Version DP/2

EGT2
ENGINEERING TRIPOS PART IIA

Friday 27 April 2018 9.30 to 11.10

Module 3B2

INTEGRATED DIGITAL ELECTRONICS

*Answer not more than **three** questions.*

All questions carry the same number of marks.

*The **approximate** percentage of marks allocated to each part of a question is indicated in the right margin.*

*Write your candidate number **not** your name on the cover sheet.*

STATIONERY REQUIREMENTS
Single-sided script paper

SPECIAL REQUIREMENTS TO BE SUPPLIED FOR THIS EXAM
CUED approved calculator allowed
Engineering Data Book

10 minutes reading time is allowed for this paper at the start of the exam.

You may not start to read the questions printed on the subsequent pages of this question paper until instructed to do so.

1 (a) Design a non-inverting Schmitt trigger buffer using only *n*-type and *p*-type Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs), and draw the schematic of your design. You do not have to write the exact size of transistors, but instead you may label individual transistors as wide (W), medium (M), or narrow (N). It is understood that *p*-type MOSFETs will have twice the width of *n*-type MOSFETs with the same label. Make sure that there is high resistance at the input of your gate, so that a change of your design would have little impact on the previous gate, i.e. the gate driving your design. Explain your answer. [40%]

(b) What function $f(a,b,c)$ does the circuit in Fig.1 implement? Explain your answer. (Here, CLK is a clock signal varying between high and low logic values.) [30%]

(c) Implement the function $f(a,b,c)=(a\bar{b}+\bar{a}b)c$ using the minimal number of transistors. You can use letters (a, $\bar{a}$, …) as direct inputs to your design. [30%]

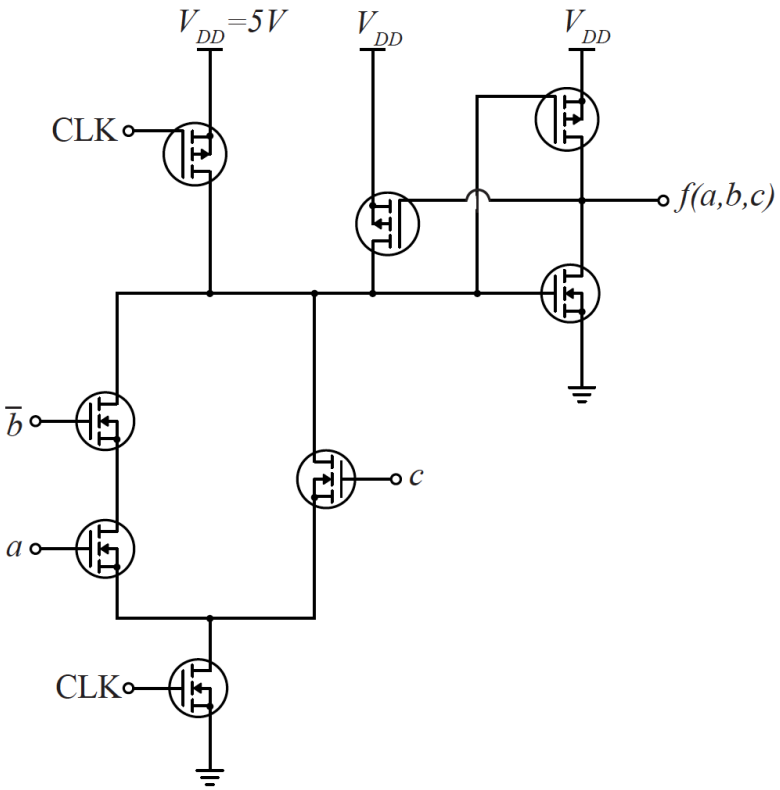

Fig. 1

2 (a) Draw and label the schematics for a minimum-sized CMOS inverter assuming the transistor parameters of Table 1. Find the channel dimensions (i.e. W/L) of the transistors in the CMOS inverter to have a switching point $V_{inv} = 0.5V_{DD}$. [30%]

(b) The inverter designed in (a) drives a total capacitance of 45 fF at 4.1 GHz. How much power does it consume? [20%]

(c) The inverter designed in (a) drives an identical copy of itself. Using the capacitance values from Table 2, find the total load capacitance seen by the driver inverter $C_T = C_P + C_{in}$, where C_P is the unloaded (intrinsic) output capacitance of the driver inverter and C_{in} is the input capacitance of the load inverter. [20%]

(d) Find the t_{pLH} (i.e. the propagation delay from low to high signal levels) for the inverter designed in (a) driving a total load capacitance C_T from (c). [30%]

You may use the following equations for the drain current I_D flowing in an *n*-type MOSFET: $I_D = \frac{k}{2}[2(V_{GS} - V_T)V_{DS} - V_{DS}^2]$ for $V_{DS} < (V_{GS} - V_T)$, and $I_D = \frac{k}{2}(V_{GS} - V_T)^2$ for $V_{DS} \geq (V_{GS} - V_T)$, where the symbols have their usual significance and $k = k'(W/L)$ is the transconductance parameter for a MOSFET.

