***King Fahd University of Petroleum and Minerals***

***College of Computer Science and Engineering***

***Computer Engineering Department***

**COE 202: Digital Logic Design (3-0-3)**

**Term 132 (Spring 2013)**

**Major Exam II**

**Saturday April 19, 2014**

**Time: 120 minutes, Total Pages: 12**

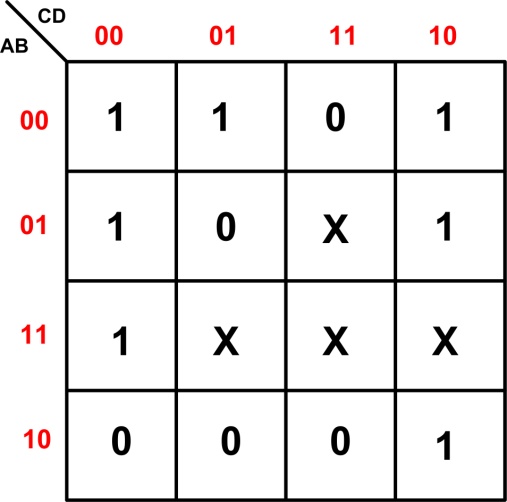
**Name:\_KEY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Section: \_\_\_\_\_\_\_**

**Notes:**

* Do not open the exam book until instructed
* **Calculators are not allowed** (*basic, advanced, cell phones, etc*.)
* Answer all questions
* All steps must be shown
* Any assumptions made must be clearly stated

|  |  |  |
| --- | --- | --- |
| **Question** | **Maximum Points** | **Your Points** |
| **1** | **8** |  |
| **2** | **12** |  |
| **3** | **8** |  |
| **4** | **12** |  |
| **5** | **13** |  |
| **6** | **10** |  |
| **7** | **7** |  |
| **Total** | **70** |  |

**Question 1 [8 Points]**



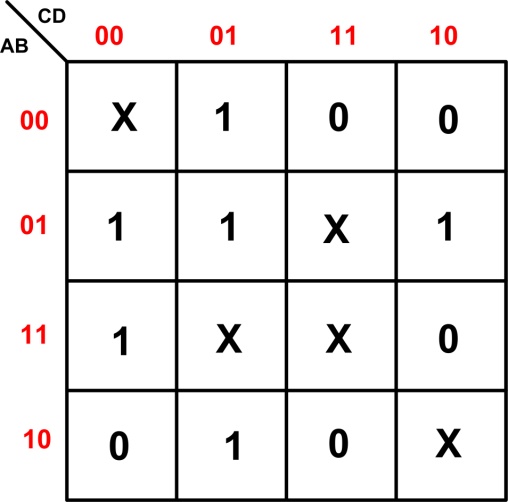
Shown to the right is the K-Map of the Boolean function **F** subject to the don’t care conditions **d**

**F**(A, B, C, D) = ∑(0, 1, 2, 4, 6, 10, 12)

**d**(A, B, C, D) = ∑(7, 13, 14, 15)

1. Derive the minimum **SOP** expression of F.

(1 point correct map + 1\*3 terms +1 overall correctness)



Shown to the right is the K-Map of the Boolean function **G** subject to the don’t care conditions **D**

**G**(A, B, C, D) = ∑(1, 4, 5, 6, 9, 12)

**D**(A, B, C, D) = ∑(0, 7, 10, 13, 15)

1. Derive the minimum **POS** expression of **G**.

**Question 2 [12 Points]**

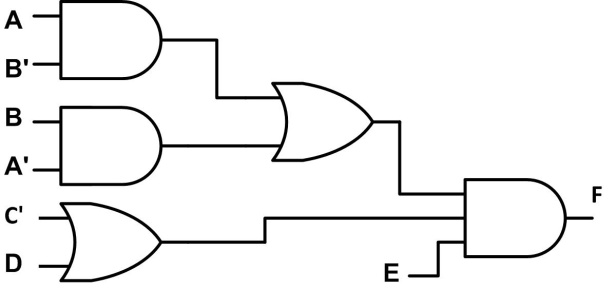
A logic circuit has *two* inputs ***x*** *&* ***y*** each is a 2-bit *unsigned* number. It has an output number ***z*** such that ***z =x2 + y*2**.



