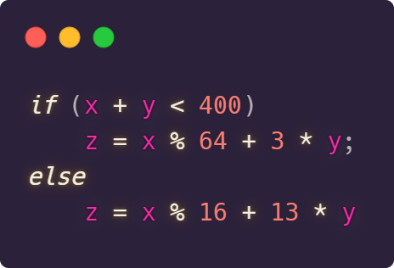
|  |
| --- |
| CSE433 |
| Embedded Systems |
| Assignment 1 Report |

|  |
| --- |
| OĞUZHAN AGKUŞ  161044003 |

# Objective

Design a circuit which will take two unsigned 8-bit numbers x and y as inputs and perform following C code in Verilog:

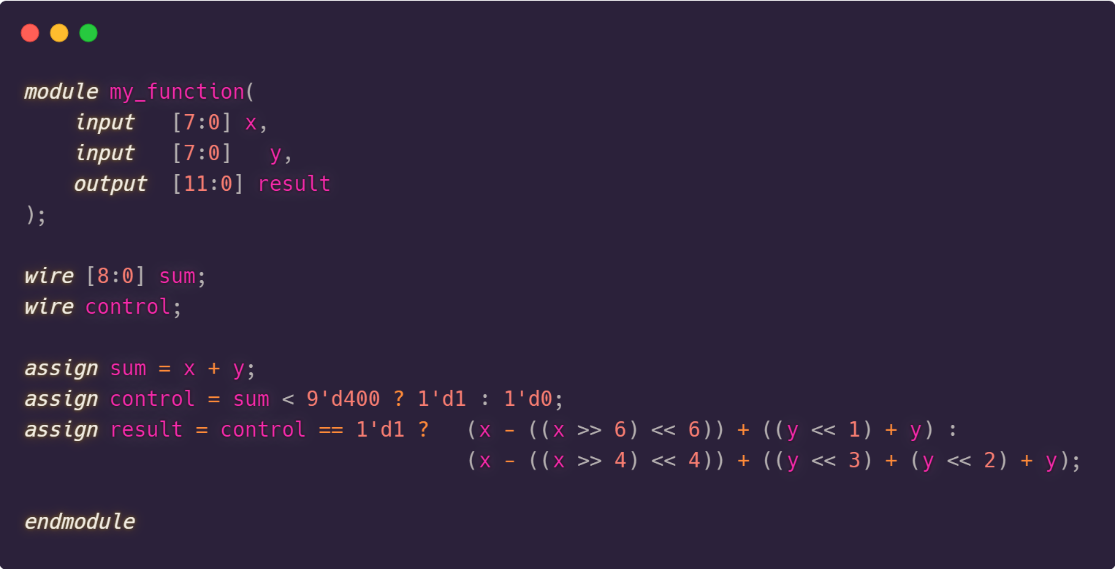


# Restrictions

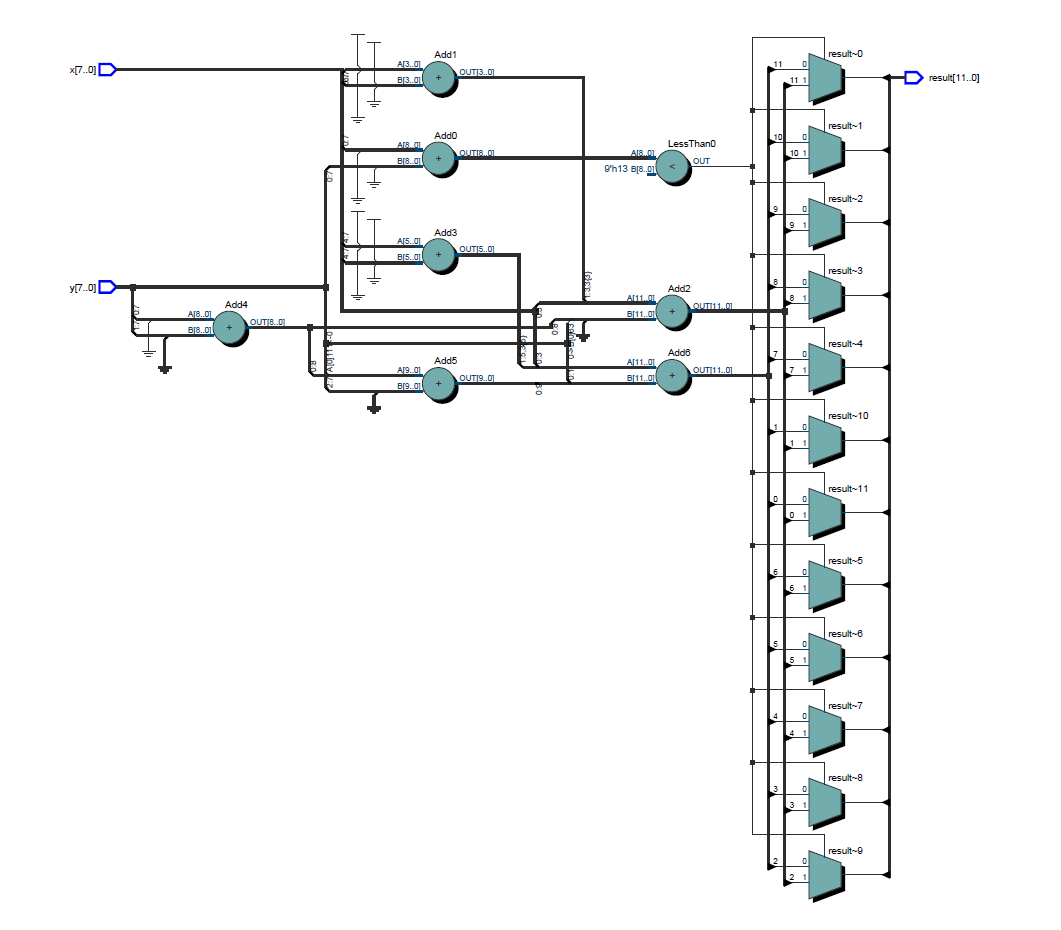
* Use structural or dataflow model Verilog, do not use behavioral Verilog
* Do not use any sequential component like registers and flip-flops
* Do not use any multiplier or \* sign
* Do not use any divisor or / sign

