Version OBA/2

EGT1
ENGINEERING TRIPOS PART IIA

Tuesday 03 May 2022 9:30 to 11:10

Module 3B2

ELECTRICAL ENGINEERING

*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.

You may not remove any stationery from the Examination Room.

1 (a) Explain how field-programmable gate arrays (FPGA) implement logic functions, such as gates, in a programmable way. How are the inputs and outputs of a logic function realised in this concept? In what way does this differ from complex programmable logic devices (CPLD)? Which circuit elements can implement the programmable part representing logic functions in an FPGA? [30%]

Solution:

A memory (also called look-up table in the FPGA world) represents logic functions. The content of the memory cell represents the output of the logic function (and is programmable). The address of the memory represents the input. CPLDs, an alternative approach for programmable logic, in contrast have mostly standard logic functions inside and mostly provides programmable interconnection resources in array form (which FPGAs often have on top). The memory in FPGAs can be implemented with typical memory circuits, such as all kinds of programmable ROMs (EPROM) or static memory (SRAM; Flip-Flops, etc.).

(b) Which two ways of describing the behaviour of a logic block (entity) exist in VHDL? What is the key difference between both? [20%]

Solution:

Architecture: Architectures describe the logic functions and interconnections as a declaration. The individual lines of the code are interpreted during the synthesis phase to form an implementation that follows that declaration. Importantly, all lines are practically running at the same time so that the order does in most cases not matter.

Process: A process mimics higher-level programming languages, such as C/C++. The individual lines are practically executed sequentially, one by one.

(c) Which function is represented by the VHDL code in Fig. 1? Simplify the function and write it as a product of sums or a sum of products. [30%]

Solution:

$$
\begin{aligned}
f(x) &= \overline{(x_4x_3x_2x_1 + x_4\overline{x_3}\,\overline{x_2}x_1 + x_1\overline{x_4} + x_1x_3 + \cancel{x_1x_2\overline{x_1}x_4} + x_1\overline{x_2}x_3 + x_1x_4x_2\overline{x_3})} \\
&= \overline{(x_1(x_4x_3x_2 + x_4\overline{x_3}\,\overline{x_2} + \overline{x_4} + x_3 + x_4x_2\overline{x_3})} \\
&= \overline{(x_1(x_4x_3x_2 + x_4\overline{x_3}\,\overline{x_2} + \overline{x_4} + x_3 + x_4x_2\overline{x_3})} \text{ with } x_3 = x_4x_3 + \overline{x_4}x_3 \\
&= \overline{\left(x_1\left(x_4(x_3x_2 + \overline{x_3}\,\overline{x_2} + x_2\overline{x_3} + x_3) + \overline{x_4}(1 + \cancel{x_3})\right)\right)} \\
&= \overline{\left(x_1\left(x_4(\cancel{x_3x_2} + \overline{x_3}\,\overline{x_2} + x_2\overline{x_3} + x_3) + \overline{x_4}\right)\right)} \\
&= \overline{\left(x_1\left(x_4\left(\overline{x_3}(\overline{x_2} + x_2) + x_3\right) + \overline{x_4}\right)\right)} \\
&= \overline{\left(x_1\left(x_4\left(\overline{x_3}(1) + x_3\right) + \overline{x_4}\right)\right)} \\
&= \overline{(x_1(x_4(1) + \overline{x_4}))} \\
&= \overline{(x_1(x_4 + \overline{x_4}))} \\
&= \overline{x_1}
\end{aligned}
$$

(d) Derive a function in standard binary coded decimal (BCD) that outputs TRUE (1) for every prime number up to 15 as sum of products (0 and 1 are not considered prime numbers). In the application, the previous and subsequent processing steps do not evaluate multiples of 5 so that these can return any result. Simplify this function to use as few terms as possible. [20%]

Solution:

Prime numbers:

2 (0010)
3 (0011)
5 (0101)
7 (0111)
11 (1011)
13 (1101)

Do-not-cares:

5 (0101)
10 (1010)
15 (1111)

$\Longrightarrow f_{prime}$(x : 4 downto 1) = (NOT x(1) AND x(2) AND NOT x(3) AND NOT x(4)) OR