1. What is the minimum number of bits required for the output number ***z***?
2. Construct the truth table of the circuit.
3. Derive the Boolean expressions of the two least significant output bits (***z*0 *, z*1)** using basic gates (NO MSI parts)

**Question 3 [8 Points]**

1. Assuming the availability of the true and complement of signals A, B, C, and D, implement the function **F = ABC + DB’C’ +A’** using a minimum number of one gate type only.
2. Assuming the availability of the true and complement of signals A, B, C, and D, implement the function **F = (A+B+C) (D+B’+C’).D** using a minimum number of one gate type only.
3. Assuming the availability of the true and complement of signals A, B, C, D and E, implement the shown circuit using minimum number of NAND gates only.



**Question 4. (12 Points)**

Assuming that all numbers are held in 6-bit storage registers, answer the following:

1. If 2’s complement binary representation is used, what is the range of values that each number may assume? **(2 points)**
2. The largest number that can be subtracted from (-15) without causing overflow is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **(2 points)**
3. Perform the following arithmetic operations *in the indicated number representation*. Then, convert the result to decimal and indicate if an *overflow* has occurred: **(8 points)**
4. (10)10 – (24)10 (using sign-magnitude binary representation).

10<24 => -ve sign.

10=001010, 24=011000

|  |
| --- |
| 11000 |
| 01010– |
| 01110 = 14 |

=> the result = 101110=-14

No overflow

1. 010010 –111111 (using 1’s complement binary representation).

010010 + 000000 = 010010 =18

No overflow

1. 100000 –100011 (using 2’s complement binary representation).

