Nov. 28, 2010

COMPUTER ENGINEERING DEPARTMENT

COE 561

Digital System Design and Synthesis

Major Exam I

(Open Book Exam)

First Semester (101)

Time: 1:00-3:30 PM

Student Name : \_KEY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID. : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **Question** | **Max Points** | **Score** |
| **Q1** | **15** |  |
| **Q2** | **15** |  |
| **Q3** | **10** |  |
| **Q4** | **20** |  |
| **Q5** | **20** |  |
| **Q6** | **20** |  |
| **Total** | **100** |  |

# **[15 Points]**

# **(Q1)** Draw the ROBDD for the function F=a⊕b⊕c⊕d, with the variable ordering {a, b, c, d}. How can we easily obtain the ROBDD for from the ROBDD for F? Don’t draw the ROBDD for , just explain.

# **scan0001.jpg**

# 

# **[15 Points]**

# **(Q2)** Write an algorithm, called **ROBDD**, that receives a function F and a variable ordering and constructs and ROBDD for the input function. Explain clearly the terminal cases and the structure of the tables you will use in your algorithm.

ROBDD(F){

If (terminal case)

return (r = trivial result)

else {

if (computed table has entry (F, r) )

return (r from computed table)

else {

x is top variable of F

t = ROBDD(Fx)

e = ROBDD(Fx’)

if ( t == e) return (t)

r = find\_or\_add\_unique\_table (x, t, e)

Update computed table with (F, r)

return (r)

}

}

}

Terminal cases are when F is a single literal i.e. x or x’ or 0 or 1.

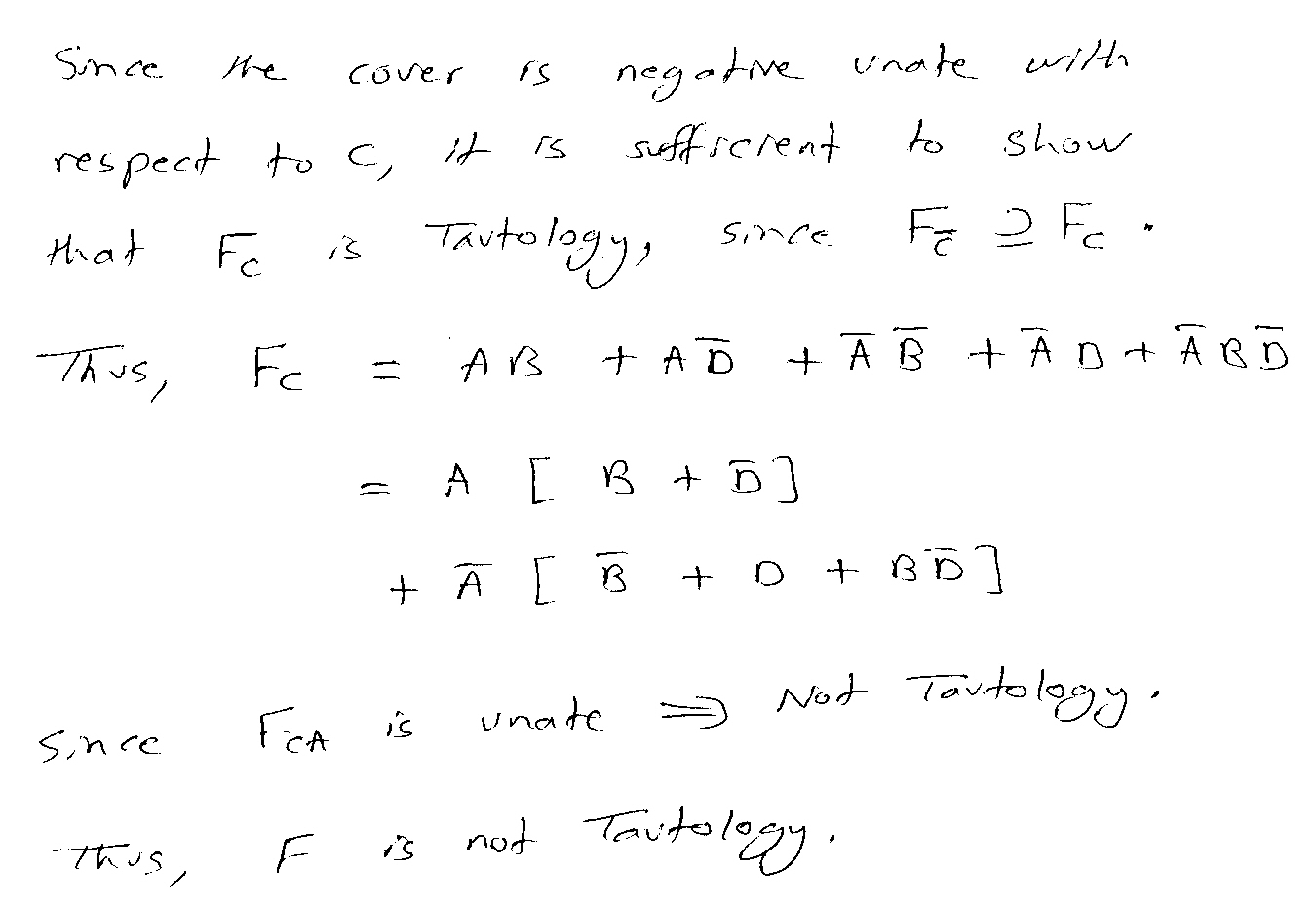
The unique table contains a key for a vertex of an ROBDD where the key is a triple of variable, identifiers of right and left children.