(x(1) AND x(2) AND NOT x(3) AND NOT x(4)) OR (x(1) AND NOT x(2) AND x(3) AND NOT x(4)) OR (x(1) AND x(2) AND x(3) AND NOT x(4)) OR (x(1) AND x(2) AND NOT x(3) AND x(4)) OR (x(1) AND NOT x(2) AND x(3) AND x(4))

$$f_{prime} = \overline{x_4}\overline{x_3}x_2\overline{x_1} + \overline{x_4}\overline{x_3}x_2x_1 + \overline{x_4}x_3\overline{x_2}x_1 + \overline{x_4}x_3x_2x_1 + x_4\overline{x_3}x_2x_1 + x_4x_3\overline{x_2}x_1 \quad (1)$$

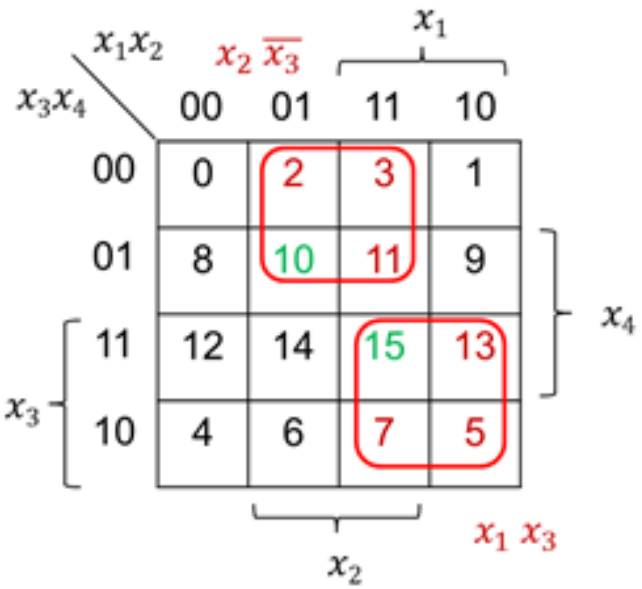
$$f_{dnc1} = x_4\overline{x_3}x_2\overline{x_1} \quad (2)$$

$$f_{dnc2} = \overline{x_4}x_3\overline{x_2}x_1 \quad (3)$$

$$f_{dnc3} = x_4x_3x_2x_1 \quad (4)$$

Each can be added to f in inverted and non-inverted if helpful for simplification.

Simplification (Karnaugh map or through de Morgan)



Hence, $f = x_2\overline{x_3} + x_1x_3$

 (cont.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY Q1 IS
      PORT ( x :  IN STD_LOGIC_VECTOR(4 DOWNTO 1);
             f :  OUT STD_LOGIC );
END Q1;

