MCP_Logic_Gates

May 23, 2021

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

[2]:
```python
def aggregate_func(inputt):
    '''
    aggregate function

    :params
     inputt : input

    :returns
     sum of all inputs
    '''
    return np.sum(inputt)
```

[3]:
```python
def logical_and(inputt):
    '''
    calculates logical AND

    :params
     inputt : input

    :returns
     resultant logical AND
    '''
    num_ele = inputt.shape[0]
    agg = aggregate_func(inputt)

    if agg == num_ele:
        return np.array([1])
    return np.array([0])
```

[4]:
```python
def logical_or(inputt):
    '''
    calculates logical OR

    :params
     inputt : input

    :returns
     resultant logical OR
    '''
    agg = aggregate_func(inputt)
    if agg > 0:
        return np.array([1])
    return np.array([0])
```

[5]:

```python
def logical_not(inputt):
    '''
    calculates logical NOT

    :params
     inputt : input

    :returns
     resultant logical NOT
    '''
    agg = inputt[0]
    if agg == 0:
        return np.array([1])
    return np.array([0])
```

[6]:

```python
def logical_nor(inputt):
    '''
    calculates logical NOT

    :params
     inputt : input

    :returns
     resultant logical NOR
    '''
    agg = aggregate_func(inputt)
    if agg == 0:
        return np.array([1])
    return np.array([0])
```

[7]:

```python
def get_num_features(operation=None):
    '''
    get features

    :params
     operations : type of operatoin to be performed

    :returns
    number of features
    '''
    if operation == 'NOT':
        return 1
    features = input("Enter number of features : ")
    features = int(features)

    if features > 1:
        return features
    else:
        raise ValueError("Invalid value for number of features")
```

[8]:

```
def driver_func():
    '''
    driver fuction of the program
    '''


    features = get_num_features()

    inputt = np.random.randint(2, size = (features))
    print("Input : ",inputt)

    print("\nLogical AND : ",logical_and(inputt))
    print("Logical OR  : ",logical_or(inputt))
    print("Logical NOR : ",logical_nor(inputt))

    features = get_num_features('NOT')

    inputt = np.random.randint(2, size = (features))
    print("\nInput : ",inputt)
    print("Logical NOT : ",logical_nor(inputt))
```

[9]:

```
driver_func()
```

```
Enter number of features :  10

Input :  [1 1 0 1 1 0 0 1 1 1]

Logical AND :  [0]
Logical OR  :  [1]
Logical NOR :  [0]

Input :  [0]
Logical NOT :  [1]
```

[ ]:

```

```