hw3

September 20, 2016

0.1 Utility functions to facilitate computations

```
In [48]: def nand(a, b):
             return not a & b

In [49]: def _and(a, b):
             return a & b

In [50]: def _or(a, b):
             return a | b

In [51]: def nor(a, b):
             return not a | b
```

0.2 Problem 3

```
In [52]: inputs = [(0, 0, 0),
                   (0, 0, 1),
                   (0, 1, 0),
                   (0, 1, 1),
                   (1, 0, 0),
                   (1, 0, 1),
                   (1, 1, 0),
                   (1, 1, 1)]

In [53]: for a, b, c in inputs:
             print((a & b) | (b & c) | (a & c))
```

```
0
0
0
1
0
1
1
1
```

0.3 Problem 4

```
In [54]: inputs = [(0, 0),
                   (1, 0),
                   (0, 1),
                   (1, 1)]

In [55]: for a, b in inputs:
             print( not ((a & b) | ((not a) & (not b))) )
```

```
False
True
True
False
```

0.4 Problem 5

```
In [56]: def circuit(s, i3, i2, i1, i0):
             o3 = _and(not s, i2)
             o2 = _or(_and(s, i3), _and(not s, i1))
             o1 = _or(_and(i2, s), _and(not s, i0))
             o0 = _and(i1, s)

             return o3, o2, o1, o0

In [57]: circuit(1, 1, 0, 1, 1)

Out[57]: (0, 1, 0, 1)
```

0.5 Problem 6

```
In [58]: inputs = [(0, 0),
                   (1, 0),
                   (0, 1),
                   (1, 1)]

In [59]: for a, b in inputs:
             print(nand(nand(a, nand(a, b)), nand(b, nand(a, b))))
```

```
False
True
True
False
```

0.6 Problem 7

```
In [38]: def circuit(a, b):
             return _or(_and(not a, b), _and(not b, a))
```

```
In [39]: for a, b in inputs:
             for c in [0, 1]:
                 print(circuit(circuit(a, b), c))
```

```
0
1
1
0
1
0
0
1
```

0.7 Problem 8

```
In [45]: inputs = [(0, 0, 0),
                   (0, 0, 1),
                   (0, 1, 0),
                   (0, 1, 1),
                   (1, 0, 0),
                   (1, 0, 1),
                   (1, 1, 0),
                   (1, 1, 1)]
```

```
In [47]: for a, b, c in inputs:
             print(_or(_and(not b, not c), _and(b, c)))
```

```
1
0
0
1
1
0
0
1
```