ARCHITECTURE Behavior OF Q1 IS
      SIGNAL mysig1 :  STD_LOGIC_VECTOR(4 DOWNTO 1) := "0000";
      SIGNAL mysig2 :  STD_LOGIC_VECTOR(4 DOWNTO 1) := "0000";
      CONSTANT one :  STD_LOGIC_VECTOR(4 DOWNTO 1) := "1111";
BEGIN
     mysig2 <= (x(1) AND NOT x(2) AND x(3)) OR (x(1) AND x(4)
AND x(2) AND NOT x(3));
      WITH x SELECT
           f <= '0' WHEN one,
                '0' WHEN mysig1,
                '0' WHEN "1001",
                '0' WHEN mysig2,
                '1' WHEN OTHERS;
     mysig1 <= (x(1) AND NOT x(4)) OR (x(1) AND x(3)) OR (x(1)
AND x(2) AND NOT x(1) AND x(4));
END Behavior;
```

Fig. 1

Assessor's Comments: This was a popular question with 54 attempts out of a total of 75 students. The mean was 59.7 out of 100 with a standard deviation of 21.4. The part (a) was about the main principles and differences of FPGA and CPLD. Most students clearly explained the fundamentals for FPGA, CPLD and their working principles. Part (b) focused on different ways of describing the behaviour of a logic block. Vast majority of the students answered correctly as well. Part (c) was about assessing the basic understanding of a given VHDL code, which included a logic function. The students were mostly very successful reading the VHDL code, however, most did not entirely completed the simplification of the function. A large group of students have minimised the function counting prime numbers in part (d). Overall, the students showed that they grasped all basic concepts regarding the FPGAs, VHDL, deriving logic functions and their minimisation.

2 Consider the circuit shown in Fig. 2.

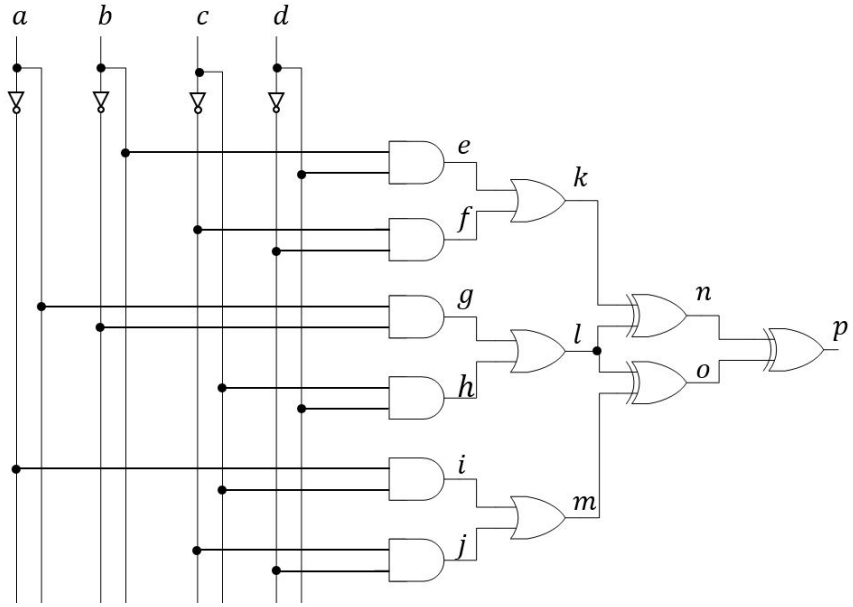


Fig. 2

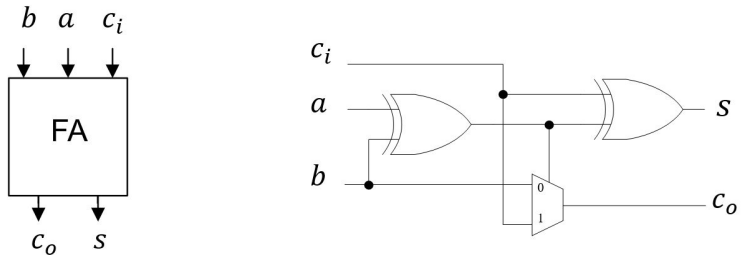


Fig. 3

(a) Find the simplest expression for the node p, where $p = f(k, l, m)$. [20%]

Solution:

$n = k \oplus l$, $o = l \oplus m$ and $p = n \oplus o$. Thus, $p = (k \oplus l) \oplus (l \oplus m)$ and by simplifying this, we get $p = k \oplus m$.

(b) To implement the circuit in Fig. 2 with minimum number of lookup table(s), how many inputs do each of the lookup table(s) require? Explain your answer. [20%]

Solution:

Any function with 4 inputs can be implemented with a 4-input lookup table (LUT). For the LUT equivalent to circuit in Fig. 2, the inputs of LUT will be a, b, c, d and the output will be p.

(c) If only 3-input lookup tables (3-LUTs) are available, what is the minimum number of 3-LUTs required to implement this circuit? Indicate the input and output of each LUT according to the node names in Fig. 2. Mark unused input(s) of a lookup table with X. Do not attempt to simplify the gate-level circuit shown in Fig. 2. [40%]

Solution:

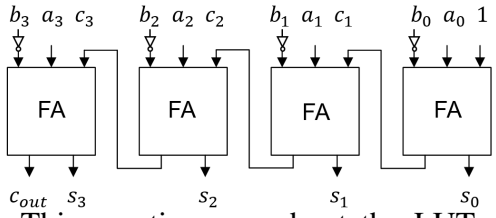
5 LUTs are required without simplifying the gate-level circuit shown in Fig. 2 as follows.

LUT#	INPUT1	INPUT2	INPUT3	OUTPUT
1	b	c	d	k
2	a	b	x	g
3	g	c	d	l
4	a	c	d	m
5	k	l	m	p

