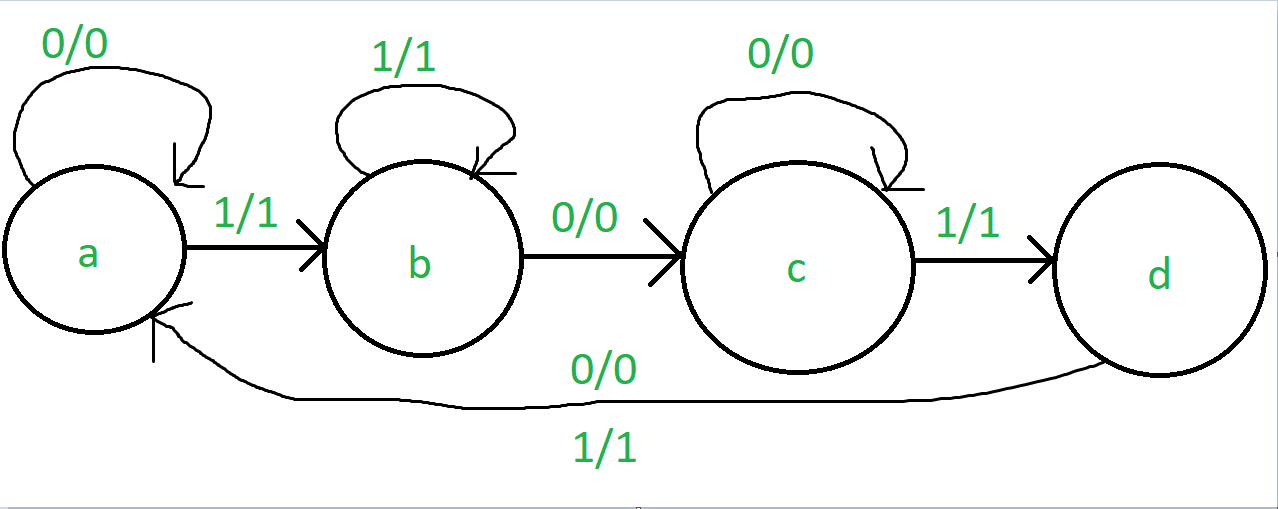
**Computer Architecture and Systems Software**

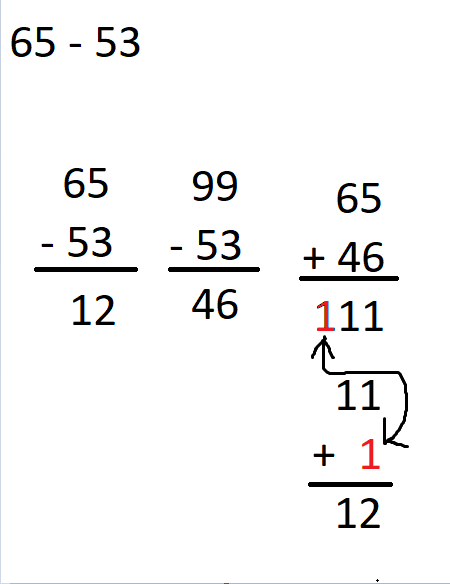
Date 10 – 12 May 2021 10:00 – 17:00

**Open Book Exam**

Question 1:

1. Give the state diagram for a ‘sequence detector’ whose output is 1 only if there is the sequence 1011 (first 1, second 0, third 1, fourth 1) at the input.



1. Perform the following subtraction using the 9s complement method: 65 – 53. Explain how this method reduces the subtraction into addition.

As shown, 65 – 53 = 12, but with the 9s compliment, I took the 53 and used it to reduce 99, as that would give me the 9s compliment of 53, that being 46.

With that, I added 65 (which is bigger than 53 so I used that from the original calculation), and added that to the 9s compliment of 53, which is 46.

This gave me 111, always when using 9s compliment with a first number being bigger than the second, there will always be an additional digit, in this case the hundredth digit. I took that away from the 111, and made a new sum with it, 11 + 1, which = 12 also, this is how you can turn a minus calculation into an addition using 9s compliment.

Question 2:

1. Use truth tables to prove that:

|  |  |  |
| --- | --- | --- |
| X | X’ | X . X’ |
| 0 | 1 | 0 |
| 1 | 0 | 0 |

So no matter what number X is, X . X’ will always be 0 as the compliment has to be the opposite.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | Y | X + Y | X’ | X’ . Y | X + X’ . Y |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |

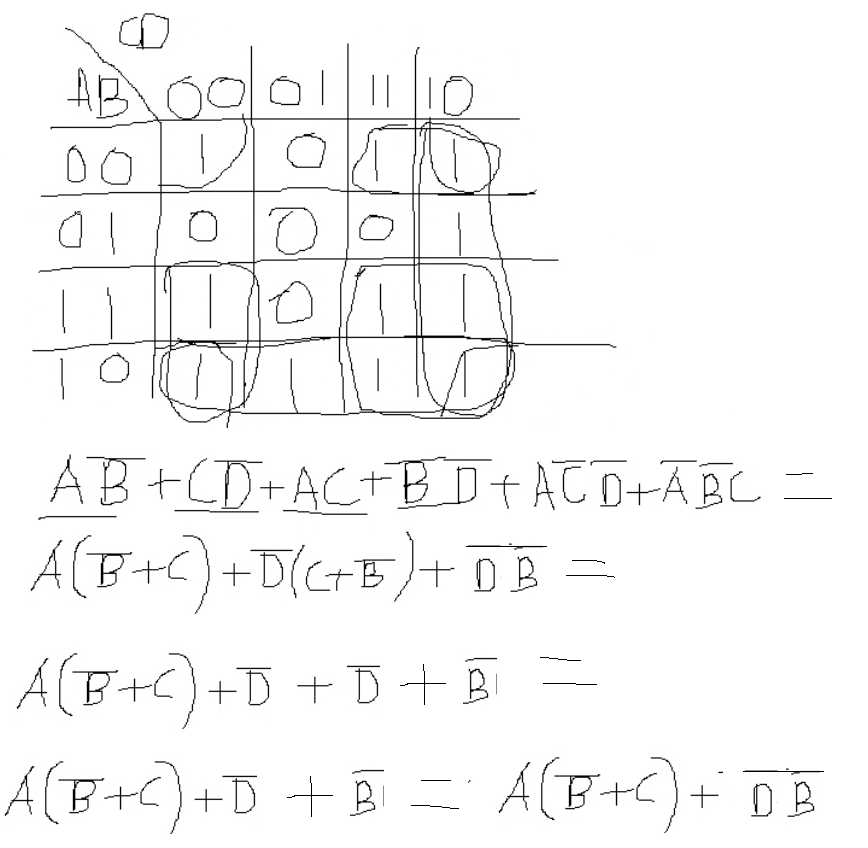
With this, as we can see whatever X + Y is, X + X’ . Y will = the same

1. Given the Boolean function 

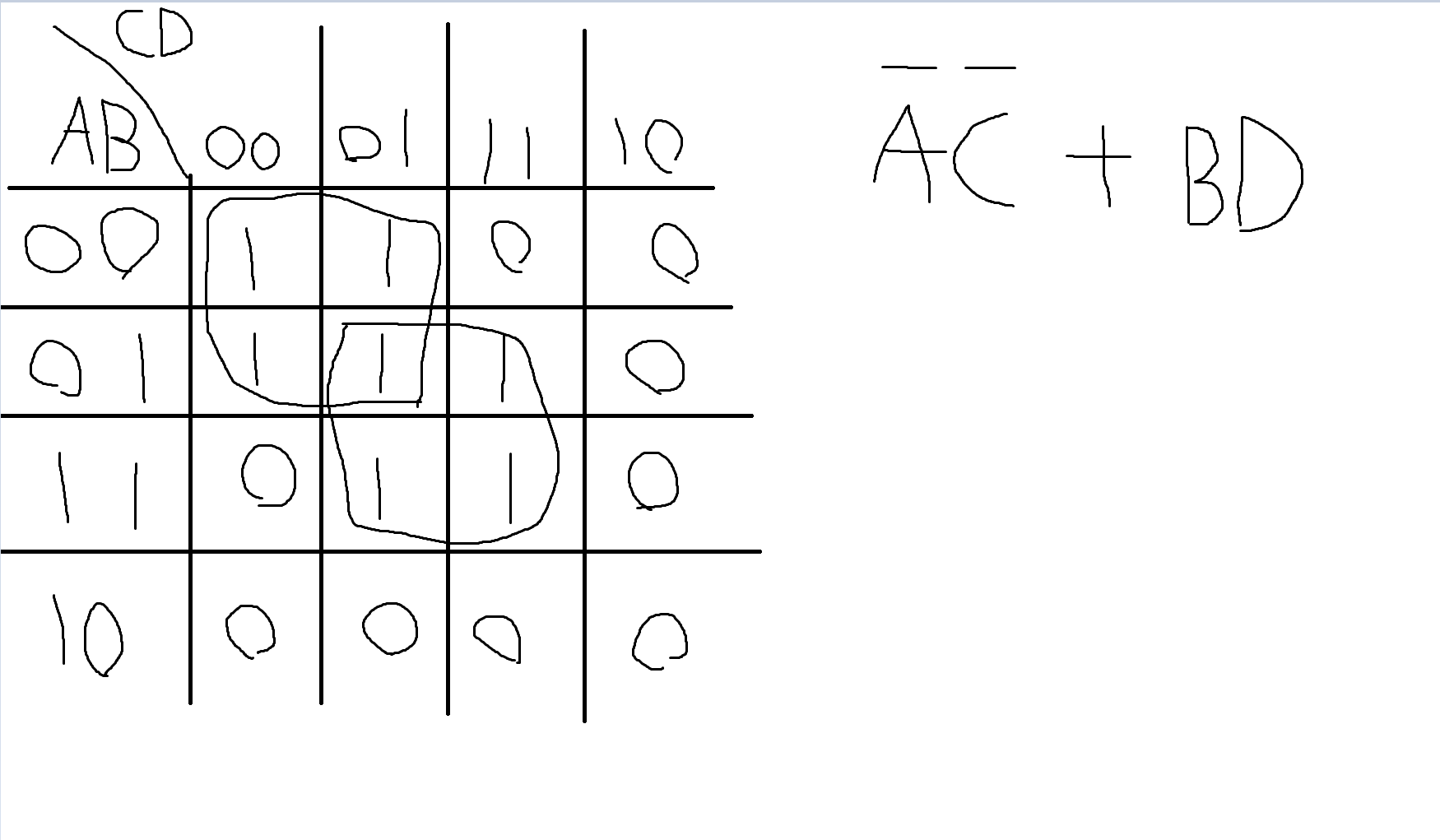
Use a Karnaugh map to derive a simpler equivalent expression as:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | Output | SOP | POS |
| 0 | 0 | 0 | 0 | 1 | ABCD | A’B’C’D’ |
| 0 | 0 | 0 | 1 | 1 | ABCD’ | A’B’C’D |
| 0 | 0 | 1 | 0 | 0 | A’B’CD’ | ABC’D |
| 0 | 0 | 1 | 1 | 0 | A’B’CD | ABC’D’ |
| 0 | 1 | 0 | 0 | 1 | AB’CD | A’BC’D’ |
| 0 | 1 | 0 | 1 | 1 | AB’CD’ | A’BC’D |
| 0 | 1 | 1 | 0 | 0 | A’BCD’ | AB’C’D |
| 0 | 1 | 1 | 1 | 1 | AB’C’D’ | A’BCD |
| 1 | 0 | 0 | 0 | 0 | AB’C’D’ | A’BCD |
| 1 | 0 | 0 | 1 | 0 | AB’C’D | A’BCD’ |
| 1 | 0 | 1 | 0 | 0 | AB’CD’ | A’BC’D |
| 1 | 0 | 1 | 1 | 0 | AB’CD | A’BC’D’ |
| 1 | 1 | 0 | 0 | 0 | ABC’D’ | A’B’CD |
| 1 | 1 | 0 | 1 | 1 | A’B’CD’ | ABC’D |
| 1 | 1 | 1 | 0 | 0 | ABCD’ | A’B’C’D |
| 1 | 1 | 1 | 1 | 1 | A’B’C’D’ | ABCD |