=100000 + 011101

|  |
| --- |
| 100000 |
| 011101+ |
| 111101 |

=-000011=-3

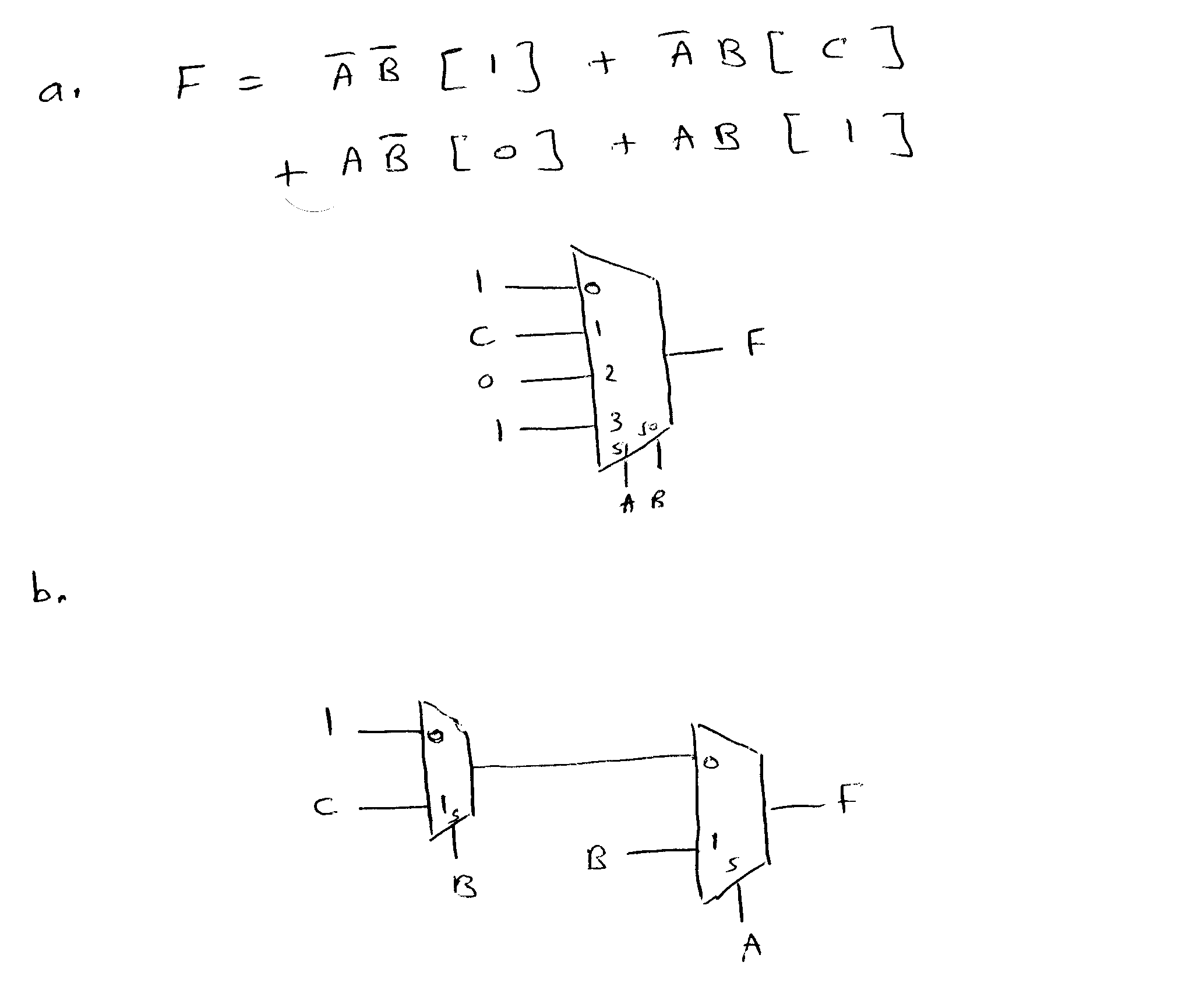