The answers which simplify the circuit based on results of part (a) would require 3 LUTs (LUT#2 and LUT#3 would not be needed and inputs of LUT#5 would be k, m, and x). This answer also gets full mark provided that input/output relations of the LUTs are correct.

(d) Using the Full-adder (FA) shown in Fig. 3, implement a logic circuit to subtract two 4-bit numbers. [20%]

Solution:



Assessor's Comments: This question was about the LUTs and combinational logic circuits, which are among the fundamental building blocks of FPGA operation. Part (a) asked for the simplest expression for the node p in the LUT circuit given in Fig 2. It was pleasing that a large number of candidates could find it correctly. The main concepts of LUTs were examined in part (b), which were also mostly understood and well answered. In part (c), the 3-LUTs implementation of the circuit was asked, many students carried the results of part (a) in terms of the simplified version of the function, nevertheless they also received full mark provided that they have correctly indicated inputs and outputs of each LUT. Part (d) focused on implementing a 4-bit binary subtractor using the given full-adder in Fig 3. Again, a large group of candidates performed this implementation correctly. Overall, the question was attempted by less number of candidates, i.e., 49 out of 75, however, pleasingly answered very well with the mean of 67.2 out of 100.

3 (a) Give two disadvantages of the use of resistive load compared to transistor (active) load in MOS logic circuit implementations. [10%]

Solution:

Resistive loads are more expensive, they use the surface area inefficiently, the parasitic capacitances/inductances in the overall processor increases due to the use of long conductors in close vicinity.

(b) The propagation delay and power dissipation values of different technologies are given in Fig. 4.

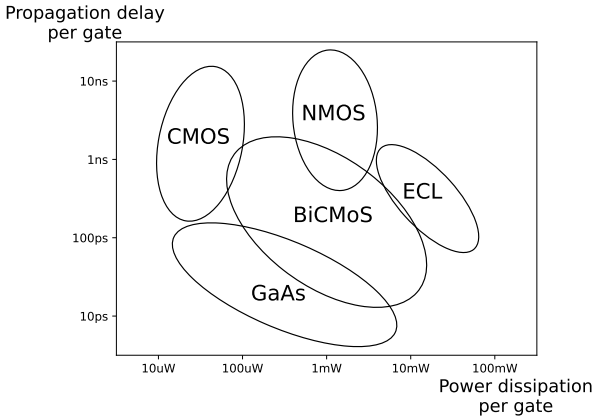


Fig. 4

(i) Suggest a technology for an integrated digital electronics device with a long-lasting battery, operating around 1-2 GHz. Give your reasoning. [10%]

Solution:

1-2 GHz corresponds to 0.5-1 ns propagation delay. All technologies shown here satisfies the frequency requirement. However, since we want the device to have a long-lasting battery, the technologies with lower power dissipation are chosen. Among CMOS and GaAs, the cheaper technology needs to be selected.

(ii) Compare NMOS and CMOS from the perspectives of power consumption, manufacturing cost and area efficiency. [20%]

Solution:

NMOS has a higher power consumption than CMOS due to the static leakage of NMOS. NMOS is cheaper to manufacture because for the PMOS part, we first need to build an n-well into the p-substrate. (Or simply saying that the procedure involves more steps is enough). NMOS technology is more area efficient than CMOS technology, because PMOS part of the CMOS occupies more area due to the low hole mobility.

(cont.

(c) A CMOS 2-input NAND gate is given in Fig. 5. Take $k'_N = 2k'_P$. Use

$$k' = \frac{\mu\varepsilon_{ox}}{t_{ox}},$$

