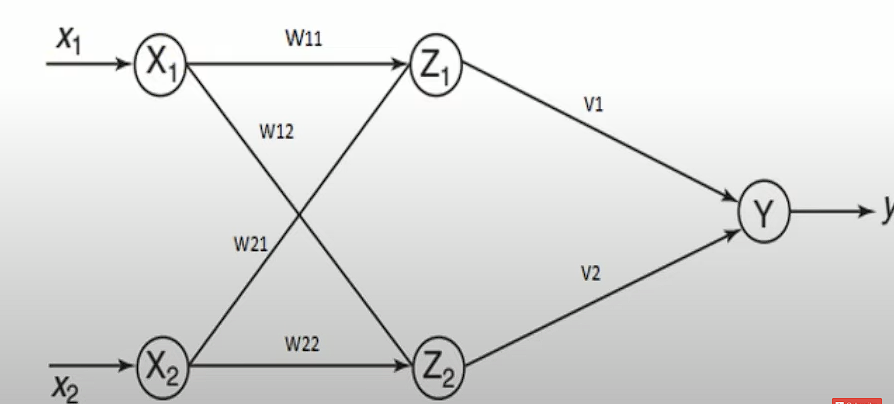
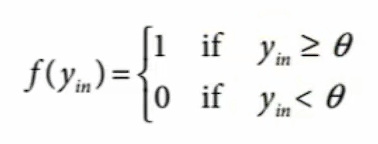
**Experiment No.: 02**

**Aim:** Write a program to implement logical XOR using McCulloch Pitts neuron model



**Steps:**

1. **Initialize Weights and Threshold:**
   * Set the weights: **w11 = 1**, **w12 = -1**, **w21 = -1**, **w22 = 1**.
   * Set the thresholds: **theta = 1**.
2. **Define Input Values:**
   * Given input values: **x1 = [0, 0, 1, 1]** and **x2 = [0, 1, 0, 1]**.
3. **Calculate zin1 and zin2:**
   * For each input pair (x1[i], x2[i]), calculate **zin1** and **zin2** using the formulas:
     + **zin1[i] = x1[i] \* w11 + x2[i] \* w21**
     + **zin2[i] = x1[i] \* w12 + x2[i] \* w22**
4. **Define Activation Function:**



* + Define an activation function **activation(z)** which outputs 1 if **z >= theta** and 0 otherwise.

1. **Apply Activation Function to zin1 and zin2:**
   * For each value in **zin1** and **zin2**, apply the activation function to obtain **z1** and **z2**:
     + **z1[i] = activation(zin1[i])**
     + **z2[i] = activation(zin2[i])**
2. **Calculate yin:**
   * For each input pair, calculate the weighted sum **yin** using the formulas:
     + **yin[i] = z1[i] \* v1 + z2[i] \* v2**

**code:**

w11 = 1

w12 = -1

w21 = -1

w22 = 1

v1 = 1

v2 = 1

theta = 1 # Threshold value

# Given input values

x1 = [0, 0, 1, 1]

x2 = [0, 1, 0, 1]

# Calculate zin1 and zin2

zin1 = [x1[i] \* w11 + x2[i] \* w21 for i in range(4)]

zin2 = [x1[i] \* w12 + x2[i] \* w22 for i in range(4)]

# Activation function for z1 and z2

activation = lambda z: 1 if z >= theta else 0

# Apply activation function to zin1 and zin2

z1 = [activation(z) for z in zin1]

z2 = [activation(z) for z in zin2]

# Calculate yin

yin = [z1[i] \* v1 + z2[i] \* v2 for i in range(4)]

print('Calculated z1:')

print(z1)

print('Calculated z2:')

print(z2)

print('Calculated output:')

print(yin)

**Output:**

