Name: KEY Id#

COE 202, Term 151

Digital Logic Design

Quiz# 3

Date: Sunday, Oct. 25

# 

**Q1**. *Assuming the availability of all variables and their complements*, simplify the following two Boolean functions F and G subject to the given don’t care conditions d1 and d2 using the K-Map method:

## Implement F using only NOR gates:

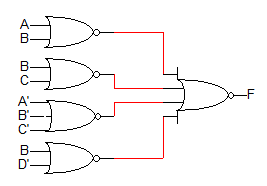
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **C D** | | | |
| 00 | 01 | 11 | 10 |
| **A B** | 00 |  |  | X |  |
| 01 | 1 | 1 | X | 1 |
| 11 | 1 | 1 |  |  |
| 10 |  | X |  | 1 |

F(A, B, C, D) = ∑(4, 5, 6, 10, 12, 13)

d1(A, B, C, D) = ∑(3, 7, 9)

To get a 2-Level NOR-NOR implementation, we use the simplified POS expression (Groups of 0’s) given by:

F= (A+B) . (B+C). (A’+B’+C’).{(C’+D’) **or** (B+D’)**}**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **C D** | | | |
| 00 | 01 | 11 | 10 |
| **A B** | 00 | 1 |  | X | 1 |
| 01 |  |  | X | X |
| 11 | X | 1 | 1 |  |
| 10 | 1 | X | 1 |  |

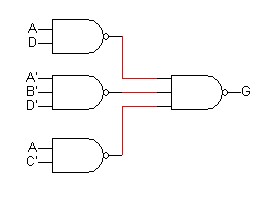
## Implement G using only NAND gates:

G(A, B, C, D) = ∑(0, 2, 8, 11, 13, 15)

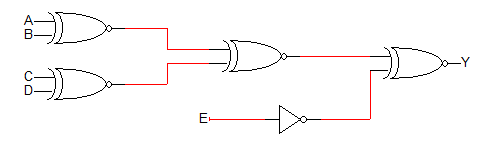
d2(A, B, C, D) = ∑(3, 6, 7, 9, 12)

Simplified SOP expression directly maps into a 2-Level NAND-NAND implementation.

G = A D + A' B' D' + A C' OR G = A D + B' C' D' + A' C



**Q2**. Implement the following circuit using only 2-input XOR gates with minimal number of gates:



Solution:

