Name: KEY Id#

COE 202, Term 112

Digital Logic Design

Quiz# 4

Date: Saturday, April 7

# 

# **Q1**.

1. Determine the decimal value of the 8-bit binary number (11010100) when interpreted as:

|  |  |  |  |
| --- | --- | --- | --- |
| An unsigned number | A signed-magnitude number | A signed-1’s complement number | A signed-2’s complement number |
| 212 | -84 | -43 | -44 |

ii. Represent the decimal value (- 40) in binary using a total of 8 bits in the following notations:

|  |  |  |
| --- | --- | --- |
| A signed-magnitude number | A signed-1’s complement number | A signed-2’s complement number |
| 10101000 | 11010111 | 11011000 |

iii. Perform the following signed-2’s complement arithmetic operations in binary using 5 bits. All numbers given are represented in the signed-2’s complement notation. Indicate clearly the carry values from the last two stages. For each of the three operations, check and indicate whether overflow occurred or not.

|  |  |  |  |
| --- | --- | --- | --- |
|  | a. 01001  +10111  00000 | b. 01000  - 10010  01000  + 01110  10110 | c. 11010  - 01101  11010  + 10011  01101 |
| Overflow Occurred? (Yes/No) | No | Yes | Yes |

**Q2**. Design a combinational circuit that receives a 4-bit unsigned number **I= I3I2I1I0** as input and generates the remainder of dividing this number by 3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I3 | I2 | I1 | I0 | R1 | R0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

# Using K-map, we will get the equations for R1 and R0 as follows:

R1= I3’I2’I1I0’ + I3’I2I1’I0 + I3I2I1I0’ + I3I2’I1’I0’ + I3I2’I1I0

= I1I0’ (I3 ⊕ I2)’ + I3I2’(I1 ⊕ I0)’ + I3’I2I1’I0

R1= I3’I2’I1’I0 + I3’I2I1’I0’ + I3’I2I1I0 + I3I2I1’I0 + I3I2’I1I0’

= I1’I0 (I3 ⊕ I2)’ + I3’I2(I1 ⊕ I0)’ + I3I2’I1I0’