Reset)
begin
If Reset = '0' then state <= (others => '0');
Elsif (clk'event and clk = '1') then
if GO = '0' then state <= state;
else
CASE UD is
 When '1' => if state = 9599 then state <= (others => '0');
else state <= state + 1; end if;
When '0' => if state = 0 then state <=
"10010101111111";
Else state <= state - 1; end if;
When others => null;
End CASE;
end if;
end if;
end process;
Count_Out <= state;
end behavioral;
```