No overflow

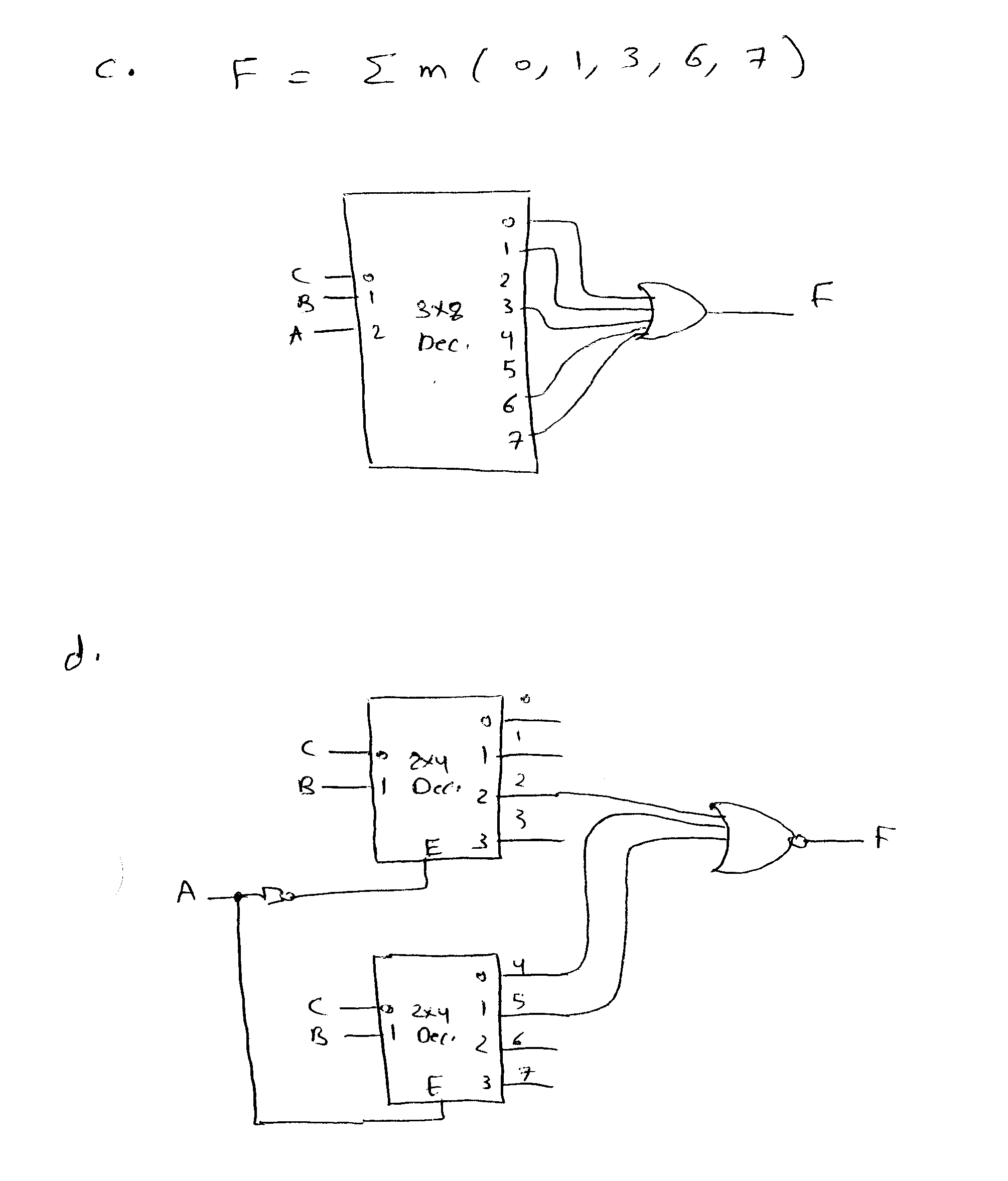
1. 010111 – 11 0111 (using 2’s complement binary representation).