The computed table stores a function and its identifier in the form (F, r) to improve the performance of the algorithm.

# **[10 Points]**

# **(Q3)** Consider the function. Using recursive paradigm, determine if the function F is **tautology** or not. You need to choose the right variable for expansion to minimize computations.

# 

****

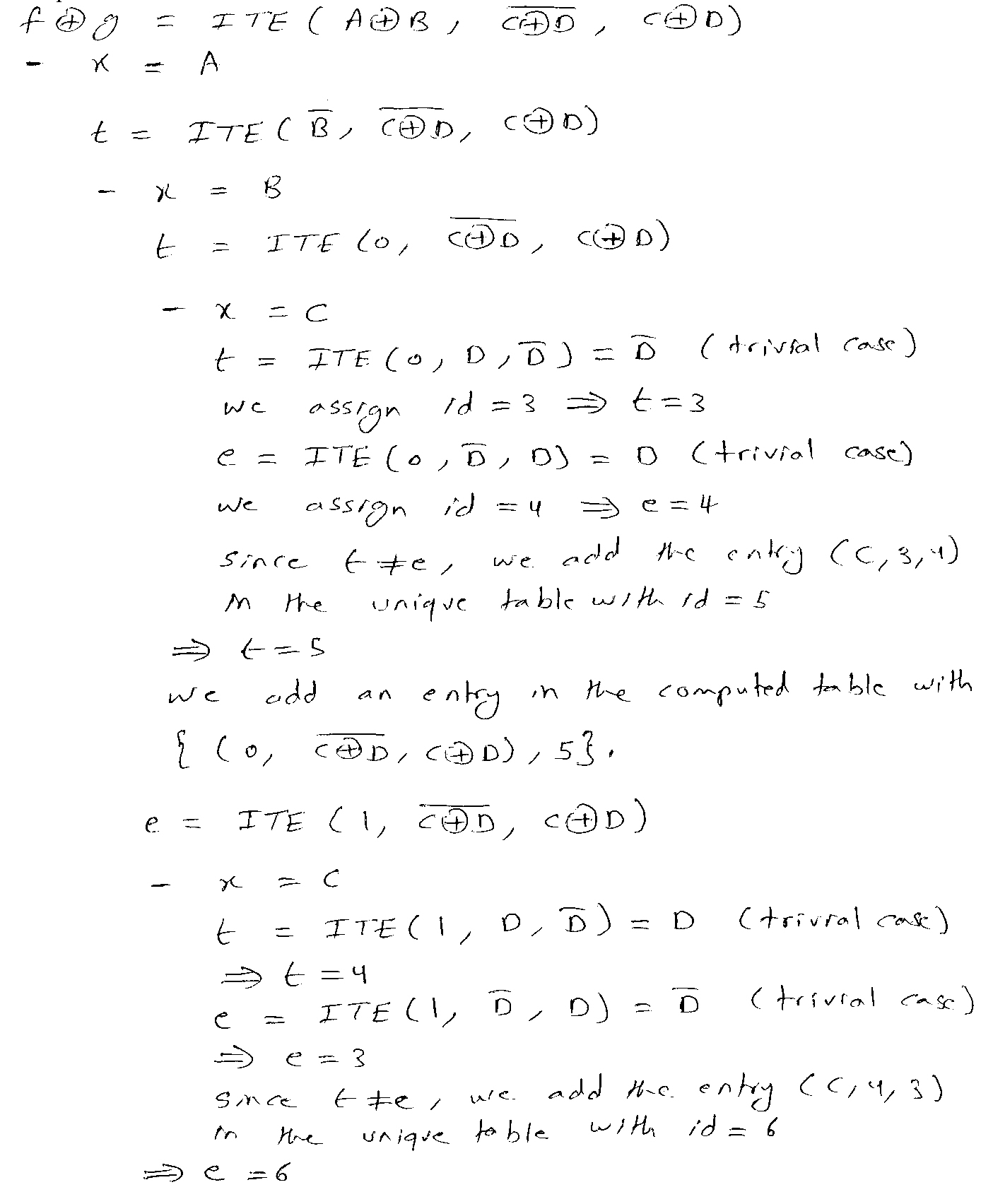
# **[20 Points]**

# **(Q4)** Consider the two Boolean functions and given below:

# 

# 

# Draw the **ITE DAG** for the function using the variable order {A, B, C, D}. Show all the details of your solution using ITE procedure including the resulting unique table and computed table.



