

# **Final Exam: B18**

**Author:** Gabriel Hofer

**Instructor:** Dr. Karlsson

**Course:** CENG-325 CPU & GPU Organization and Architecture

**Date:** December 8, 2020

## Grade for a C

The following are implemented in b18.py:

- The wire file provided in the text in the writeup
- XOR gate
- 2 to 4 decoder

## Usage

```
$ python3 b18.py <input-file> j k m n
```

To run the wire file example provided in the writeup (for a D) :

```
$ python3 b18.py example.txt 4 2 3 2
```

To run the XOR circuit (for a C) :

```
$ python3 b18.py xor.txt 4 4 2 4
```

To run the decoder circuit (for a C) :

```
$ python3 b18.py decoder.txt 4 4 2 4
```

## Submission

Files contained in the Dropbox submission:

- b18.py
- Makefile
- writeup-example.txt
- xor.txt
- decoder.txt

## Program Description

b18.py contains a class called b18 and the main function. main creates an instance of the b18 class and evaluates the circuit in the input file. All command line arguments are passed to the b18 constructor. The b18 class constructor loads the input file, parses it, and stores the pin mappings in a dictionary. b18 has the following member functions:

**nand\_out\_idx** - This function simply maps nand input pins to the output pin belonging to the same nand gate.

**nand** - Returns the nand of two operands.

**output** - This function prints the truth table for the circuit contained in the input file.

**update** - This function has one parameter which is an integer, *it*, pointing to a row in the truth table. update assigns values to input pins (based on the row in the table) and computes the result for each nand in the circuit. Values of the output pins are stored in a dictionary called *out* which is returned by the function.