**Question 5. (13 Points)**

Implement the Boolean function:

1. Using a single 4x1 multiplexer. (4 Points)
2. Using a minimum number of 2x1 multiplexers. (2 Points)
3. Using a single 3x8 decoder and an OR gate. (3 Points)
4. Using a single NOR gate and the minimum number of 2x4 decoders with enable. (4 Points)



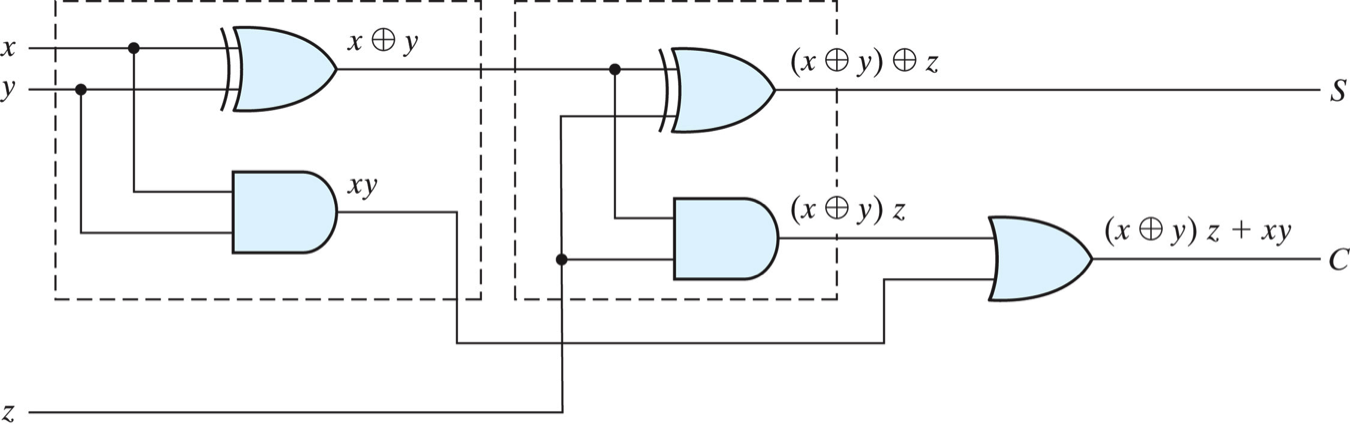
****

**Question 6. (10 Points)**

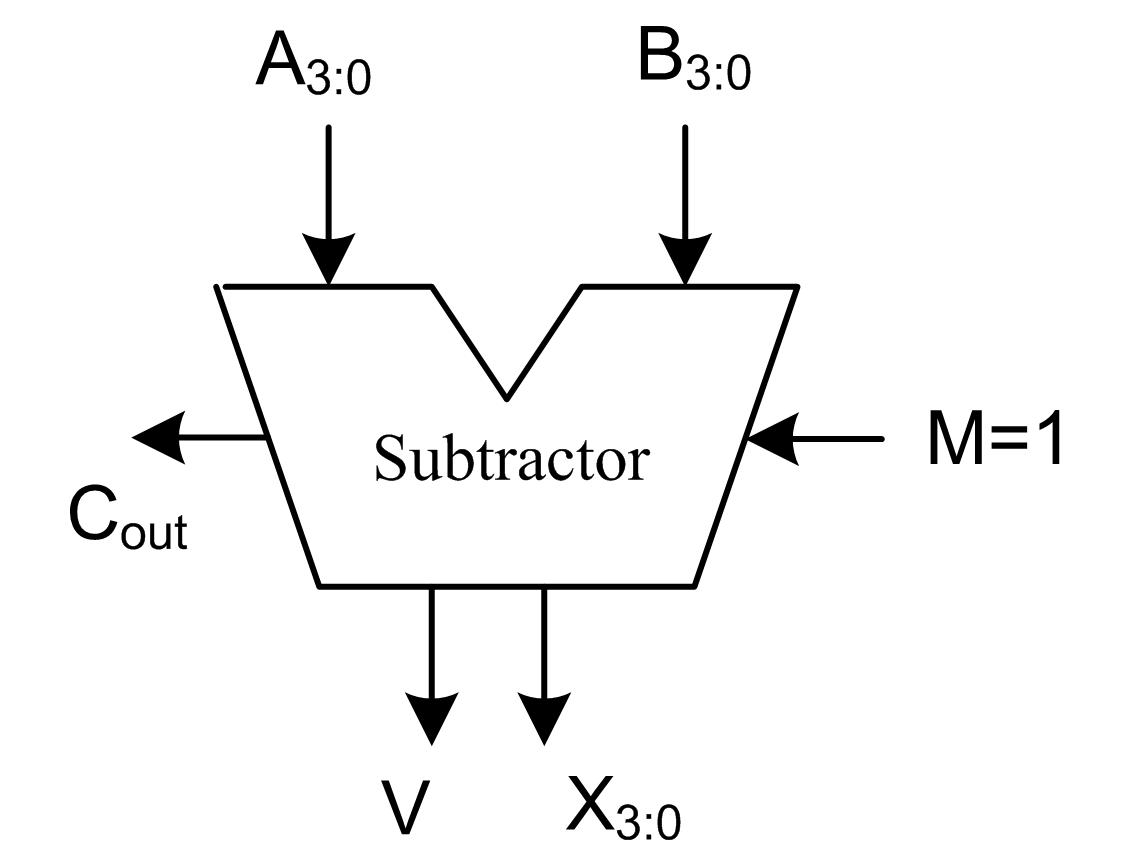
1. Design a 4-bit adder/subtractor circuit which uses the least number of Full-Adders (FAs). The circuit receives two 4-bit signed numbers **A** and **B** (2’s complement representation) and one control input (). If the control input M =0, the 4-bit circuit output equals (**A+B**). If the control input M =1, it equals (**A-B**). The circuit has another output **V** which equals **1** **only** in case of *overflow*.

|  |  |
| --- | --- |
| **Gate** | **Delay (ns)** |
| AND | 2 |
| OR | 2 |
| XOR | 3 |

1. Given the FA circuit shown below, calculate the worst-case delay of this adder/subtractor circuit assuming gate delays as given in the table to the right.

****

**Question 7. (7 Points)**



A 4-bit adder/subtrctor circuit like the one designed in problem 6, is used here as a **subtractor** with the input control **M=1** (see Figure).