# Design Decisions

* I choose to use dataflow model of Verilog. So, I used “assign” statements.
* Since using \* or / is not allowed, I used bitwise shift operators. So, they can easily multiply and divide a number by 2 and its powers.
* To perform modulo operation, I combined shift operators and addition (subtraction) operator. For example, to find x % 64 equals (x - ((x >> 6) << 6)). Firstly, shift right 6 bits (6 comes from log264) to divide by 64. Then shift right 6 bits to multiple by 64. Finally, get diffidence of input and result, it will give use the remainder.
* Multiplying by powers of 2 is easy but other power are must be combined from powers of 2. For example, 13 \* y = 8 \* y + 4 \* y + 1 \* y = (y << 3) + (y << 2) + y
* By examining given equation, I decided to output should be 12-bit. Biggest result can derive from else condition. Since inputs are 8-bit, y can be 255 at most. So, 13 \* 255 = 3315, which can be show as 12 bits.
* I also needed two temporary data. First one is, sum which represents sum of x + y. It is 9-bit because sum of two 8-bit variable can be bigger than 8-bit. Other one is control bit which represents sum is smaller from 400 or not.

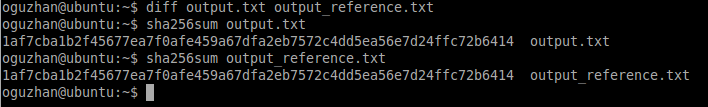


# Semantics



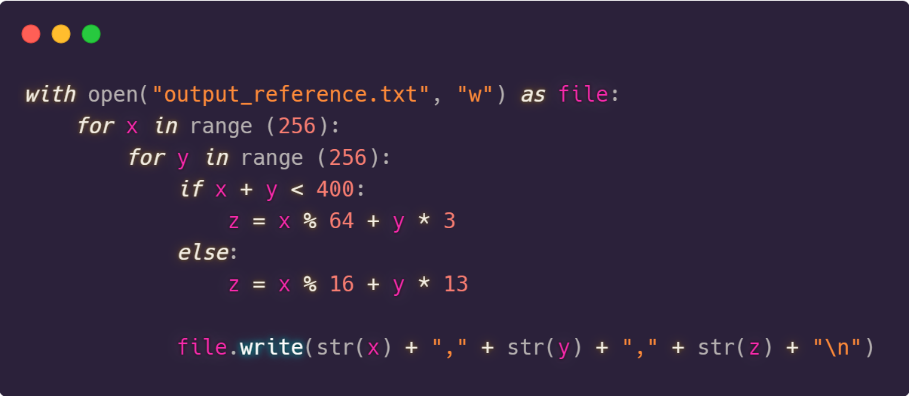
# Testbench and Results

* I have written a testbench which tries each possible input.
* Each input is 8-bit, so it can get 256 different values. Since we have 2 inputs, we have totally 256\*256 = 65536 possible situations.
* It is hard to follow each of them one by one. So, I decide to print all output to the file and check it automatically.
* I write a Python script. It also calculates and prints the results of all combinations. Then I check both output files with extra tools. I used diff command on Linux and compared hashes of both files. Output is here