## scan0004.jpg

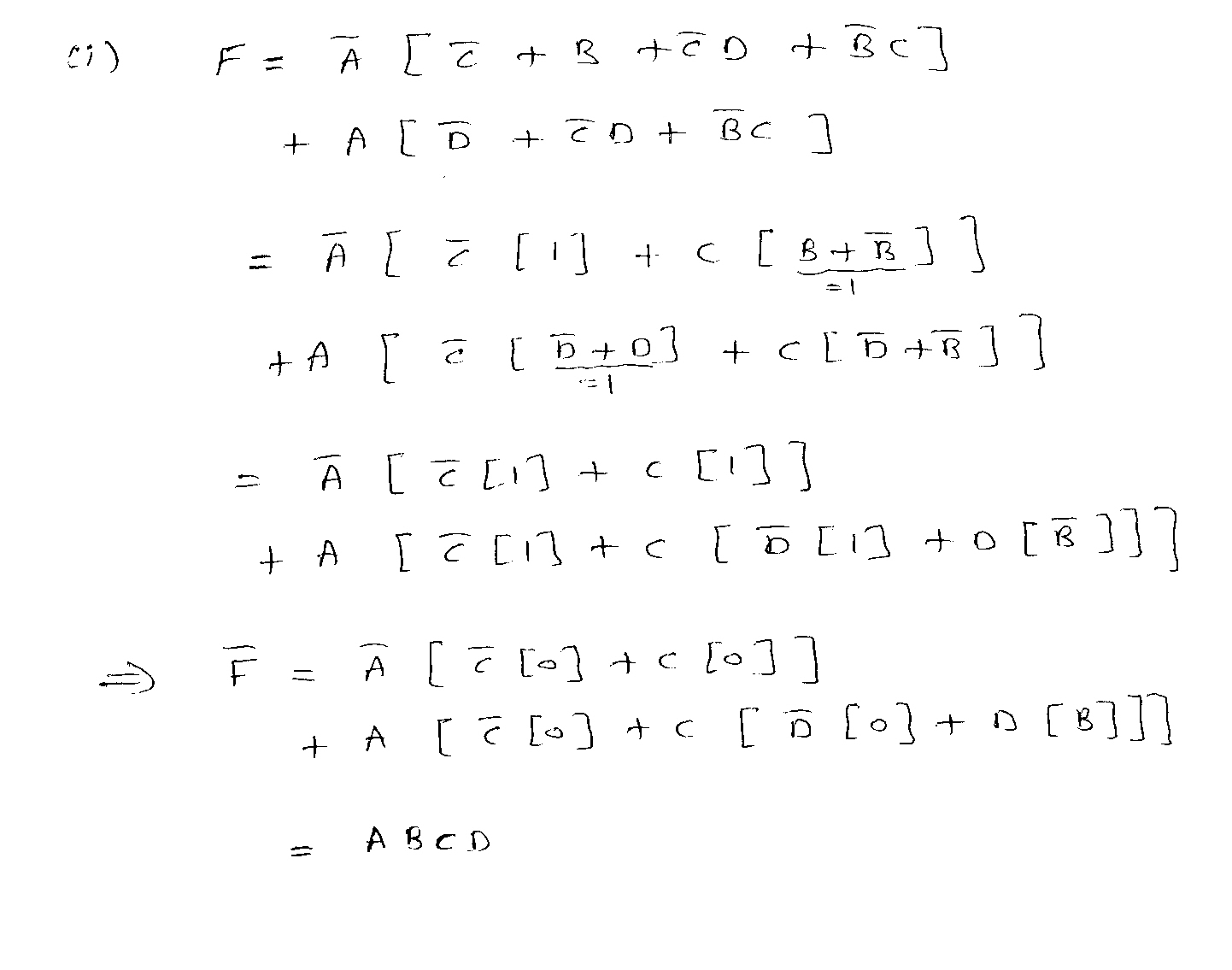
# 

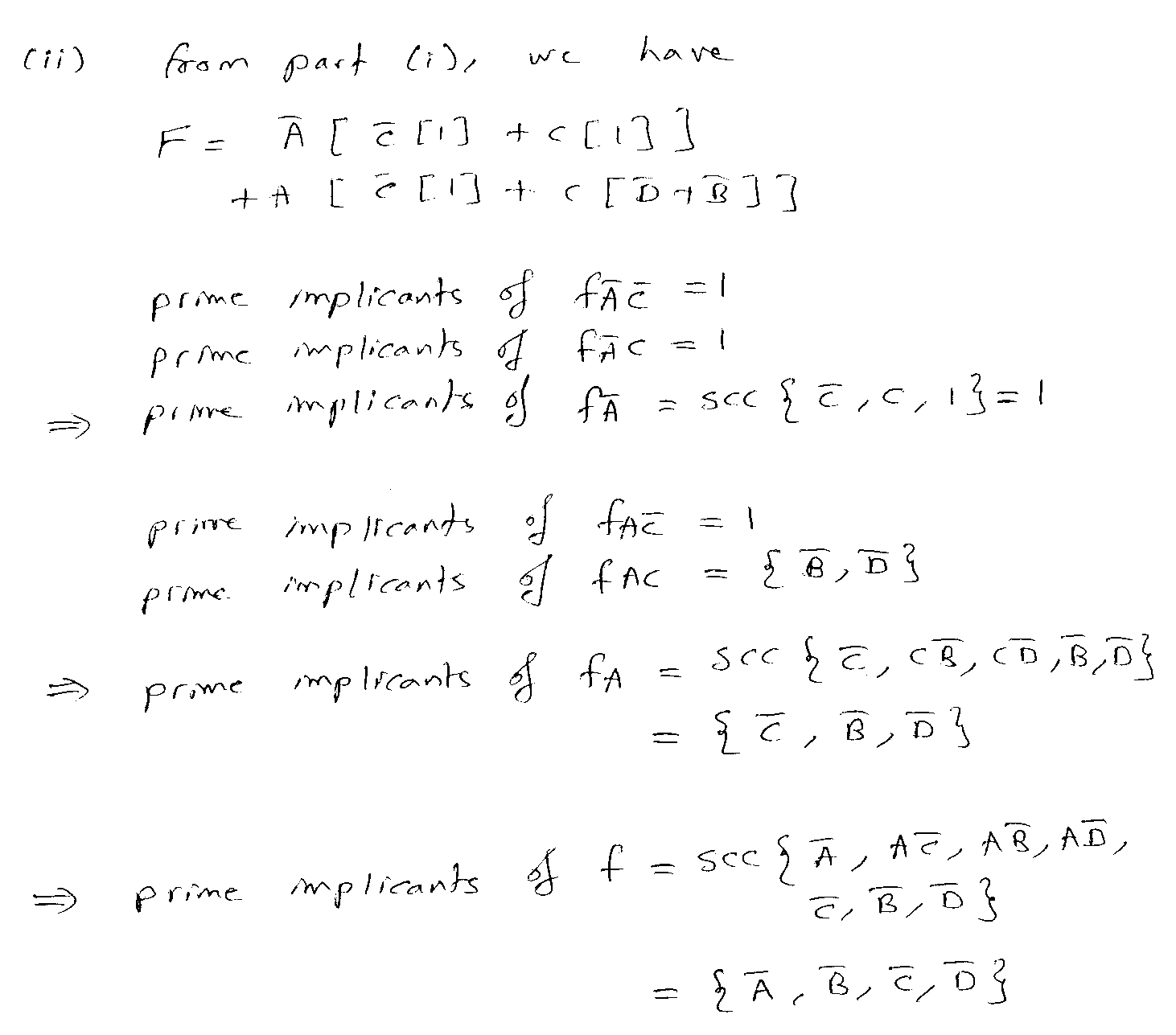
# **[20 Points]**

# **(Q5)** Consider the function