$$k = k'\left(\frac{W}{L}\right),$$

where k is the transconductance parameter, μ is the carrier mobility, ε_{ox} is the permittivity of the oxide layer, t_{ox} is the oxide thickness and $\left(\frac{W}{L}\right)$ is the width/length ratio of the device. For NMOS transistors, take $\left(\frac{W}{L}\right) = 5$.

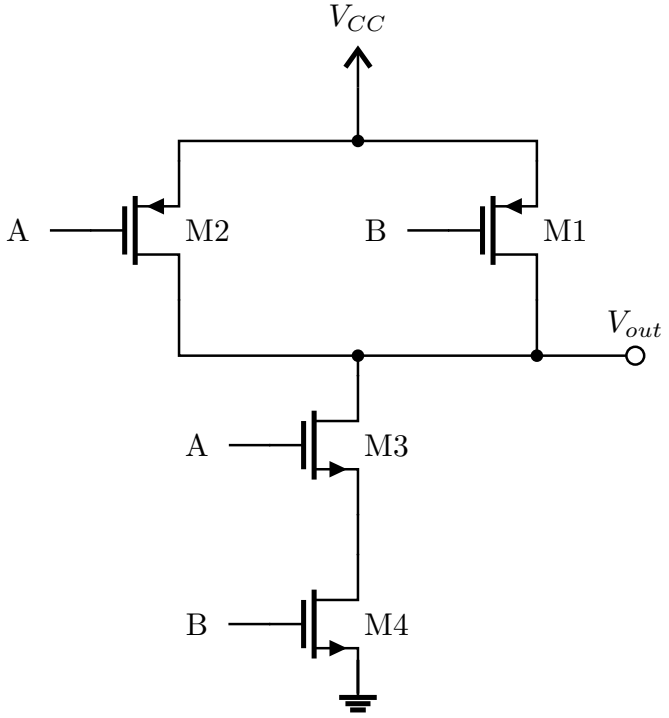


Fig. 5

(i) Find the required $\left(\frac{W}{L}\right)$ ratio for PMOS transistors of the NAND gate in Fig. 5 such that the rise time, t_r^{NAND}, for the output when the input is AB=00, is equal to the fall time, t_f^{NAND}. [15%]

Solution:

We know that t_r^{NAND} and t_f^{NAND} are proportional to the time constants $\tau_r = R_r C_L$ and $\tau_f = R_f C_L$, respectively. Thus, if $t_f^{NAND} = t_r^{NAND}$ when AB=00, $R_f = R_r$ when AB=00. Therefore, $R_f \sim \frac{1}{k_N} + \frac{1}{k_N} = \frac{2}{k_N} = \frac{2}{5k'_N}$. For AB=00, both PMOS transistors are ON, so that R_r is composed of two parallel resistors, each proportional to $1/k_P$, i.e., $R_r \sim \frac{1}{2k_P} = \frac{2}{5k'_N}$

Substituting $k'_N = 2k'_P$, we have $\frac{1}{2k_P} = \frac{1}{5k'_P}$, which gives $k_P = 2.5k'_P$, and $(W/L) = 2.5$.

(ii) Due to a fabrication error, $\left(\frac{W}{L}\right)$ for M2 is 20% less than what it should have been when matched properly. How do you think this will impact the overall operation of the gate? [15%]

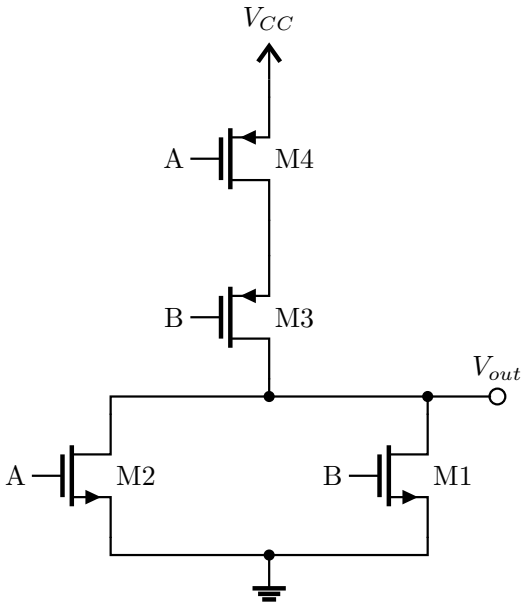
Solution:

I_{DS} is proportional with $k'_P\left(\frac{W}{L}\right)$. Concurrently, if W/L is smaller than what it should be, I_{DS} is smaller. Since propagation delay is a measure of the charging of the load capacitor, it is inversely proportional with $k'_P\left(\frac{W}{L}\right)$. As a result, whenever M2 is ON, the propagation delay is higher than it would be for a non-erroneous M2. Thus, the maximum achievable operational frequency of this gate is lower than it would be for a correctly manufactured gate. To compansate for the fabrication errors, the overall operational frequency of the device should be reduced accordingly.

(iii) Draw a transistor-level implementation of a CMOS 2-input NOR gate. [15%]

Solution:

(cont.



(iv) Find the required $\left(\frac{W}{L}\right)$ ratio for all transistors of the CMOS 2-input NOR gate, if the fall time of NOR gate, t_f^{NOR}, when NOR input is AB=11, is equal to the fall time of NAND gate, t_f^{NAND}, and the rise time of NOR gate, t_r^{NOR}, is equal to the rise time of NAND gate, t_r^{NAND}, when NAND input is AB=00. [10%]

Solution:

In part c (i), we state that $t_r^{NAND} \sim \tau_r = R_r C_L$ and $t_f^{NAND} \sim \tau_f = R_f C_L$.

For AB=11, NOR pull-down happens with two parallel NMOS transistors, so that $t_f^{NOR} \sim \frac{1}{2k_N}$. Equating t_f^{NOR} with t_f^{NAND}, $\frac{1}{2k_N} = \frac{2}{5k'_N} \Longrightarrow k_N = 1.25k'_N$, which results in (W/L)=1.25 for NMOS transistors.