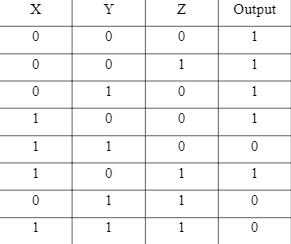
* sum-of-products



* products-of-sums



Question 3:

1. Write the Boolean expression for the truth table
2. Give your results in ‘sum-of-products’ form and draw the corresponding ‘AND-to-OR’ gate network

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Z | Output | SoP |
| 0 | 0 | 0 | 1 | X’Y’Z’ |
| 0 | 0 | 1 | 1 | X’Y’Z |
| 0 | 1 | 0 | 1 | X’YZ’ |
| 1 | 0 | 0 | 1 | XY’Z’ |
| 1 | 1 | 0 | 0 | XYZ’ |
| 1 | 0 | 1 | 1 | XY’Z |
| 0 | 1 | 1 | 0 | X’YZ |
| 1 | 1 | 1 | 0 | XYZ |

1. Give your results in ‘product-of-sums’ form and draw the corresponding ‘AND-to-OR’ gate network

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Z | Output | PoS |
| 0 | 0 | 0 | 1 | X+Y+Z |
| 0 | 0 | 1 | 1 | X+Y+Z’ |
| 0 | 1 | 0 | 1 | X+Y’+Z |
| 1 | 0 | 0 | 1 | X’+Y+Z |
| 1 | 1 | 0 | 0 | X’+Y+Z’ |
| 1 | 0 | 1 | 1 | X+Y’+Z’ |
| 0 | 1 | 1 | 0 | X+Y’+Z’ |
| 1 | 1 | 1 | 0 | X’+Y’+Z’ |

1. Discuss the analogy between set theory and Boolean algebra. For the various operations in set theory, explain what are the corresponding operations in Boolean algebra.

In set Theory, there are lots of terms necessary, however when comparing to Boolean Algebra there are lots of similar terms that use different symbols, for instance

A ∪ B would mean A “OR” B (And can be both), where as in Boolean Algebra they instead use the + operator, so in this example that would be A + B.

Other examples include:

A ∩ B = “AND”, in Boolean Algebra this is A · B

A ⊻ B = “XOR” (meaning Exclusively Or, has to be one or the other, can not be both), where as in Boolean algebra this is A ⊕ B

In Boolean Algebra there is the negation symbol which can be represented as ‘, for example A’, and can also be called “compliment”, in Set Theory the ‘ symbol can also represent compliment, meaning that all the objects that do not belong to set A, so this is slightly different to each other and is not the same.

Then lastly, in Boolean Algebra you would combine these symbols to create the other gates such as NOR being  (the  Being another way to represent compliment), and in set theory this is similar where you would represent this by using pre existing terms.