## Compute the **complement** of the function using the recursive complementation procedure outlined in section 7.3.4.

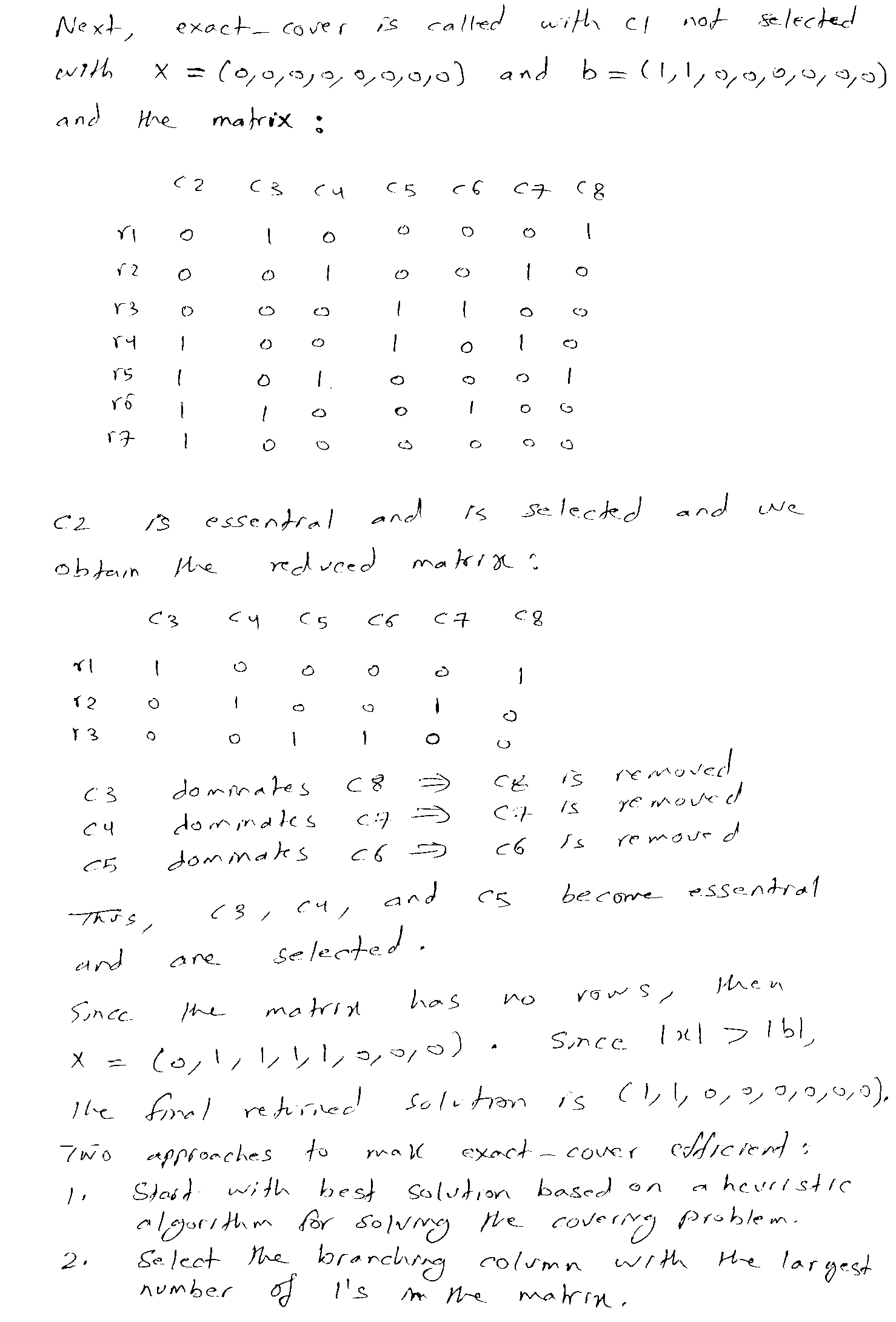
## **C**ompute all the **prime implicants** of the function using the method outlined in section 7.3.4.



****

# 

# **scan0007.jpg**

****