**Matrix Manipulation Constraints Foundation for Symmetric Matrices:**

**Owned By: Het Dani**

|  |  |  |  |
| --- | --- | --- | --- |
| Index | 0 | 1 | 2 |
| 0 | **5 (0,0)** | **6 (0,1)** | **7 (0,2)** |
| 1 | **9 (1,0)** | **9 (1,1)** | **1 (1,2)** |
| 2 | **1 (2,0)** | **2 (2,1)** | **3 (2,2)** |

We have a 3x3 matrix highlighted in yellow color above. This solution to the problems around matrix manipulation can scale for any NxN symmetric matrix.

The rows will be referenced using iterator “i” & the columns using iterator “j” for simplicity purposes.

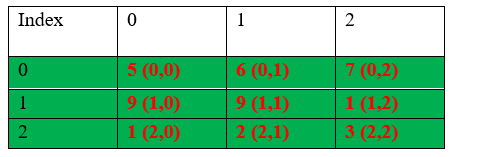
**Basics:**

1)**Rows:** The ones highlighted in ***green color*** below are the rows of the 3x3 matrix.

a) Row0 – (0,0), (0,1), (0,2)

b) Row1 – (1,0), (1,1), (1,2)

c) Row2 – (2,0), (2,1), (2,2)

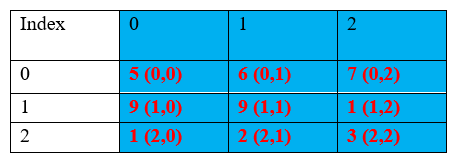


2)**Columns:** The ones highlighted in ***blue color*** below are the columns of the 3x3 matrix.

a) Col0 – (0,0), (1,0), (2,0)

b) Col1 – (0,1), (1,1), (2,1)

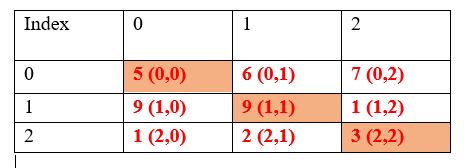
c) Col2 – (0,2), (1,2), (2,2)



3)**(Main) Diagonal**: The ones highlighted in ***orange color*** below are the columns of the 3x3 matrix.

a) Main Diagonal – (0,0), (1,1), (2,2)

🡪***If “i” represents rows & “j” represents columns of a matrix then main diagonals will always be located at i==j locations in a symmetric matrix.***

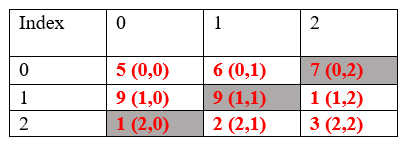


4)**(Anti)-Diagonal**: The ones highlighted in ***grey color*** below are the columns of the 3x3 matrix.

a) Anti-Diagonal – (0,2), (1,1), (2,0)

🡪***If “i” represents rows & “j” represents columns of a matrix then anti-diagonals will always be situated such that it abides by the following relation:***

(***i + j == sizeof(matrix) – 1)***



With the foundational understanding around 2-D matrix above, the below-mentioned reference links will help perform a 2-D matrix manipulation on either rows, columns diagonal or anti-diagonal or any combination between rows, columns, diagonal & anti-diagonal.

**References:** (For accurate response, refer to Dave rich’s comments & solutions in the below-mentioned reference links)

1)How to access columns in a 2-d array: <https://verificationacademy.com/forums/t/how-to-constrain-the-2nd-dimension-of-a-2d-array-in-systemverilog/37993/2>

2)How to access diagonals & anti-diagonals in a 2-d arrays: <https://verificationacademy.com/forums/t/magic-square-on-system-verilog/36496/11>

3)Adjacent neighboring elements in a 2d-array are different:- <https://verificationacademy.com/forums/t/constraint-question-write-a-constraint-for-2d-array-where-value-of-an-element-should-be-different-from-its-neighbor/45757/5>