Table 1: Transistor parameters

Parameter	**NMOS**	**PMOS**
V_T	0.8V	-0.8V
k'	300μA/V^2	100μA/V^2
W_{min}	1.0μm	1.0μm
L_{min}	1.0μm	1.0μm
λ	0.0V^{-1}	0.0V^{-1}
V_{DD}	5V	5V

Table 2: PMOS and NMOS capacitances per unit width

Capacitance/Width(fF/μm)	**PMOS**	**NMOS**
C_{gs}	1.5	1.4
C_{gd}	1.5	1.4
C_{db}	1.0	0.9
C_{sb}	1.0	0.9

3 (a) Explain the differences between Complex Programmable Logic Devices (CPLDs) and Field-Programmable Gate Arrays (FPGAs). [10%]

(b) Give two examples of logic functions with five inputs, x_0, x_1, … , x_4 that can be realized using 2 four-input Look-up Tables (4-LUTs). [20%]

(c) Figure 2 shows the circuit for a 4-LUT.

(i) Find the storage cell contents needed to implement the function:

$$f = \bar{x}_0\bar{x}_1 + x_0x_1\bar{x}_2\bar{x}_3 + x_0x_1x_2x_3.$$ [20%]

(ii) Give a circuit that implements a 4-to-1 multiplexer with select inputs s_1 and s_0 and data inputs x_3, x_2, x_1 and x_0 using only 4-LUT circuits. Show the storage cell contents inside each LUT. [30%]

(d) What is the minimum number of 4-LUTs needed to construct a 4-to-1 multiplexer. [20%]

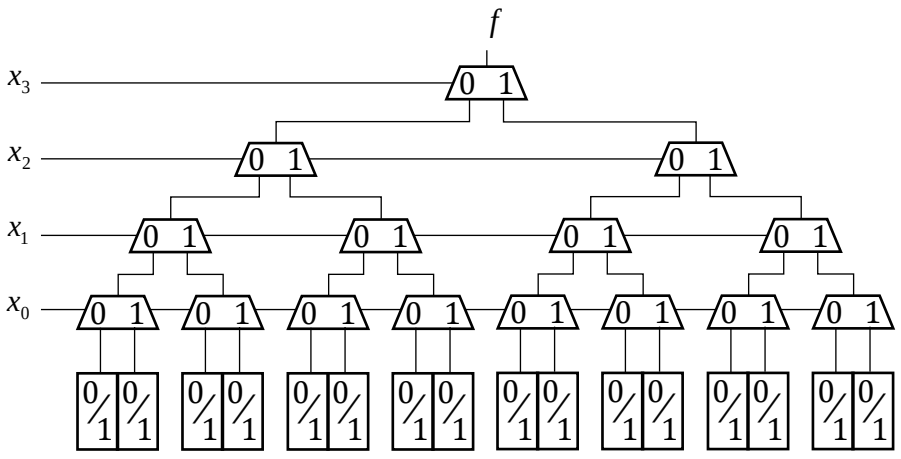

Fig. 2

4 (a) Compare the EPROM, EEPROM and SRAM programming technologies. [20%]

(b) A Finite State Machine (FSM) is described by the VHDL code in Fig. 3.

(i) Is this a Mealy or a Moore configuration? Explain your answer. [20%]

(ii) Draw the state diagram and the state table for the FSM in Fig. 3. Describe its functionality. [40%]

(iii) Draw a circuit implementation using D bistables and logic gates. [20%]

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;

ENTITY FSM IS
        PORT ( Clock, Reset, x      : IN STD_LOGIC ;
                             z      : OUT STD_LOGIC);
END FSM ;

ARCHITECTURE Behavior OF FSM IS
        TYPE State_type IS (A, B, C) ;
        SIGNAL y : State_type ;
BEGIN
        PROCESS ( Reset, Clock )
        BEGIN
                IF Reset = '0' THEN
                        y <= A ;
                ELSIF (Clock'EVENT AND Clock = '1') THEN
                        CASE y IS
                                WHEN A =>
                                        IF x = '0' THEN
                                                y <= B ;
                                        ELSE
                                                y <= C;
                                        END IF ;
                                WHEN B =>
                                        IF x = '0' THEN
                                                y <= B ;
                                        ELSE
                                                y <= C;
                                        END IF ;
                                WHEN C =>
```

```
                                IF x = '0' THEN
                                    y <= B ;
                                ELSE
                                    y <= C;
                                END IF ;
                        END CASE ;
                END IF ;
        END PROCESS ;

        PROCESS ( y, x )
        BEGIN
                CASE y IS
                        WHEN A =>
                                z <= '0' ;
                        WHEN B =>
                                z <= NOT x ;
                        WHEN C =>
                                z <= x ;
                END CASE ;
        END PROCESS ;
END Behavior ;
```

Fig. 3

END OF PAPER