For AB=00, NAND pull-up happens with two parallel PMOS transistors, and using our result in part c (i), we have $t_r^{NAND} \sim \frac{1}{5k'_P}$. Since NOR pull-up happens through two series transistors $t_r^{NOR} \sim \frac{2}{k_P}$. Equating these two, $\frac{1}{5k'_P} = \frac{2}{k_P} \Longrightarrow k_P = 10k'_P$, making $(W/L) = 10$ for PMOS transistors.

(v) Both NAND and NOR gates are universal, i.e., any logic function can be expressed solely by NAND or NOR gates. Which one, i.e., NAND or NOR, is a better choice in terms of area efficiency? Give your reasoning. [5%]

Solution:

NAND gates are smaller in size due to the higher mobility of electrons, i.e., $k'_N > k'_P$, which is assumed to be $k'_N = 2k'_P$ in this question. Hence, using NMOS transistors in series is better in terms of area efficiency.

Assessor's Comments: The question was about CMOS fundamentals and how they affect certain performance metrics such as power, cost, frequency of digital circuits. The question was also popular as 55 candidates out of 75 attempted. The average was 59.6 out of 100 with a standard deviation of 18. Part (a) asked the main disadvantages of the use of resistive load compared to transistor load in MOS logic implementations. While overall majority of the candidates answered correctly, surprisingly a large number of candidates did not consider the surface area inefficiency of the resistive loads. Part (b)(i) was about finding suitable transistor technology for a given operating frequency and power constraints using Fig. 4. Most students successfully identified potential solutions, while few recommended to use ECL and BiCMoS despite more power-efficient alternatives are available. Majority of the students attempted the part (b)(ii) were very successful in comparing NMOS and CMOS from the perspectives of power, manufacturing cost and area efficiency. Part (c)(i),(ii),(iv) were about the relation between device dimensions, transistor currents, and hence, switching performance under different settings. Some of the candidates performed very well in these questions. Part (c)(iii) was about a transistor-level implementation of a CMOS 2-input NOR gate. The students were able to provide this implementation without major difficulty. Part (c)(v) focused on the comparison of NAND and NOR gates in terms of the area efficiency. While some students directly concluded that they would perform the same as they have the same number of transistors, however, most students acknowledged the differences in the circuit structures of these gates yielding NAND as more favorable in terms of the area efficiency.

4 (a) What are the advantages and disadvantages of Schottky barrier diodes for logic gates compared to PN diodes? [10%]

Solution:

SBD usually has a faster switching rate and better noise margins, but it performs very bad for large scale circuitry because it has a leakage current three orders larger than PN.

(b) How is the switching frequency of a bipolar transistor impacted by charge storage? How can this be mitigated? [20%]