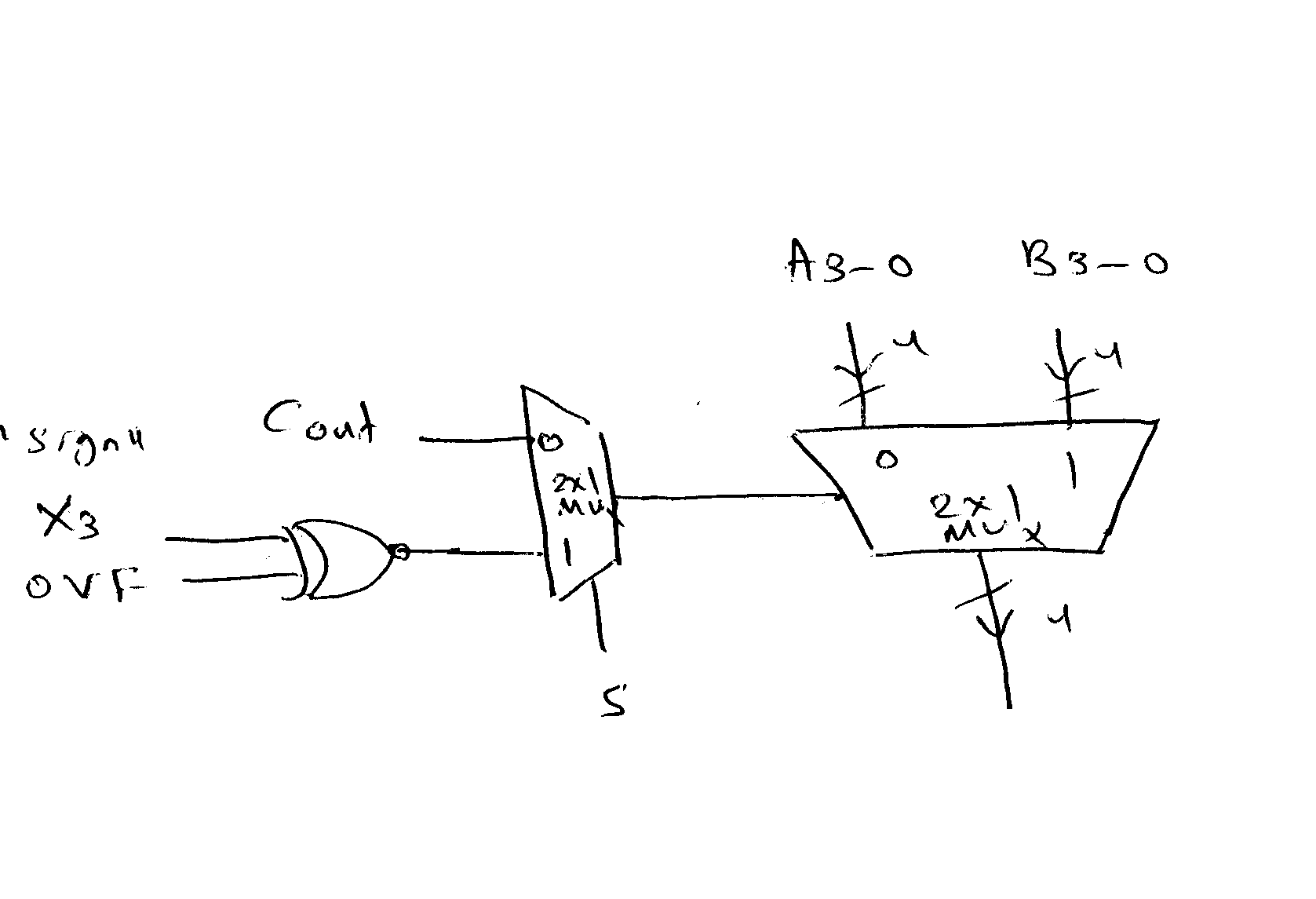
It subtracts two 4-bit numbers (**A**, and **B**) producing a 4-bit result (**X**). It also produces the overflow flag **V**, and **C**out**.**

This **subtractor** can be used to compare both *unsigned* and *signed* 4-bit input numbers (**A** and **B)** by computing **(A-B)**. It can be shown that the comparator output (**A ≥ B**) is given by:

|  |  |
| --- | --- |
| **Type of Input Operands (A & B)** | **Comparator Output (A** ≥ **B)** |
| **Unsigned** | = 1 iff **C**out = 1  = 0 otherwise |
| **Signed**  (2’s Complement) | = 1 iff **V = Sign** of the result **X**  = 0 Otherwise |

Using this subtractor, design a circuit that compares two 4-bit input numbers A3-0 and B3-0 to output the larger of the two. The input numbers (A & B) *may be signed or unsigned*. An additional input signal **S** indicates whether the input numbers are **signed** (**S=1**) or **unsigned** (**S=0**).

In addition to the subtractor, you *may use* multiplexers of any size, and other needed gates. **You MAY NOT USE** any magnitude comparator.  **(7 Points)**



OR (less efficient solution):

