Assignment 5

In [1]:
```
import numpy as np
```

Define BAM Network

In [2]:
```
class BAM:
    def __init__(self, input_size, output_size):
        self.weights = np.zeros((output_size, input_size))

    def train(self, input_patterns, output_patterns):
        for i in range(input_patterns.shape[0]):
            x = input_patterns[i]
            y = output_patterns[i]
            self.weights += np.outer(y, x)

    def recall_input(self, output_patterns):
        return np.dot(self.weights, output_patterns)

    def recall_output(self, input_patterns):
        return np.dot(self.weights, input_patterns)
```

Implement BAM Network

In [3]:
```
input_size = 2
output_size = 2
bam = BAM(input_size, output_size)
```

Train the Network

In [8]:
```
input_patterns = np.array([[1, -1], [-1, 1]])
output_patterns = np.array([[-1, 1], [1, -1]])
bam.train(input_patterns, output_patterns)
```

Test input to output

In [5]:
```
test_input = np.array([1, -1])
output_recall = bam.recall_output(test_input)
print("Input:", test_input)
print("Recalled Output:", output_recall)
```

```
Input: [ 1 -1]
Recalled Output: [-4.  4.]
```

Test output to input

In [6]:
```
test_output = np.array([-1, 1])
input_recall = bam.recall_input(test_output)
print("Input Recall:", input_recall)
print("Recalled Output:", test_output)
```

```
Input Recall: [ 4. -4.]
Recalled Output: [-1  1]
```