Solution:

Fast switching in bipolar transistors is prevented by the removal of minority carrier charges in the base region during saturation. There are three options to this effect: (i) Transistor-transistor logic which uses high conductance paths to drive the base, (ii) Emitter-Coupled logic which avoids saturation by controlling base emitter voltage, (iii) Schottky-Transistor logic which uses Schottky transistors with modified base-collector structure, preventing saturation.

(c) Consider the circuit shown in Fig. 6. Two inputs, A and B are provided for an output V_{out}.

(i) Which technology is used for this circuit? [10%]

Solution:

BiCMoS

(ii) Which logic function is implemented by this circuit? [10%]

Solution:

NAND operation. The active load transistors, given in part (d) give clues on which transistors are responsible for the logical operations. When we remove the amplifier stage as well, which is given in part (e), the circuit is very similar to the NAND gate of CMOS.

(iii) Explain the operation of this circuit for AB=[00,10,11]. Comment on which transistors are ON and which are OFF. [30%]

Solution:

For AB=00, M1 and M2 are ON. Q1 base voltage is $V_{CC} - V_{TP}$, so M7 is ON. M7 discharges Q2 Base. Q1 base is charged by both M1 and M2. Q1 is in forward active.

For AB=10, it is the same as AB=00 except that M2 is OFF. Q1 is only charged by M1.

For AB=11, M1 and M2 are OFF. M3 and M4 are ON. Q1 is discharged through M3-M4 path. M5 and M6 are ON. Q2 base is charged through M5-M6 path. Q2 is

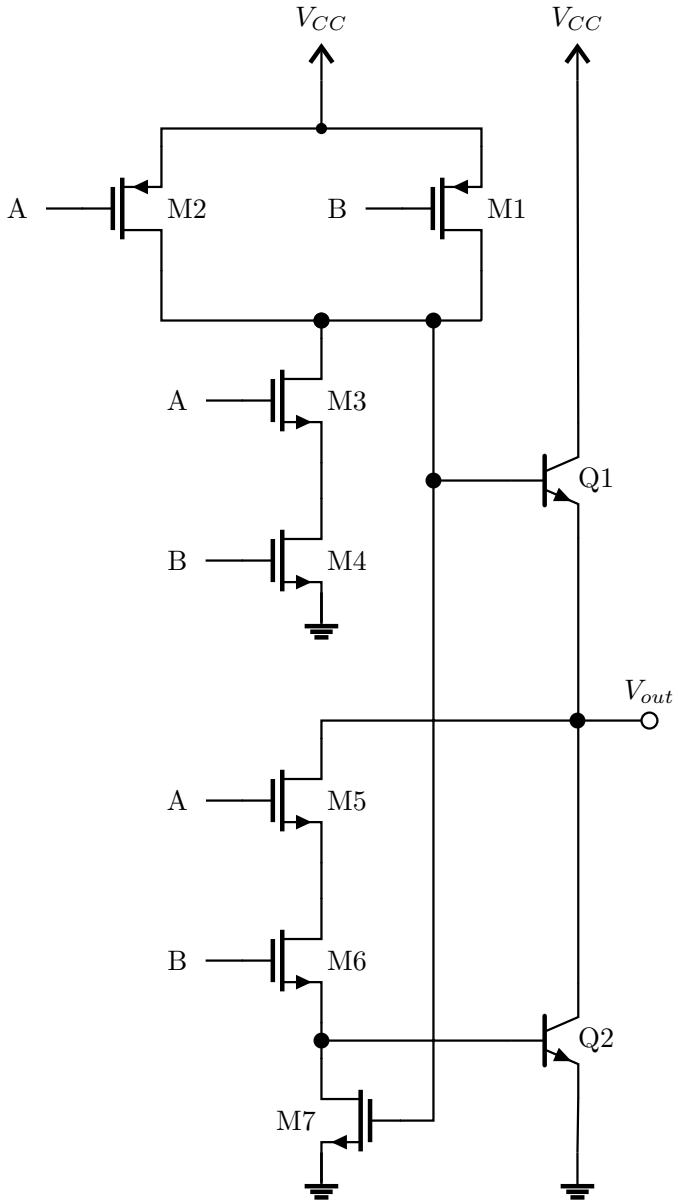


Fig. 6

in forward active.

(iv) The active loads M3, M4 and M7 are all NMOS. Briefly explain why active loads are NMOS rather than PMOS. [10%]

Solution:

NMOS devices are area efficient due to the higher mobility of electrons, therefore they are almost exclusively used as active loads. While this is enough, students can also mention that PMOS loads require negative potential, complicating the design further.

(v) Briefly explain why the amplifier transistors, Q1 and Q2, are both NPN, rather than PNP. [10%]

Solution:

Similar to the case of NMOS, NPNs are also area efficient. Since the only purpose of Q1 and Q2 is to amplify the output current, there is no reason to use a PNP which would require a higher surface area for the same amplification.

Assessor's Comments: This was the most popular question, with 61 out 75 candidates attempted. It has the average of 69.4 and the standard deviation of 21.7. The question measures the candidates' understanding of the transistor logic, their shortcomings and mitigation approaches. Part (a) and (b) were about the fundamentals, pros/cons of Schottky barrier diodes and bipolar devices, respectively. Students have successfully answered these. Part (c) focused on a given BiCMOS NAND gate and its detailed operation with certain inputs, which were mostly correctly identified by a large number of candidates as well.

END OF PAPER

THIS PAGE IS BLANK