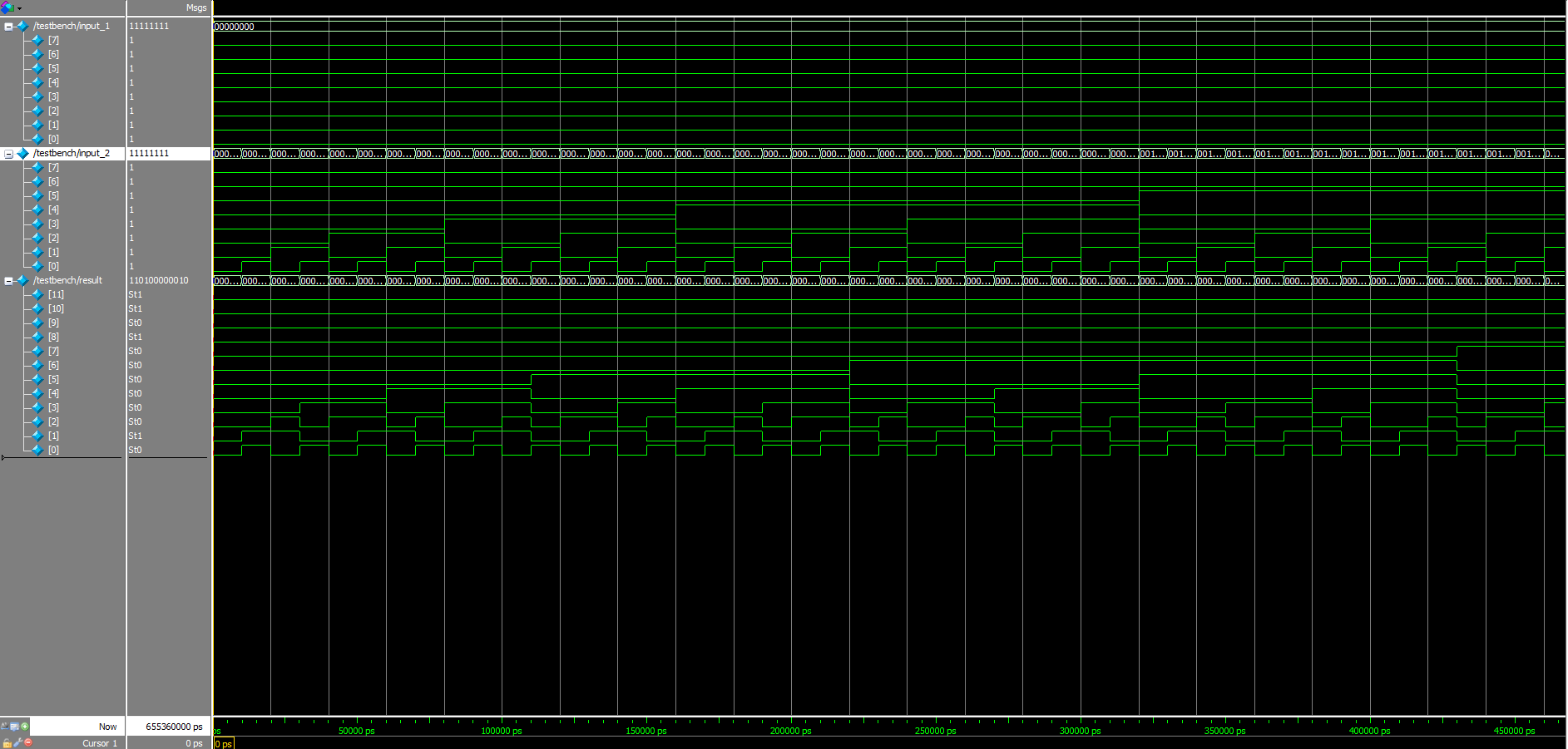




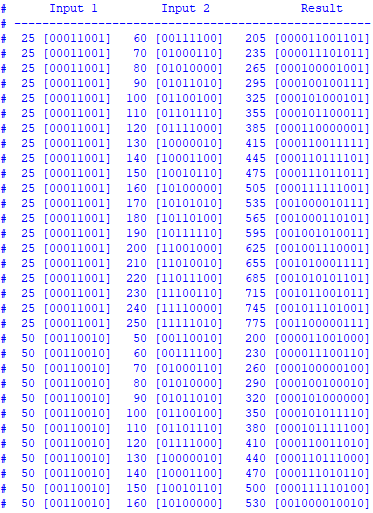
Testbench file



Python script



Simulation screenshot



Terminal output of simulation